Differentially Private Algorithms for Efficient Online Matroid Optimization

by

Kushagra Chandak

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

Master of Science

Department of Department of Computing Science
University of Alberta

Abstract

A matroid bandit is the online version of combinatorial optimization on a matroid, in which the learner chooses K actions from a set of L actions that can form a matroid basis. Many real-world applications such as recommendation systems can be modeled as matroid bandits. In such learning problems, the revealed data may involve sensitive user information. Therefore, privacy considerations are crucial. We propose two simple and practical differentially private algorithms for matroid bandits built on the well-known Upper Confidence Bound algorithms and Thompson Sampling. The key idea behind our first algorithm, Differentially Private Upper Confidence Bound for Matroid Bandits (DPUCB-MAT), is to construct differentially private upper confidence bounds. The second algorithm, Differentially Private Thompson Sampling for Matroid Bandits (DPTS-MAT), is based on the idea of drawing random samples from differentially private posterior distributions. Both algorithms achieve $O\left(L\ln(n)/\Delta + LK\ln(n)\min\left\{K, \ln(n)\right\}/\varepsilon\right)$ regret bounds, where Δ denotes the mean reward gap and ε is the required privacy parameter. Our derived regret bounds rely on novel technical arguments that deeply explore the special structure of matroids. We show a novel way to construct ordered pairs between the played actions and the optimal actions, which contributes to decomposing a matroid bandit problem into K stochastic multi-armed bandit problems. Finally, we conduct experiments to demonstrate the empirical performance of our proposed learning algorithms on both a synthetic dataset and a real-world movie-recommendation dataset.

Preface

This thesis is based on the paper with the same title that is accepted at the Conference on Lifelong Learning Agents - CoLLAs, 2023. The work was primarily done in collaboration with Bingshan Hu and Nidhi Hegde. Bingshan Hu suggested the regret decomposition in Chapter 4 and proof ideas for Theorems 9 and 11. Csaba Szepesvári helped refine Chapters 2, 4 and 5. The proofs for Theorem 4, Theorem 7 and Theorem 8 are by Csaba.

Acknowledgements

Looking back on my two years of master's at the University of Alberta, I am amazed at how much I have learned in that period and I am indebted to a lot of people for this journey. First off, I am very grateful to Csaba Szepesvári for opening the doors of ML theory research for me. Thank you Csaba for all the discussions on ML, sports, and life in general; I promise I will get a road bike soon.

I am also very thankful to Nidhi Hegde and Bingshan Hu for providing me with the opportunity to work on the project on which this thesis is based; I learned more than I could have imagined. A big thanks is also due to Johannes Kirschner for the mentorship and the patience he showed in answering all kinds of absurd questions. It would be remiss of me to not thank Vlad Tkachuk and Alireza Bakhtiari for all the long discussions on every possible topic. I hope our discussions on instance-dependent regret will end one day. Special thanks to Flore Sentenac for pointing out the relevant literature for "Privacy guarantees a user gets." Obviously, I cannot end without thanking my friends and family for all the support they have given me so far. Hopefully, I won't miss any more events in the name of "working on my thesis."

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we study the problem of stochastic optimization on a matroid in a sequential and adaptive way, while taking privacy considerations into account. A matroid is a combinatorial structure that generalizes the notion of independence which will be made precise in Chapter 2. To protect the privacy of users in the dataset, we employ differential privacy as it is the de facto notion for privacy-preserving data analysis. We model the stochastic, sequential and adaptive nature of the problem as a bandit problem.

1.1 Background

A bandit problem is a classic problem to model sequential decision-making under uncertainty. The name 'bandit' comes from slot machines, also called one-armed bandits, in casinos where a gambler is trying to maximize their total money won by choosing different slot machines. More generally, a bandit problem is a sequential game between a *learner* and an *environment*. In each round of the interaction, the learner chooses an *action* (or an "arm") and gets some feedback or a *reward*. The goal of the learner is to choose actions in a way that maximizes the total expected reward collected over all the rounds of interaction. The central challenge that the learner faces is the *exploitation-versus-exploration* dilemma: The learner needs to decide whether to choose an action that it knows has given a high reward in the past or to choose a new action that might give an even higher reward.

In this thesis, we study a variant of the bandit problem called stochastic matroid ban-

dits first proposed by Kveton, Wen, Ashkan, Eydgahi, et al. (2014). Matroids (Whitney, 1935) are well-studied in combinatorial optimization because many common combinatorial problems, like finding a minimum spanning tree, can be formulated as a matroid optimization problem. Solving a combinatorial optimization problem can be computationally hard in general. However, optimization on a matroid can be solved greedily and hence problems that can be modeled on a matroid are often computationally efficient.

We are interested in stochastic and online optimization of a matroid using the framework of matroid bandits. In a stochastic matroid bandit problem, we have a matroid $\mathcal{M} = (E, \mathcal{I})$ and a stochastic environment ν, where E is a set of L items called *base arms* and $\mathcal{I}$ is a family of *independent sets* formed using the subsets of E. Each base arm $e \in E$ is associated with a weight or a *reward* that is independently drawn from a fixed but unknown probability distribution P_e. The environment ν is given by $\nu = (P_e : e \in E)$. As matroids generalize the notion of independence, we are interested in a *basis* of a matroid which is a maximal independent set $A \in \mathcal{I}$. All the bases of a matroid have the same size denoted by K. Each basis can be viewed as a *super arm* which consists of exactly K base arms. In matroid bandits, we want to learn the basis or the super arm with the maximum total reward by interacting sequentially with environment ν over n rounds. In each round $t \in [n]$, the learner chooses a super arm A^t and simultaneously, the environment generates a random *reward* vector $w_t := (w_t(e_1), w_t(e_2), \ldots, w_t(e_L))$ with independent components for all the base arms $e_i \in E$. At the end of round t, the learner observes each individual reward $w_t(e)$ for all $e \in A^t$ and obtains as *return* the sum of the rewards of the base arms in A^t. The goal of the learner is to choose super arms sequentially to maximize the total return over a finite number of n rounds.

As a motivating example, consider the problem of sequentially and adaptively recommending a set of diverse movies to users using their movie-ratings data. Such a problem can be formulated as a matroid bandit problem (Kveton, Wen, Ashkan, Eydgahi, et al., 2014). In this example, each movie can be characterized by a feature vector denoting the genres of that movie. The set of feature vectors for all the movies form the set E of base arms.

In each round, the set of recommended movies can be viewed as a super arm. Since we would like the recommended movies to have diverse genres, the matroid structure ensures that the feature vectors of the recommended movies are linearly independent. This means that the recommended movies do not include movies with similar genres. At the end of the round, based on the rating feedback from the users, the learning algorithm adjusts its future recommendations.

Since the learner only observes the weights associated with base arms in the selected super arm, it still faces the *exploitation-versus-exploration* dilemma. In each round, the learner needs to decide whether to choose a super arm with the highest empirical return based on the past information (exploitation) or choose an under-explored super arm to gain information about the unknown environment (exploration). Upper Confidence Bound (UCB)-based (Auer et al., 2002; Garivier et al., 2011) and Thompson Sampling-based learning algorithms (Kaufmann et al., 2012; Agrawal et al., 2017) are both successful in balancing the exploitation-versus-exploration trade-off. In the UCB-based algorithms, each arm maintains a UCB index, which is an upper bound of the constructed confidence interval around the mean reward estimate. The learning algorithm makes a decision based on UCB indices for all the arms. Different from the UCB-based algorithms that rely on the UCB indices to tackle the exploitation-versus-exploration dilemma, Thompson Sampling-based algorithms maintain a posterior distribution on a parameter for each arm. For each arm, a random sample is drawn from the posterior distribution. The learning algorithm makes a decision based on the random samples for all the arms. If the learning algorithm knows the mean rewards of all the arms, choosing the arm with the highest mean reward is always the best strategy to achieve the highest expected reward. However, we do not know the mean rewards and seek to learn them. Recall that the learning algorithm aims to maximize the total expected reward collected during the interaction with the environment. Equivalently, the performance of bandit algorithms is measured by *regret*, which represents the expected cumulative performance gap between always choosing the arm with the highest expected reward (fixed but unknown) and the arms selected by the learning algorithm.

The example of recommending diverse movies highlights the necessity of preserving the privacy of users. The feedback information from users (e.g. movie ratings) also reveals their watch history or preferences towards the recommended movies. Being users, we may wish to keep this information private. Take the Netflix Prize dataset for example. As shown by Narayanan et al. (2008), just anonymizing the ratings is not enough to preserve privacy. In this paper, we focus on the learning problem of matroid bandits with differential privacy. Differential privacy is the most commonly-used notion of privacy for machine learning algorithms (Dwork and Roth, 2014). If a learning algorithm is implemented in a differentially private manner, the information associated with an individual has almost no impact on the output of the learning. In other words, differentially private learning algorithms are not sensitive to information from a single individual. In this work, we focus on ε-differential privacy, where ε is the privacy parameter. The parameter ε can be viewed as the privacy budget that can be distributed among different components of the learning algorithm. It also measures the information leakage by the learning algorithm.

1.2 Contributions

Now, we summarize the key contributions of this work.

1. In this work, we propose two *sample efficient* and *computationally fast* differentially private algorithms for matroid bandits. Our first algorithm, `DPUCB-MAT`, is built upon the well-established UCB1 policy of Auer et al. (2002). The regret bound of `DPUCB-MAT` is

$$O\left(L\ln(n)/\Delta + \min\left\{K, \log(n)\right\} LK\ln(n)/\varepsilon\right) ,$$

   where Δ is the mean reward gap and ε is the required privacy parameter. Our second algorithm, `DPTS-MAT`, is built on Thompson Sampling which has demonstrated competitive practical performances (Chapelle et al., 2011). The regret bound of `DPTS-MAT` is

$$O\left(L\ln(n)/\Delta + \min\left\{K, \log(n)\right\} LK\ln(n)/\varepsilon\right) + O\left(K\ln(n)/\Delta\right) .$$

2. Regarding the regret analysis, we propose a unified approach to decompose the regret of `DPUCB-MAT` and `DPTS-MAT`. Our novel regret decomposition relies on the introduction of a round-dependent permutation on the order of the base arms in the optimal super arm. The permutation contributes to decomposing the regret of matroid bandits into K different stochastic bandit problems.

3. We conduct experiments to evaluate the empirical performance of our proposed algorithms by using both synthetic and real-world movie-recommendation datasets. The experimental results demonstrate that our proposed differentially private algorithms are efficient.

Outline of the thesis. We start off by describing the problem setting in Chapter 2. In this chapter, we define matroid bandits and give an overview of differential privacy. We also define how differential privacy is formulated in the context of matroid bandits. In the next chapter, Chapter 3, we discuss some closely related works to that of ours. In Chapter 4, we describe our algorithms with their regret bounds. Next, in Chapter 5, we provide proofs for the regret bounds. We then describe our experimental setting and show the experimental results in Chapter 6. We conclude with a summary and future work in Chapter 7.

Chapter 2

Setting

In this chapter, we introduce our problem setting formally. We first introduce matroid bandits. We then introduce differential privacy and discuss differential privacy in the context of matroid bandit learning algorithms.

2.1 Matroid Bandits

A stochastic matroid bandit (Kveton, Wen, Ashkan, Eydgahi, et al., 2014) problem is a tuple $(\mathcal{M}, \nu)$, where $\mathcal{M} = (E, \mathcal{I})$ is a matroid and ν is an environment that a learner interacts with. Before describing how a matroid characterizes the environment, let us first define what a matroid is.

2.1.1 Matroids

Definition 1. *(Matroid (Kveton, Wen, Ashkan, Eydgahi, et al., 2014)). A matroid $\mathcal{M}$ is a tuple $(E, \mathcal{I})$. The first component of a matroid $\mathcal{M} = (E, \mathcal{I})$ is a set E of L items and can be written as $E = \{1, \ldots, L\}$. The second component $\mathcal{I}$ is a family of subsets of E, also called the family of* independent sets*, defined by the following three properties.*

1. *Empty set is independent: $\varnothing \in \mathcal{I}$.*
2. *A subset of an independent set is independent: If $X \subset Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$.*
3. *Augmentation: If $X, Y \in \mathcal{I}$ such that $|X| = |Y| + 1$, then there exists $e \in X \setminus Y$ such that $Y \cup \{e\} \in \mathcal{I}$.*

A matroid generalizes the idea of independence, for example, in vector spaces. Similar to vector spaces, a maximal independent set $A \in \mathcal{I}$ is called a *basis* of the matroid $\mathcal{M}$. All the bases have the same cardinality denoted by K. Additionally, if a weight $\bar{w}(e) \in [0, 1]$ is associated with each item $e \in E$, the matroid is called a *weighted matroid*.

2.1.2 Environment and the Learning Protocol

Recall that each item $e \in E$ is called a base arm. Each base arm e is associated with a weight that is drawn independently of other base arms and independently over time from a fixed but unknown probability distribution P_e with mean $\bar{w}(e)$. The collection $\nu = (P_e : e \in E)$ defines the environment. We assume that P_e is supported on $[0, 1]$. In matroid bandits, we are interested in sequentially choosing a basis. Each basis $A \subset E$ can be viewed as a *super arm* which consists of exactly K base arms. More concretely, we are interested in learning the super arm with the maximum total weight, i.e., learn the super arm $A^* = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \bar{w}(e)$. This is the central challenge of matroid bandits. If we knew $\bar{w}(e)$ for all $e \in E$, the optimal super arm A^* could be found greedily as shown in Algorithm 1. For simplicity, we assume that A^* is unique.

In the matroid bandit problem, we do not assume we know $\bar{w}(e)$ in advance. Instead, a learner can interact with the environment ν sequentially to learn $\bar{w}(e)$ with the following learning protocol. In each round $t \in [n]$:

1a. The environment generates a random weight vector $w_t = (w_t(e_1), \ldots, w_t(e_L))$ with each entry $w_t(e_i)$ i.i.d over time according to P_{e_i}. Each component of the vector is also independent of the others.

1b. Simultaneously, the learner selects a super arm $A^t = \{a_1^t, \ldots, a_K^t\}$ (or, a basis of the matroid).

2. The learner observes the random weights $\{w_t(e) : e \in A^t\}$ and receives as return the sum of all the weights in A^t, i.e., the learner obtains a return $f(A^t, w_t) = \sum_{e \in A^t} w_t(e)$.

The goal of the learner is to choose super arms sequentially over a finite of n rounds to maximize the expected cumulative return. Let $A^* := \{a_1^*, \ldots, a_k^*, \ldots, a_K^*\}$ be the optimal super arm with $\bar{w}(a_1^*) \geq \ldots \geq \bar{w}(a_k^*) \geq \ldots \geq \bar{w}(a_K^*)$. We use (pseudo)-regret R_n to measure the quality of the learner's strategy for deciding which super arms to choose, which is defined as

$$R_n = \sum_{t=1}^{n} \mathbb{E}\left[f(A^*, \bar{w}) - f(A^t, \bar{w})\right] , \tag{2.1}$$

where the expectation is over the randomness in the learner's action selection strategy.[1]

Algorithm 1 Greedy algorithm to find a maximum weight basis of a matroid (Kveton, Wen, Ashkan, Eydgahi, et al., 2014)

Input: Matroid $(E, \mathcal{I})$ and all weights $\bar{w}(e)$ for all $e \in E$; **Output:** A^*
Sort all weights such that $\bar{w}(e_1) \geq \ldots \geq \bar{w}(e_L)$ for all $e_i \in E$
Initialize $A^* = \varnothing$
for $i = 1, \ldots, K$ **do**
    $\sigma(A^*) = \{e : e \in E \setminus A^*, A^* \cup \{e\} \in \mathcal{I}\}$ ▷ Base arms that can be added to A^*
    $e_i = \arg\max_{e \in \sigma(A^*)} \bar{w}(e)$.
    $A^* \leftarrow A^* \cup \{e_i\}$ ▷ Add the base arm with the highest weight in $\sigma(A^*)$ to A^*
end for

2.2 Differential Privacy

Differential privacy (DP) is the most commonly used notion for preserving the privacy of individuals in machine learning algorithms. It is based on the idea that the change of a single user's data in the dataset cannot impact the output of the learning algorithm too much. The insensitivity to a single user's data guarantees that from the algorithm's output, an adversary cannot learn the existence or anything useful about that user. In other words, differential privacy describes a promise made to a user that the output of a differentially private algorithm would almost be the same even if the user was not present in the dataset. So the user might as well participate in the dataset for the statistical analysis carried out by the algorithm.

[1] Note that $\bar{w}$ is the mean reward vector accounting for all the base arms.

Before we define differential privacy for matroid bandits, we first define it for any randomized mechanism, along with some of its properties.

Definition 2. *(Randomized mechanism). A randomized mechanism $\mathcal{A}$ is a map $\mathcal{A} : \mathcal{X} \times [0,1] \to \mathcal{Y}$, where $\mathcal{X}$ is an arbitrary set and $\mathcal{Y}$ is a measurable space. On input $\mathbf{x} \in \mathcal{X}$, the mechanism also takes in $U \sim \text{Unif}[0,1]$ and outputs $\mathcal{A}(\mathbf{x}, U) \in \mathcal{Y}$.*

Note that in the definition of a randomized algorithm, we have only used a uniformly distributed random variable. However, this is not restrictive (as long as $\mathcal{Y}$ is "reasonable"). When $\mathcal{Y} = \mathbb{R}$, to get samples from any other distribution, we can use the inverse cumulative distribution function or inverse CDF. This idea is called *inverse transform sampling*. To state this formally, for a real-valued random variable X, we let $\mathbb{P}_X$ denote its probability distribution (over the reals).

Claim 1. *For any $\mathcal{A}_0 : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ and a distribution P over $\mathcal{Z}$, there exists $\mathcal{A} : \mathcal{X} \times [0,1] \to \mathbb{R}$ such that for all $\mathbf{x} \in \mathcal{X}$, $\mathbb{P}_{\mathcal{A}_0(\mathbf{x},Z)} = \mathbb{P}_{\mathcal{A}(\mathbf{x},U)}$ where $Z \sim P$, $U \sim \text{Unif}[0,1]$.*

Proof. Let $\mu(\mathbf{x}, r) = \mathbb{P}_{\mathcal{A}_0(\mathbf{x},Z)}((-\infty, r]))$. Let $\mathcal{A}(\mathbf{x}, u) = \sup\{r \in \mathbb{R} : \mu(\mathbf{x}, r) < u\}$. Note that $u \mapsto \mathcal{A}(\mathbf{x}, u)$ is measurable because the supremum can be taken over rational numbers r and the function $u \mapsto \mu(\mathbf{x}, r) - u$ is measurable. Now, for any $s \in \mathbb{R}$, since the map $u \mapsto \mu(\mathbf{x}, u)$ is lower-semicontinuous and increasing, for any $u \in [0,1]$, $\mathcal{A}(\mathbf{x}, u) \leq s$ holds if and only if $u \leq \mu(\mathbf{x}, s)$. Thus,

$$\mathbb{P}(\mathcal{A}(\mathbf{x}, U) \leq s) = \mathbb{P}(U \leq \mu(\mathbf{x}, s)) = \mu(\mathbf{x}, s)\,.$$

Since this holds for any $s \in \mathbb{R}$, $\mathbb{P}_{\mathcal{A}(\mathbf{x},U)} = \mathbb{P}_{\mathcal{A}_0(\mathbf{x},Z)}$. ■

For convenience, we allow $(\mathcal{A}_0, P)$ as randomized mechanisms (Definition 2). Next we define the notion of neighboring datasets which will be used to define differential privacy. For the remainder of this chapter, we assume that $\mathcal{X}_0$ is some arbitrary set.

Definition 3. *(Hamming distance). The Hamming distance between $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_0^n$ is defined as $h(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n \mathbb{1}\{x_i \neq x_i'\}$.*

Definition 4. *(Neighboring datasets).* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_0^n$ *are neighboring if* $h(\mathbf{x}, \mathbf{x}') = 1$.

Intuitively, if we view each $x_i \in \mathcal{X}_0$ as a user's record then neighboring datasets $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_0^n$ differ in one user's record. We remark that even though $\mathbf{x} \in \mathcal{X}_0^n$ is a tuple, we call it a data**set**. Throughout this thesis, we call such a tuple a dataset following the convention of the differential privacy literature. Now we are ready to define differential privacy.

Definition 5 (ε-differential privacy for randomized mechanisms (Dwork and Roth, 2014, Definition 2.4))**.** *Fix* $\varepsilon > 0$. *A randomized mechanism* $\mathcal{A}$ *is* ε*-differentially private if for all measurable* $S \subseteq \mathbf{Range}(\mathcal{A})$ *and for every pair of neighboring datasets* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_0^n$*, we have*

$$\mathbb{P}\left\{\mathcal{A}(\mathbf{x}, U) \in S\right\} \leq e^{\varepsilon} \cdot \mathbb{P}\left\{\mathcal{A}(\mathbf{x}', U) \in S\right\} ,$$

where $U \sim \text{Unif}[0, 1]$.

Discussion. Notice that differential privacy is a feature of the *algorithm* or the *mechanism*. Therefore, to ensure differential privacy we need to understand the process that generated the output from the input data. The privacy parameter ε is used to control how far apart the output distributions of $\mathcal{A}$ are under $\mathbf{x}$ and $\mathbf{x}'$. It is usually a small number but not cryptographically small like 2^{-100}. Making the privacy parameter that small would make the output distributions *too* close thereby preventing us from doing anything useful with the algorithm.

We next state two important properties of differential privacy: post-processing and group privacy. Post-processing states that the privacy guarantee cannot get worse if we release a function of the output instead of the output itself. Group privacy states that the differential privacy guarantee degrades nicely for a group of people.

Lemma 2 ((Dwork and Roth, 2014, Proposition 2.1))**.** *Let* $\mathcal{A} : \mathcal{X} \times [0, 1] \to \mathcal{Y}$ *and* $\mathcal{A}' : \mathcal{Y} \times [0, 1] \to \mathcal{Z}$ *be two randomized mechanisms. If* $\mathcal{A}(\cdot, U)$ *is* ε*-DP with* $U \sim \text{Unif}[0, 1]$*, then the composition* $\mathcal{A}'(\mathcal{A}(\cdot, U), U')$ *is also* ε*-DP, where* $U' \sim \text{Unif}[0, 1]$ *is independent of* U.

Lemma 3. *(Group privacy). Let* $\mathbf{x}$ *and* $\mathbf{x}'$ *be two datasets that differ in* k *positions for* $k \geq 1$*. If a randomized mechanism* $\mathcal{A}$ *is* ε*-differentially private, then for all measurable* $S \subseteq \mathbf{Range}(\mathcal{A})$*, we have*

$$\mathbb{P}\left\{\mathcal{A}(\mathbf{x}, U) \in S\right\} \leq e^{k\varepsilon} \cdot \mathbb{P}\left\{\mathcal{A}(\mathbf{x}', U) \in S\right\} ,$$

where $U \sim \mathrm{Unif}[0, 1]$.

The property of group privacy also gives us a lower bound on the privacy parameter below which no utility can be extracted from a private mechanism. Clearly, two datasets can differ at maximum n positions. Using group privacy for such datasets, we have that $k = n$ and

$$\mathbb{P}\left\{\mathcal{A}(\mathbf{x}, U) \in S\right\} \leq e^{n\varepsilon} \cdot \mathbb{P}\left\{\mathcal{A}(\mathbf{x}', U) \in S\right\} .$$

So if $\varepsilon \ll 1/n$, then the output distributions on the two datasets would almost be the same. This means that for two datasets that are completely different from each other, the output distributions are almost the same, which is undesirable (assuming the algorithm needs to compute something non-constant). Therefore, we want that for useful differential private algorithms

$$\varepsilon \gg \frac{1}{n} .$$

2.2.1 Privacy guarantees a user gets

The next theorem shows how the success rate of an adversary whose goal is to recover private information about any one particular user is controlled by an ε-differential private algorithm. In fact, the result shows that the failure rate of such an adversary is high (and close to random chance), when ε is close to zero. The theorem's value is that it gives a precise expression of how high the failure rate of *any* adversary needs to be.

To state this result, we need a little preparation. Fix a user $i \in [n]$, whose data we need to protect. Let $s : \mathcal{X}_0 \to \{0, 1\}$ describe the bit of information we want to protect. The question we ask is as follows:

Given a dataset $\mathbf{x} \in \mathcal{X}_0^n$*, with what success rate can an adversary recover* $s(x_i)$ *having access to* (i) *the data of all the users* $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ *other than user* i*,* (ii) *the output* Y *of an* ε*-differentially private algorithm* $\mathcal{A}$ *(i.e.,* $Y = \mathcal{A}(\mathbf{x}, U)$ *with* U *uniform random from* $[0,1]$*) and* (iii) *the algorithm* $\mathcal{A}$ *itself.*

Of course, if $s(x) = 0$ for all $x \in \mathcal{X}_0$, there is no secret, hence to make the problem meaningful, assume that there is $x, x' \in \mathcal{X}_0$ such that $s(x) \neq s(x')$. In any case, based on the above problem definition, an adversary would use some function $r : \mathcal{Y} \to \{0,1\}$ to recover the bit where $\mathcal{Y} = \mathbf{Range}(\mathcal{A})$ is the set of values $\mathcal{A}$ can produce. (The function r here can depend on $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$: this dependence is suppressed to minimize clutter.) Then, given $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, we define the failure probability of the adversary who uses r to be

$$\varphi_{\mathbf{x}_{-i}}(r) = \max_{x \in \mathcal{X}_0} \mathbb{P}_{x, \mathbf{x}_{-i}} \{ r(Y) \neq s(x) \} \ .$$

where $\mathbb{P}_{x, \mathbf{x}_{-i}}$ is a probability distribution under which the distribution of Y is the same as the distribution of $Y = \mathcal{A}(\mathbf{x}, U)$ and U is a uniform $[0,1]$-valued random variable where $\mathbf{x} = (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$. In words, the failure rate of the adversary using r is the probability of the adversary failing to recover the truth for *some* value of $x \in \mathcal{X}_0$ of the ith user's data. For completeness, one could also allow adversaries who randomize. The weakest adversary then is the one who would guess the secret bit based on the outcome of flipping an unbiased coin. The failure probability of such an adversary is $1/2$. More intelligent adversaries should have smaller failure rates. Note that adversaries are assessed by their worst-case failure rate. We do this to capture the notion that we want the failure rate to reflect how much the adversary can learn from some data release. Indeed, an adversary who always bets on just a fixed bit of either zero or one will succeed with probability one(!) in some cases. Yet, this adversary clearly did not learn much about the user from the data release, but was just merely lucky when its bet was correct.

Theorem 4 (Limits of adversaries against differentially private algorithms)**.** *Take any* $\varepsilon > 0$*, any non-constant function* $s : \mathcal{X}_0 \to \{0,1\}$*, any integer* $i \in [n]$ *and any incomplete dataset*

$\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathcal{X}_0^n$ *that is missing the data of user* i*. Then, the best failure rate that an adversary can achieve against an* ε*-differentially private algorithm even when given* $\mathbf{x}_{-i}$ *is* $1/(1+e^{\varepsilon})$*:*

$$\inf_{r:\mathcal{Y}\to\{0,1\}} \varphi_{\mathbf{x}_{-i}}(r) \geq \frac{1}{1+e^{\varepsilon}} .$$

While the theorem does not allow for randomizing adversaries, this is only for the sake of simplicity. At the price of a slightly more complicated notation, the theorem extends easily to randomizing adversaries, who are subject to the same limits as non-randomizing adversaries. This should also be intuitive: An adversary should not be able to gain information about the "secret" hidden from them by just injecting extra noise in the way they decipher the secret.

Proof. Fix $r : \mathcal{Y} \to \{0,1\}$. Let $x, x' \in \mathcal{X}_0$ such that $s(x) = 0$ and $s(x') = 1$. These exist because s is not constant. Let $\mathbf{x} = (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$ and $\mathbf{x}' = (x_1, \ldots, x_{i-1}, x', x_{i+1}, \ldots, x_n)$. Let $\mathcal{A}$ be the ε differentially private mechanism. Let $\mathcal{Y} = \mathbf{Range}(\mathcal{A})$ be the range of values that $\mathcal{A}$ can produce. Let $\mathbb{P}_x = \mathbb{P}_{x,\mathbf{x}_{-i}}$ and $\mathbb{P}_{x'} = \mathbb{P}_{x',\mathbf{x}_{-i}}$. Then,

$$\begin{aligned}
\mathbb{P}_{x'}\{r(Y) \neq 0\} &= \mathbb{P}\{r(\mathcal{A}(\mathbf{x}', U)) \neq 0\} \\
&= \mathbb{P}\left\{\mathcal{A}(\mathbf{x}', U) \in \mathcal{Y} \setminus r^{-1}(0)\right\} \\
&\leq e^{\varepsilon}\mathbb{P}\left\{\mathcal{A}(\mathbf{x}, U) \in \mathcal{Y} \setminus r^{-1}(0)\right\} && \text{(using Definition 5)} \\
&= e^{\varepsilon}\mathbb{P}_x\{r(Y) \neq 0\} . && (2.2)
\end{aligned}$$

Hence, we have

$$
\begin{aligned}
\varphi_{\mathbf{x}_{-i}}(r) &\geq \max\left\{\mathbb{P}_x\left\{r(Y) \neq s(x)\right\}, \mathbb{P}_{x'}\left\{r(Y) \neq s(x')\right\}\right\} \\
&= \max\left\{\mathbb{P}_x\left\{r(Y) \neq 0\right\}, \mathbb{P}_{x'}\left\{r(Y) \neq 1\right\}\right\} \\
&= \max\left\{\mathbb{P}_x\left\{r(Y) \neq 0\right\}, 1 - \mathbb{P}_{x'}\left\{r(Y) \neq 0\right\}\right\} \\
&\geq \max\left\{\mathbb{P}_x\left\{r(Y) \neq 0\right\}, 1 - e^{\varepsilon}\mathbb{P}_x\left\{r(Y) \neq 0\right\}\right\} && \text{(because of Eq. (2.2))} \\
&\geq \inf_{0 \leq \gamma \leq 1} \max\left\{\gamma, 1 - e^{\varepsilon}\gamma\right\} && \text{(because } \mathbb{P}_x\left\{r(Y) \neq 0\right\} \in [0,1]\text{)} \\
&= \frac{1}{1+e^{\varepsilon}}, && \text{(} \gamma = 1/(1+e^{\varepsilon}) \text{ is the minimizer)}
\end{aligned}
$$

finishing the proof. ■

Figure 2.1 shows the failure probability of an adversary as a function of the privacy parameter ε. From the figure, we can see that for small values of ε, we can think of the failure probabilities of optimal adversaries to be linearly related to ε.

There are results similar to Theorem 4 in the literature, for example, Wasserman et al. (2010, Theorem 2.4). They use the Neyman-Pearson lemma to derive their result which, as our proof shows, is not needed. Our proof is essentially the same as that of Geng et al. (2014) who is reproving the result of Wasserman et al. (2010). Kairouz et al. (2015) shows a similar result for (ε, δ)-differential privacy.

While differential privacy guarantees some protection for the user, any information release, differentially private or not, can give an adversary critical information with which they can recover some protected information about a user. To explain how this can happen, consider the "height parable" told by Dwork and Naor (2010):

> *"Suppose one's exact height were considered a sensitive piece of information, and that revealing the exact height of an individual were a privacy breach. Assume that the database yields the average heights of women of different nationalities. An adversary who has access to the statistical database and the auxiliary information "Terry Gross is two inches shorter than the average Lithuanian woman" learns Terry Gross' height, while anyone learning only the auxiliary information,*

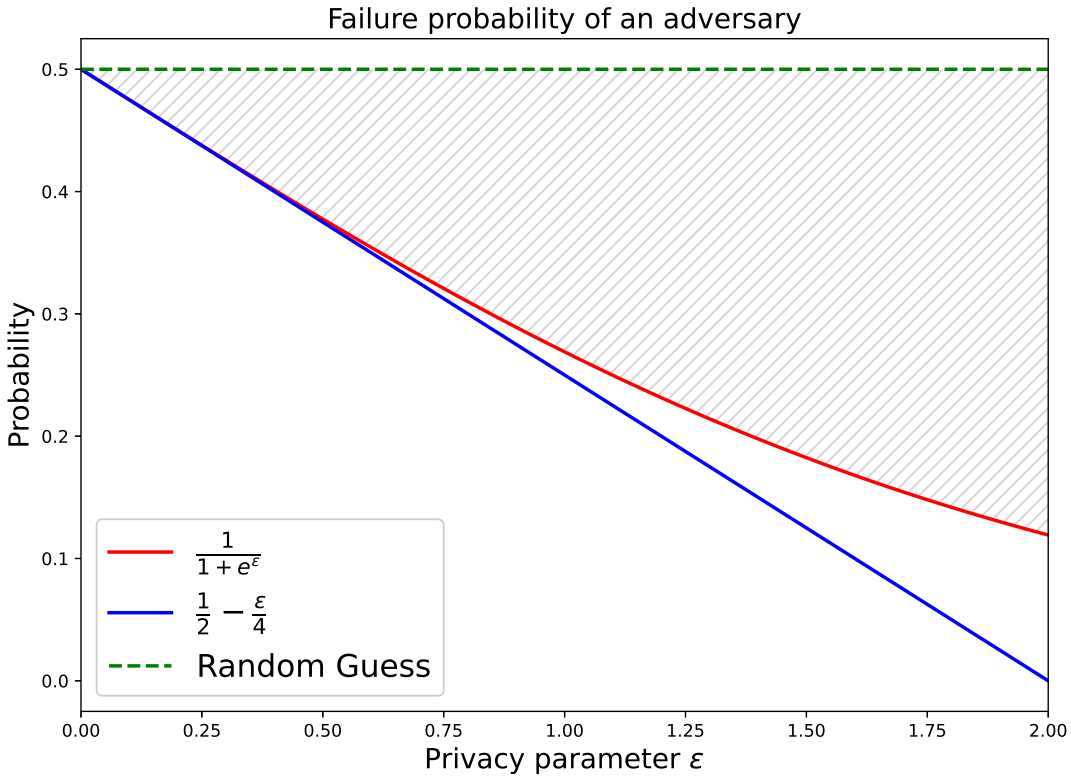


Figure 2.1: **Failure probability of an adversary**. The shaded region represents that the failure probability of an adversary is restricted to, as a function of ε. The blue curve represents the line $1/2 - \varepsilon/4$. This linear approximation gives a lower bound for the adverseries' failure probabilities, which, for small values of ϵ is tight.

without access to the average heights, learns relatively little."

From the above example, we see that there is more information learned about a user (Terry Gross) after using the average height information than before. How is this possible given Theorem 4? For example, let the secret bit to be protected is whether Terry Gross is shorter than a certain number h. Let Terry Gross be the first user in the database, i.e., the height of Terry Gross is x_1. Furthermore, let $x_2, \ldots, x_n$ be the height of the other Lithuanian women. In the setting of Theorem 4, the adversary has access to $x_2, \ldots, x_n$. Now, if the adversary also knows that Terry Gross's height is 2 inches below the average then they can set up the linear equation $x_1 = \frac{1}{n}\sum_{i=1}^{n} x_i - 2$, from which $x_1 = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=2}^{n} x_i - 2\right)$, which the adversary can easily evaluate, and from which, the bit that was supposed to be protected can be obtained with certainty.

Why does this not contradict Theorem 4? This is because in this result we implicitly removed the possibility of adversaries that have prior information about what configurations in $\mathcal{X}_0^n$ are possible. The theorem can be trivially extended to the case of adversaries with such prior information. Let $V \subset \mathcal{X}_0^n$ be this set. Given V, the definition of the worst-case failure rate of adversaries that know V needs to be modified to

$$\varphi_{\mathbf{x}_{-i},V}(r) = \max\{\mathbb{P}_{x,\mathbf{x}_{-i}}\{r(Y) \neq s(x)\} : (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n) \in V\}.$$

The modified theorem looks as follows:

Theorem 5 (Limits of adversaries that have prior knowledge)**.** *Take any $\varepsilon > 0$, any non-constant function $s : \mathcal{X}_0 \to \{0,1\}$, any integer $i \in [n]$ and any incomplete dataset $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathcal{X}_0^n$ that is missing the data of user i. Let $V \subset \mathcal{X}_0^n$ arbitrary. Then, if*

$$\begin{aligned} &\text{there exists } x, x' \in \mathcal{X}_0 \text{ such that } (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n), \\ &(x_1, \ldots, x_{i-1}, x', x_{i+1}, \ldots, x_n) \in V \text{ and } s(x) \neq s(x') \end{aligned} \tag{2.3}$$

then

$$\inf_{r:\mathcal{Y}\to\{0,1\}} \varphi_{\mathbf{x}_{-i},V}(r) \geq \frac{1}{1+e^{\varepsilon}}.$$

As the proof goes through without any changes from the last one, it is omitted.

It is also clear that Eq. (2.3) is a necessary condition in the sense that if this condition is not met, there will be adversaries who can uncover the protected bit of the user with certainty, no matter the protection applied in the data release. This is, however, an uninteresting case as the adversary, before the release of the data, already possessed the knowledge that was supposed to be protected, assuming that they had access to $\mathbf{x}_{-i}$. This nuance is important: The theorems are formulated so that they allow the adversary to have access to $\mathbf{x}_{-i}$ – this makes the guarantees for user i stronger. However, if an adversary did not have access to $\mathbf{x}_{-i}$, but if $\mathbf{x}_{-i}$ is "leaked", the adversary gains extra knowledge concerning user i. This calls attention to the fact that adversaries can gain decisive information about a user even if the user's data is not part of the data released. Furthermore, this is not even impacted by the fact whether the data release was itself differentially private: Any information release, together with the "right background information", may reveal sensitive information about any person: This is just the very nature of information release.

To sum up the discussion, the guarantee for a user that differential privacy can give is that regardless of whether their data is included in the dataset or not, adversaries will have approximately the same information about them. Thus, a user whose power is to decide whether they are willing to participate in a process that leads to data release, should have no concerns. Yet, this clearly does not imply that differentially private data release will not contribute to information available to potential adversaries about users. At the risk of repeating the obvious, the fact that information is released at all can and generally will increase the knowledge available to anyone who has access to the released information.

Next we discuss how we can make a randomized mechanism differentially private.

2.2.2 The Laplace Mechanism

One way to ensure differential privacy is to add calibrated noise to mask the output of a function. This function typically computes a statistic that we are interested in, e.g., an estimate of the mean reward. The scale of the added noise is defined using the *sensitivity* of

the function:

Definition 6. *(Sensitivity (Dwork and Roth, 2014)). The sensitivity of a function* $f : \mathcal{X} \to \mathbb{R}$ *is defined to be* $\Delta f = \sup_{\mathbf{x},\mathbf{x}':h(\mathbf{x},\mathbf{x}')=1} |f(\mathbf{x}) - f(\mathbf{x}')|$.

The idea of adding calibrated noise to satisfy ε-differential privacy is captured by the Laplace mechanism. In the Laplace mechanism, noise added is calibrated according to the sensitivity of f and sampled from a Laplace distribution. A Laplace distribution with mean 0 and scale $b > 0$ has density

$$h_b(y) = \frac{1}{2b} e^{-\frac{|y|}{b}} .$$

We denote by $\text{Lap}(b)$ a Laplace distribution with mean 0 and variance $2b^2$. For brevity, we will sometimes denote $X \sim \text{Lap}(b)$ as simply $\text{Lap}(b)$. We next define the Laplace mechanism.

Definition 7. *(Laplace Mechanism (Dwork, 2006; Dwork, McSherry, et al., 2006)). Let* $f : \mathcal{X} \to \mathbb{R}$ *be some function with sensitivity* Δf*. For the function* f*, a dataset* $\mathbf{x} \in \mathcal{X}$*, and the privacy parameter* $\varepsilon > 0$*, the Laplace mechanism* $\mathcal{A}_{f,\varepsilon}$ *is given by the pair* $(\mathcal{A}_{f,\varepsilon}, \text{Lap}(\Delta f/\varepsilon))$ *where* $\mathcal{A}_{f,\varepsilon} : \mathcal{X}_0^n \times \mathbb{R} \to \mathbb{R}$ *is defined as*

$$\mathcal{A}_{f,\varepsilon}(\mathbf{x}, z) = f(\mathbf{x}) + z \qquad (\mathbf{x} \in \mathcal{X}_0^n, z \in \mathbb{R}) . \tag{2.4}$$

$\mathcal{A}_{f,\varepsilon}$ *takes as input a datasaet* $\mathbf{x}$ *and a random variable* $Z \sim \text{Lap}(\Delta f/\varepsilon)$ *and adds* Z *to* $f(\mathbf{x})$.

Theorem 6. *(Dwork and Roth, 2014). The Laplace mechanism given by Definition 7 is* ε*-differentially private.*

Proof. Fix $f : \mathcal{X}_0^n \to \mathbb{R}$, the neighboring datasets $\mathbf{x}$ and $\mathbf{x}'$. Let $Z \sim \text{Lap}(\Delta f/\varepsilon)$. Let $p_{\mathbf{x}}$ and $p_{\mathbf{x}'}$ be the density functions of $\mathcal{A}_{f,\varepsilon}(\mathbf{x}, Z)$ and $\mathcal{A}_{f,\varepsilon}(\mathbf{x}', Z)$, respectively. Take any real

$z \in \mathbb{R}$. Then we have

$$\begin{aligned}
\frac{p_{\mathbf{x}}(z)}{p_{\mathbf{x}'}(z)} &= \frac{\exp\left(-\frac{\varepsilon}{\Delta f}|z - f(\mathbf{x})|\right)}{\exp\left(-\frac{\varepsilon}{\Delta f}|z - f(\mathbf{x}')|\right)} \\
&= \exp\left(\frac{\varepsilon(|z - f(\mathbf{x}')| - |z - f(\mathbf{x})|)}{\Delta f}\right) \\
&\leq \exp\left(\frac{\varepsilon \cdot |f(\mathbf{x}') - f(\mathbf{x})|}{\Delta f}\right) \\
&\leq \exp(\varepsilon)\,,
\end{aligned}$$

where the first inequality follows from the triangle inequality and the second inequality follows from the definition of sensitivity. Now, if the ratio of the densities is bounded by e^{ε} for any $z \in \mathbb{R}$, it is easy to see that the ratio of the output probability distributions for any $S \subseteq \mathbb{R}$ is also bounded by e^{ε}. ■

The next theorem reformulates (Dwork, 2006, Theorem 5). We also include a concise proof. We will use this theorem to prove the privacy guarantees of our algorithms (Algorithm 2 and Algorithm 5) in Chapter 4. To introduce the algorithm we let $\mathrm{Lap}_p(\sigma)$ denote the Laplace distribution for p-dimensional vectors with scale σ. The density of this distribution at $z \in \mathbb{R}^p$ is $h_\sigma(z_1) \dots h_\sigma(z_p)$ (and in particular, the components of a p-dimensional Laplace distribution are independent of each other).

Theorem 7. *Let $f_1 : \mathcal{X}_0^n \to \mathbb{R}^p$, $f_2 : \mathcal{X}_0^n \times \mathbb{R}^p \to \mathbb{R}^p$, …, $f_d : \mathcal{X}_0^n \times (\mathbb{R}^p)^{d-1} \to \mathbb{R}^p$ for some positive integers p and d. Let F be defined by*

$$\begin{aligned}
R_1 &= f_1(x) + Y_1 \\
R_2 &= f_2(x, R_1) + Y_2 \\
&\vdots \\
R_d &= f_d(x, R_1, \dots, R_{d-1}) + Y_d\,,
\end{aligned}$$

where $x \in \mathcal{X}_0^n$ and $Y_1, \dots, Y_d \sim \mathrm{Lap}_p(\sigma)$ are $\mathbb{R}^p$-valued and are independent of each other.[2]

[2]Formally, the mechanism is $(F, \mathrm{Lap}_p^{\otimes d}(\sigma))$, where $F : \mathcal{X} \times P^d \to P^d$ where $P = \mathbb{R}^p$ and for $x \in \mathcal{X}$, $y = (y_1, \dots, y_d) \in P^d$, $F(x,y)_1 = f_1(x) + y_1$, $F(x,y)_2 = f_2(x, F(x,y)_1) + y_2$, …, $F(x,y)_d = f_d(x, F(x,y)_{d-1}) + y_d$.

Then F is ε-differentially private with $\varepsilon = \sup_{z\in\mathbb{R}^{d-1}} \Delta\tilde{f}_z/\sigma$, where

$$\tilde{f}_z(x) = (f_1(x), f_2(x, z_1), \ldots, f_d(x, z_1, \ldots, z_{d-1}))$$

and for $f : \mathcal{X}_0^n \to \mathbb{R}^d$, $\Delta f = \sup_{x,x':h(x,x')=1} \|f(x) - f(x')\|_1$ is the sensitivity of f.

Proof. Let p_x be the density of $F(x,\cdot) \in \mathbb{R}^d$ and let h_σ be the density of all the Y_i's. Note that $h_\sigma(z) \propto e^{-\frac{\|z\|_1}{\sigma}}$. Since $(Y_1, \ldots, Y_d)$ are independent of each other, by the chain rule for densities, we have

$$p_x(z_1, \ldots, z_d) = h_\sigma(z_1 - f_1(x)) \cdot h_\sigma(z_2 - f_2(x, z_1)) \cdot \ldots \cdot h_\sigma(z_d - f_d(x, z_1, \ldots, z_{d-1})).$$

Hence, for two neighboring datasets $x, x' \in \mathcal{X}_0^n$, we have

$$\begin{aligned}
\frac{p_x(z)}{p_{x'}(z)} &= \frac{\exp\left(-\frac{1}{\sigma}\left(\|z_1 - f_1(x)\|_1 + \|z_2 - f_2(x, z_1)\|_1 + \ldots\right)\right)}{\exp\left(-\frac{1}{\sigma}\left(\|z_1 - f_1(x')\|_1 + \|z_2 - f_2(x', z_1)\|_1 + \ldots\right)\right)} \\
&= \exp\Bigg(\frac{1}{\sigma}(\|z_1 - f_1(x')\|_1 - \|z_1 - f_1(x)\|_1 \\
&\qquad + \|z_2 - f_2(x', z_1)\|_1 - \|z_2 - f_2(x, z_2)\|_1 + \ldots)\Bigg) \\
&\leq \exp\left(\frac{1}{\sigma}\left(\|f_1(x) - f_1(x')\|_1 + \|f_2(x, z_1) - f_2(x', z_1)\|_1 + \ldots\right)\right) \\
&\leq \exp\left(\frac{\Delta\tilde{f}_{\,z_1,\ldots,z_{d-1}}}{\sigma}\right) \leq \exp\left(\frac{\sup_{z\in\mathbb{R}^{d-1}} \Delta\tilde{f}_z}{\sigma}\right),
\end{aligned}$$

finishing the proof. ■

2.2.3 Differential Privacy in Matroid Bandits

First, let us define a matroid bandit algorithm.

Definition 8. *(Matroid bandit algorithm). Let $A_{\mathcal{M}}$ be the set of all bases of a matroid $\mathcal{M}$. Let $\mathcal{W}_K := [0,1]^K$, where K is the total number of base arms. A matroid bandit algorithm is a map $\mathcal{B} : \left(\bigcup_{t\in[n-1]} A_{\mathcal{M}}^t \times \mathcal{W}_K^t\right) \times [0,1] \to A_{\mathcal{M}}$.*

In applications, the data that needs to be protected is $\mathcal{W}_L^n$ since the rewards w_t come from users. For example, in movie-recommendation, if movie e is recommended to a user in

round t, then the feedback $w_t(e)$ could be whether the user has watched the movie or not, which we want to keep private. An adversary can try to reconstruct a particular user's data using the output of a matroid bandit algorithm, which is a super arm. Protection from such adversaries can be provided by making the the mechanism that maps user data to super arms differentially private. So we restate the definition of differential privacy for matroid bandits in terms of such a map.

Definition 9. *(Differential privacy for matroid bandits). Fix $\varepsilon > 0$. Let $\mathcal{W}_L := [0,1]^L$, where L is the total number of base arms and recall that $A_{\mathcal{M}}$ is the set of all bases of a matroid $\mathcal{M}$. Further, for $a \subset [L]$ and $\mathbf{w} \in \mathcal{W}_L$, let $\mathbf{w}(a) := (w_{a_1}, \ldots, w_{a_k})$ with $a_1 < \ldots < a_k$.*

We call a matroid bandit algorithm $\mathcal{B}$ ε-differentially private if $\tilde{\mathcal{B}} : \mathcal{W}_L^n \times [0,1] \to A_{\mathcal{M}}^n$ defined by $\tilde{\mathcal{B}}(\mathbf{w}_1, \ldots, \mathbf{w}_n, U) = (a_1, \ldots, a_n)$ where

$$
\begin{aligned}
a_1 &= \mathcal{B}(U) \\
a_2 &= \mathcal{B}(a_1, \mathbf{w}_1(a_1), U) \\
&\vdots \\
a_n &= \mathcal{B}(a_1, \mathbf{w}_1(a_1), \ldots, a_{n-1}, \mathbf{w}_{n-1}(a_{n-1}), U)\,,
\end{aligned}
$$

is ε-differentially private, where $U \sim \mathrm{Unif}[0,1]$.

In the matroid bandit setting, each reward vector $\mathbf{w} \in \mathcal{W}_L$ can encode private information associated with an individual, which we wish to protect. Definition 9 expresses the view that the data to be protected is $(\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathcal{W}_L^n$. A dataset for a matroid bandit algorithm can be viewed as a sequence of reward vectors $(\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathcal{W}_L^n$ which is fed into $\tilde{\mathcal{B}}$ as defined in Definition 9. Recalling the definition of neighboring datasets, two reward sequences $\mathbf{w}_{1:n} \in \mathcal{W}_L^n$ and $\mathbf{w}'_{1:n} \in \mathcal{W}_L^n$ are called neighboring if the Hamming distance between them is 1, that is, $h(\mathbf{w}_{1:n}, \mathbf{w}'_{1:n}) = 1$. More concretely, $\mathbf{w}_{1:n}$ and $\mathbf{w}'_{1:n}$ are neighboring if they differ in one round $r \in [n]$. That is, $\mathbf{w}_r \neq \mathbf{w}'_r$ and $\mathbf{w}_s = \mathbf{w}'_s$ for all $s \in [n] \setminus \{r\}$.

Chapter 3

Related Works

Kveton, Wen, Ashkan, Eydgahi, et al. (2014) initiated the study of learning the maximum weight basis for matroid bandits. They proposed a UCB-based algorithm, Optimistic Matroid Maximization (OMM), that achieves the optimal $O(L\ln(n)/\Delta)$ regret bound. The key idea in OMM is to construct upper confidence bounds based on UCB1 of Auer et al. (2002) for all base arms. As OMM is built upon UCB1, OMM cannot be asymptotically optimal. Later, Talebi et al. (2016) proposed another UCB-based algorithm, Efficient Sampling for Matroids (KL-OSM), that achieves the asymptotically optimal $L\ln(n)\Delta/d_{\mathrm{KL}}(\mu_*-\Delta,\mu_*)+o(\ln(n))$ regret bound, where μ_* denotes the mean reward of an optimal base arm and $d_{\mathrm{KL}}(a,b)$ denotes the KL-divergence between two Bernoulli distributions with parameters $a,b\in(0,1)$. The key idea to achieve asymptotic optimality is to construct upper confidence bounds based on KL-UCB of Garivier et al. (2011). Other than the aforementioned UCB-based algorithms, Wang et al. (2018) proposed Combinatorial Thompson Sampling (CTS), a Thompson Sampling-based algorithm with Beta priors, for combinatorial bandits. Combinatorial bandits generalize the setting of matroid bandits with the following key features. In combinatorial bandits, the set of super arms does not have any special structure that we can utilize. Also, the size of a super arm is at most K instead of exactly K. Since matroid bandits are special cases of combinatorial bandits, by a refined regret analysis, CTS achieves an $O(L\ln(n)/\Delta)+O\left(L/\Delta^4\right)$ regret for matroid bandits.

Mishra et al. (2015) initiated the study of stochastic multi-armed bandits with differential

privacy and proposed both the UCB-based and Thompson Sampling-based learning algorithms. Their proposed algorithms rely on the post-processing property of differential privacy (Lemma 2). More specifically, they first guarantee that the internal learning algorithm computing the empirical means is ε-differentially private (ε-DP). Then, from post-processing, they immediately conclude that the proposed bandit algorithms are ε-DP. However, their proposed learning algorithms are very sub-optimal due to the usage of the Tree-based Mechanism (Chan et al., 2011; Dwork, Naor, et al., 2010) to inject noise to preserve privacy. Later, Chen et al. (2020) introduced differential privacy in combinatorial bandits and proposed a UCB-based algorithm, Differentially Private Combinatorial UCB (CUCB-DP), for differentially private combinatorial bandits. CUCB-DP achieves an $O\left(LK\ln^2(n)/\Delta\right) + \tilde{O}\left(LK\ln^3(n)/\varepsilon\right)$ regret upper bound and an $\Omega\left(LK\ln(n)/\Delta + LK\ln(n)/\varepsilon\right)$ regret lower bound.[1]

Recently, Hu, Huang, et al. (2021) and Azize et al. (2022) devised optimal UCB-based algorithms and Hu and Hegde (2022) devised an optimal Thompson Sampling-based algorithm with Beta priors for differentially private stochastic bandits. These algorithms all achieve the optimal $O\left(L\ln(n)/\Delta + L\ln(n)/\varepsilon\right)$ regret bound. The key ideas to achieve optimality are the usage of "laziness" and "forgetfulness" along with the Laplace Mechanism (Theorem 6) to inject noise to mask the true empirical means instead of using the Tree-based Mechanism. The idea of laziness is to update the differentially private empirical mean of an arm in a delayed manner. We only update the DP empirical mean of an arm when a certain number of observations are available from that arm. The idea of forgetfulness is to use fresh observations to update the DP empirical mean. Once observations have been used, we abandon them. By the definition of DP, since the change of one reward vector only impacts the aggregated reward by at most one, from the Laplace Mechanism, we can add a noise drawn from $\text{Lap}\left(1/\varepsilon\right)$ to the aggregated reward of each arm when updating the DP empirical mean.

Our proposed UCB-based algorithm `DPUCB-MAT` (Algorithm 2) can be viewed as a differentially private version of OMM of Kveton, Wen, Ashkan, Eydgahi, et al. (2014). When

[1] The $\tilde{O}(\cdot)$ notation hides an extra $\log\log n$ factor.

$\varepsilon \to \infty$, i.e., in the non-private matroid bandit setting, the regret bound of `DPUCB-MAT` (Theorem 9) recovers the regret bound of OMM. With a suitable privacy parameter ε, our regret bound removes an extra $\log(n)$ factor as compared to the regret bound shown by Chen et al. (2020). Although our proposed Thompson Sampling-based algorithm `DPTS-MAT` (Algorithm 5) could be seen as a differentially private version of CTS (Wang et al., 2018) for matroid bandits, we use different proof techniques. Section 4.1.3 presents more detail. Table 3.1 summarizes the regret bounds for matroid bandits in both the non-private and the private settings. A point to note here is that the regret bound for the TS-based algorithm in the non-private setting in Table 3.1 is for Beta priors whereas our private result is for Gaussian priors.

Algorithms	Non-private results	Private results (ours)
UCB1-based	$O\left(\frac{L\ln(n)}{\Delta}\right)$	$O\left(\frac{L\ln(n)}{\Delta} + \frac{\min\{K,\log(n)\}LK\ln(n)}{\varepsilon}\right)$
TS-based	$O\left(\frac{L\ln(n)}{\Delta} + \frac{L}{\Delta^4}\right)$	$O\left(\frac{L\ln(n)}{\Delta} + \frac{\min\{K,\log(n)\}LK\ln(n)}{\varepsilon}\right)$

Table 3.1: Regret upper bounds for UCB and TS-based algorithms for matroid bandits.

Chapter 4

Algorithms and Results

In this chapter, we present our algorithms and their theoretical guarantees. We first start with some notation used by our algorithms.

Notation. We denote by $T_e(t-1)$ the "effective" number of observations used to compute the empirical mean of a base arm $e \in E$ by the end of round $t-1$ and denote by $\hat{w}_{e,T_e(t-1)}(t-1)$ the empirical mean of a base arm e using these $T_e(t-1)$ observations. Let $\varepsilon_0 := \varepsilon/K$. Let $\tilde{w}_{e,T_e(t-1)}(t-1) = \hat{w}_{e,T_e(t-1)}(t-1) + \frac{\text{Lap}(1/\varepsilon_0)}{T_e(t-1)}$ denote the differentially private (DP) mean estimate of a base arm e, where $\text{Lap}(1/\varepsilon_0)$ is a random variable sampled from a Laplace distribution with scale $1/\varepsilon_0$. For each base arm e, we store the reward observations in a list $\mathcal{T}_e$, which is initialized as an empty set. We also have a counter s_e which is incremented by 1 every time we update the differentially private mean estimate $\tilde{w}_{e,T_e(t-1)}(t-1)$.

For the regret bounds presentation, we also need the following notation. Let $\bar{A}^* = E \setminus A^*$ denote the set of all sub-optimal base arms. Let $\Delta_{e,k} = \bar{w}(a_k^*) - \bar{w}(e)$ denote the mean reward gap between a base arm $e \in E$ and an optimal base arm $a_k^* \in A^*$. For a sub-optimal base arm $e \in \bar{A}^*$, let $\Delta_{e,\min} = \min_{k\in[K]} \Delta_{e,k}$ denote the minimum mean reward gap.

4.1 Algorithms

Now we introduce our two algorithms. Our algorithms are based on the popular ideas of Upper Confidence Bound and Thompson Sampling. For both algorithms, we use the Laplace mechanism (Definition 7) to add noise and ensure differential privacy. To have a differentially

private version of $\hat{w}_{e,T_e(t-1)}(t-1)$, both the algorithms employ the ideas of "laziness" and "forgetfulness" introduced in Hu, Huang, et al. (2021), Azize et al. (2022), and Hu and Hegde (2022). Using the idea of laziness, we do not update the differentially private mean estimates $\tilde{w}_{e,T_e(t-1)}(t-1)$ after every round. Instead, we update it after every 2^k number of rounds for $k = 0, 1, \ldots$. Using the idea of forgetfulness, we do not reuse the observations after using them once to update $\tilde{w}_{e,T_e(t-1)}(t-1)$. Both the ideas of laziness and forgetfulness help in ensuring that we do not add too much noise in the mean estimates. The noise to be added during each update of $\tilde{w}_{e,T_e(t-1)}(t-1)$ is a random variable sampled from the Laplace distribution as described in Chapter 2. The Laplace noise added has a scale of $1/\varepsilon_0$, where $\varepsilon_0 = \varepsilon/K$. We explain later why we need to calibrate the noise in this way. Both our algorithms also pay a regret of LK for the first L rounds. This can be slightly improved by using the algorithm in Edmonds (1965). However, the extra LK regret is a lower order term in our bounds hence the improved initialization is not required in our case.

4.1.1 Differentially Private UCB for Matroid Bandits

The central idea of UCB-based algorithms is to construct confidence bounds around the mean estimates. Then the action with the largest upper bound on the confidence intervals is selected. This ensures sufficient exploration of the arms in the algorithm.

The UCB-based differentially private algorithm for matroid bandits, `DPUCB-MAT`, is shown in Algorithm 2. The idea is to construct *differentially private upper confidence bound* $U_t(e)$ (Line 5) for each base arm $e \in E$. We construct $U_t(e)$ as

$$U_t(e) := \tilde{w}_{e,T_e(t-1)}(t-1) + \sqrt{\frac{3\ln(Kt)}{T_e(t-1)}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)} \quad . \tag{4.1}$$

We set $U_t(e) := \infty$ when $T_e(t-1) = 0$. After we have all these private upper confidence bounds $U_t(e)$ in hand, `DPUCB-MAT` selects the best super arm A^t in a greedy way, i.e., invoking Algorithm 1 with all $U_t(e)$ as input and A^t as output. In other words, `DPUCB-MAT` plays $A^t = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} U_t(e)$. Then, the rewards $w_t(e)$ for $e \in A^t$ are revealed. For each $w_t(e)$, we add it to the corresponding $\mathcal{T}_e$ (Line 9). If the number of observations of any base

arm e hits 2^{s_e+1}, then, it is the right time to update the DP mean estimate $\tilde{w}_{e,T_e(t-1)}(t-1)$ using the 2^{s_e+1} observations stored in $\mathcal{T}_e$ (Line 12). Since now all the observations in $\mathcal{T}_e$ have been processed, we increment the counter s_e by 1 and reset $\mathcal{T}_e$ (Line 13).

Note that when any of the $U_t(e)$'s are ∞, they are placed at the end after the sorting in Algorithm 1. If there are multiple ∞ values, ties are broken randomly.

Algorithm 2 DPUCB-MAT

1: **Input**: Matroid $(E, \mathcal{I})$ and privacy parameter ε
2: Set $T_e \leftarrow 0, s_e \leftarrow -1, \tilde{w}_{e,T_e} \leftarrow 0, \mathcal{T}_e \leftarrow ()$, where $\varepsilon_0 = \varepsilon/K$ ▷ Initialization
3: **for** $t = 1, 2, \ldots$ **do**
4: **for** $e \in E$ **do**
5: $U_t(e) := \tilde{w}_{e,T_e} + \sqrt{\frac{3\ln(Kt)}{T_e}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e}$ ▷ Set to ∞ when $T_e = 0$
6: **end for**
7: Invoke Algorithm 1 with $U_t(e)$ for all $e \in E$ as input and A^t as output
8: Incur reward for choosing super arm A^t
9: Add $w_t(e)$ to $\mathcal{T}_e$ for all $e \in A^t$
10: Find $B^t = \{e \in A^t : |\mathcal{T}_e| = 2^{s_e+1}\}$ ▷ Find all the base arms with the number of observations hitting 2^{s_e+1}
11: **for** $e \in B^t$ **do**
12: $T_e \leftarrow 2^{s_e+1}$, $\tilde{w}_{e,T_e} = \frac{\left(\sum_{s \in \mathcal{T}_e} s\right) + \mathrm{Lap}(1/\varepsilon_0)}{T_e}$
13: $s_e \leftarrow s_e + 1$, $\mathcal{T}_e \leftarrow ()$ ▷ Doubling the "effective" number of observations and reset $\mathcal{T}_e$
14: **end for**
15: **end for**

We now present theoretical guarantee for Algorithm 2. We first provide the privacy guarantees for Algorithm 2.

Theorem 8. *Algorithm 2 is (2ε)-differentially private.*

Proof. We use Theorem 7 and the postprocessing property (Lemma 2) to show that Algorithm 2 is differentially private. First, we express the algorithm in the notation of Theorem 7.

Let $\mathcal{W}_L = [0,1]^L$, where L is the number of base arms. We let $\mathcal{W} = \mathcal{W}_L^n$. We define the functions $f_1 : \mathcal{W} \to \mathbb{R}^L$, $f_2 : \mathcal{W} \times \mathbb{R}^L \to \mathbb{R}^L$, $\ldots$, $f_i : \mathcal{W} \times (\mathbb{R}^L)^{i-1} \to \mathbb{R}^L$, $\ldots$, $f_n : \mathcal{W} \times (\mathbb{R}^L)^{n-1} \to \mathbb{R}^L$ as follows: Fix $i \in [n]$, $\mathbf{w} \in \mathcal{W}$, $z_i \in (\mathbb{R}^L)^{i-1}$. Here, recalling Theorem 7, z_i encodes all the private information created in previous steps of the algorithm

and $f_i(\mathbf{w}, z_i)$ is the information to be protected in step i. In our case, the private information is going to be the sums computed in Line 12. To define the function for all base arms e, the function f_i will not only store these sums but also return zero at coordinates which correspond to base arms $e \in [L]$ whose private information is not updated in round i.

Algorithm 3 shows how the functions f_t's can be computed in the context of Algorithm 2: lines in red show the difference compared to Algorithm 2 and lines in blue show how the output f_{t_*} is calculated for some t_*. Apart from the matroid and ε, Algorithm 3 takes as input the target time t_* for which we want to calculate f, the user data $w = (w_1, \ldots, w_n)$, and function values $z = (z_1, \cdots, z_{t_*-1})$ for previous time steps. The change on Line 9 emphasizes the fact that user data is not accessed on Line 9 but only on Line 12, where the sum $\sum_{t \in \mathcal{T}_e} w_t(e)$ is calculated. On Line 15 - Line 18, we are just calculating dummy outputs so that the f_t is well-defined for all $e \in E$. Further, κ_e is the last time when the stats of e was updated.

We also need the function $a_i(z)$ that computes the super-arm to be recommended in round i (line 7). That this super-arm can be computed based on z only follows by construction, again. For $z = (z_1, \ldots, z_{n-1}) \in (\mathbb{R}^L)^{n-1}$ define

$$a^*(z) = (a_1, a_2(z_1), \ldots, a_n(z_1, \ldots, z_{n-1}))\,.$$

Thus, $a^* : (\mathbb{R}^L)^{n-1} \to (2^{[L]})^n$.

Now, to express our algorithm in the notation of Theorem 7, we also need to compute noisy sums

$$\begin{aligned}
R_1 &= f_1(\mathbf{w}) + Y_1 \\
R_2 &= f_2(\mathbf{w}, R_1) + Y_2 \\
&\vdots \\
R_n &= f_n(\mathbf{w}, R_1, \ldots, R_{n-1}) + Y_n\,,
\end{aligned}$$

where $Y_1, \ldots, Y_n \in \mathbb{R}^L$ are independent, with $Y_i \in \mathrm{Lap}(1/\varepsilon_0)$ (each component of Y_i is sampled from $\mathrm{Lap}(1/\varepsilon_0)$). We show this computation in Algorithm 4, which outputs a

sequence $(r_{t,e})_{t\in[n],e\in[L]}$ of noisy sums. Again, the changes compared to Algorithm 2 are in red and how the output is computed is in blue.

Note that for some t_* and $e \in B^{t_*}$, $r_{t_*,e} = f_{t_*}(w; z_{1:t_*-1}) + (y_t)_e$, where f_{t_*} is the output from Algorithm 3. That is, Algorithm 4 indeed computes noisy sums as desired. Since user data is accessed only on Line 12 in Algorithm 4, for computing private statistics we only care about values of $r_{t_*,e}$ for $e \in B^{t_*}$, that is, base arms whose statistics are updated in time step t_*. So for fixed user data $w = (w_1, \ldots, w_n)$, if $e \in B^{t_*}$, then the sum $\sum_{s\in\mathcal{T}_e} s$ computed on Line 12 is the same as $f_{t_*}(w; z_{1:t-1})$. Therefore, for fixed user data w and a fixed noise sequence $y_1, \ldots, y_n$, we have $r_{t_*,e} = f_{t_*}(w; z_{1:t_*-1}) + (y_t)_e$.

Further, note that that for fixed user data $w = (w_1, \ldots, w_n)$ and noise sequence $y_1, \cdots, y_n$, the super arms chosen by Algorithm 4 is the same as Algorithm 2. Let Algorithm 2 also take as input the same noise sequence $y_1, \ldots, y_n$ as Algorithm 4, that is, $\mathrm{Lap}(1/\varepsilon_0)$ on Line 12 in Algorithm 2 is $(y_t)_e$ for some t and e. Again, since the user data is accessed only on Line 12, and the noise sequence is same for both Algorithm 2 and Algorithm 4, the UCBs computed on Line 5 will be the same for both the algorithms and therefore, the super arms chosen on Line 7 will also be the same.

It follows that, with the user weights fixed to $\mathbf{w}$, running Algorithm 2 is equivalent to computing $R_1, \ldots, R_n$ as defined above and then returning $a^*(R_1, \ldots, R_{n-1})$.

Now, by Theorem 7, the mechanism $\mathbf{w} \mapsto (R_1, \ldots, R_n)$ is κ-private with $\kappa = \Delta^*\varepsilon_0 = \Delta^*\varepsilon/K$ where

$$\Delta^* = \sup_{z\in(\mathbb{R}^L)^{n-1}} \Delta\tilde{f}_z$$

and for $z = (z_1, \ldots, z_{n-1}) \in (\mathbb{R}^L)^{n-1}$,

$$\tilde{f}_z(\mathbf{w}) = (f_1(\mathbf{w}), f_2(\mathbf{w}, z_1), \ldots, f_n(\mathbf{w}, z_1, \ldots, z_{n-1}))$$

while for $f : \mathcal{W}_L^n \to (\mathbb{R}^L)^{n-1}$, $\Delta f = \sup_{\mathbf{w},\mathbf{w}'\in\mathcal{W}_L^n : h(\mathbf{w},\mathbf{w}')=1} \|f(\mathbf{w}) - f(\mathbf{w}')\|_1$. By the postprocessing lemma (Lemma 2), the mechanism $\mathbf{w} \mapsto a^*(R_1, \ldots, R_{n-1})$ is also κ-private and thus the result follows if we show that $\Delta^* \le 2K$.

To upper bound Δ^*, fix $z \in (\mathbb{R}^L)^{n-1}$ and let $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ be neighbours. In particular, assume that $\mathbf{w}, \mathbf{w}'$ differ only at their i-th component: $w_i \neq w'_i$ and $w_j = w'_j$ for $j \neq i$. Then,

$$\begin{aligned}\|\tilde{f}_z(\mathbf{w}) - \tilde{f}_z(\mathbf{w}')\|_1 = \|f_1(\mathbf{w}) - f_1(\mathbf{w}')\|_1 + \|f_2(\mathbf{w}, z_1) - f_2(\mathbf{w}', z_1)\|_1 \\ + \dots \\ + \|f_n(\mathbf{w}, z_{1:n-1}) - f_n(\mathbf{w}', z_{1:n-1})\|_1 \,,\end{aligned}$$

where we introduced the shorthand $z_{1:i} = (z_1, \dots, z_i)$.

Since $\mathbf{w}$ and $\mathbf{w}'$ differ in their ith component, and this is not accessed until round i, we have

$$f_1(\mathbf{w}) = f_1(\mathbf{w}'), \dots, f_{i-1}(\mathbf{w}, z_{1:i-2}) = f_{i-1}(\mathbf{w}', z_{1:i-2})\,. \tag{4.2}$$

Now, the super-arm chosen in round i is $a_i(z_{1:i-1})$, by the definition of a_i. Let $e_1, \dots, e_K$ denote the elements of this super-arm: $a_i(z_{1:i-1}) = \{e_1, \dots, e_K\}$. It follows that this choice is independent of the weights of user i. Let $\tau_1(\mathbf{w}, z_{1:n}), \dots, \tau_K(\mathbf{w}, z_{1:n})$ be the first rounds after round i when $w_i(e_1), \dots, w_i(e_K)$ are accessed in line 12.

Let $i \leq t \leq n$ be such that $t \notin \{\tau_1(\mathbf{w}, z), \dots, \tau_K(\mathbf{w}, z), \tau_1(\mathbf{w}', z), \dots, \tau_K(\mathbf{w}', z)\}$. For such a t, we have

$$f_t(\mathbf{w}, z_{1:t-1}) = f_t(\mathbf{w}', z_{1:t-1})\,.$$

Now consider the case when $t \in \{\tau_1(\mathbf{w}, z), \dots, \tau_K(\mathbf{w}, z), \tau_1(\mathbf{w}'z), \dots, \tau_K(\mathbf{w}', z)\}$. If $e \notin \{e_i : t = \tau_i(\mathbf{w}, z) \text{ or } t = \tau_i(\mathbf{w}', z) \text{ for } i \in [K]\}$, then

$$(f_t(\mathbf{w}, z_{1:t-1}))_e = (f_t(\mathbf{w}', z_{1:t-1}))_e\,.$$

Otherwise,

$$|(f_t(\mathbf{w}, z_{1:t-1}))_e - (f_t(\mathbf{w}', z_{1:t-1}))_e| \leq 1$$

since $w_i(e) \in [0, 1]$. Hence,

$$\begin{aligned}\|\tilde{f}_z(\mathbf{w}) - \tilde{f}_z(\mathbf{w}')\|_1 &= \sum_{t=i}^{n} \|f_t(\mathbf{w}, z_{1:t-1}) - f_t(\mathbf{w}', z_{1:t-1})\|_1 \\ &\leq \sum_{t=i}^{n} |\,\{e_j : t = \tau_j(\mathbf{w}, z) \text{ or } t = \tau_j(\mathbf{w}', z) \text{ for } j \in [K]\}\,| \\ &\leq 2K\,.\end{aligned}$$

It follows that $\Delta\tilde{f}_z \leq 2K$ and since z is arbitrary, it also follows that $\Delta^* \leq 2K$, finishing the proof.

Algorithm 3 `DPUCB-MAT-for-DP-proof`

Input: Matroid $(E, \mathcal{I})$, privacy parameter ε,
target time $t_* \in [n]$, user data $w = (w_1, \ldots, w_n) \in ([0,1]^L)^n$,
function values for previous time steps $z = (z_1, \ldots, z_{t_*-1}) \in (\mathbb{R}^L)^{t_*-1}$.
Output: $f_{t_*}(w, z_1, \ldots, z_{t_*-1})$.
Set $T_e \leftarrow 0, s_e \leftarrow -1, \sout{\tilde{w}_{e,T_e} \leftarrow 0}$, $z_{0,e} = 0$, $\kappa_e = 0$, $\mathcal{T}_e \leftarrow ()$, where $\varepsilon_0 = \varepsilon/K$ ▷ Initialization
for $t = 1, 2, \ldots, t_*$ **do**
  for $e \in E$ **do**
    $U_t(e) := \sout{\tilde{w}_{e,T_e}}\ z_{\kappa_e,e} + \sqrt{\frac{3\ln(Kt)}{T_e}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e}$ ▷ Set to ∞ when $T_e = 0$
  end for
  Invoke Algorithm 1 with $U_t(e)$ for all $e \in E$ as input and A^t as output
  Incur reward for choosing super arm A^t
  ~~Add $w_t(e)$ to $\mathcal{T}_e$ for all $e \in A^t$~~
  Add t to $\mathcal{T}_e$ for all $e \in A^t$
  Find $B^t = \{e \in A^t : |\mathcal{T}_e| = 2^{s_e+1}\}$ ▷ Find all the base arms with the number of observations hitting 2^{s_e+1}
  for $e \in B^t$ **do**
    $T_e \leftarrow 2^{s_e+1}$, $\sout{\tilde{w}_{e,T_e} = \frac{\left(\sum_{s \in \mathcal{T}_e} s\right) + \mathrm{Lap}(1/\varepsilon_0)}{T_e}}$, $\kappa_e = t$
    if $t = t_*$ **then** $f_t(w; z_{1:t-1})_e = \sum_{i \in \mathcal{T}_e} w_i(e)$
    $s_e \leftarrow s_e + 1$, $\mathcal{T}_e \leftarrow ()$ ▷ Doubling the "effective" number of observations and reset $\mathcal{T}_e$
  end for
  if $t = t_*$ **then**
    for $e \in [L] \setminus B^t$ **do**
      $f_t(w; z_{1:t-1})_e = 0$
    end for
  end if
end for
return $f_{t_*}(w; z_{1:t_*-1})$

■

We remark that the algorithm can be made ε-differentially private by choosing ε_0 as $\varepsilon/2K$.

Now we show the regret guarantee for Algorithm 2.

Theorem 9. *The regret of Algorithm 2 is*

$$R_n = \sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K, \log(Kn)\} \cdot \ln(Kn)}{\varepsilon/K}\right), \tag{4.3}$$

Algorithm 4 `DPUCB-MAT-for-DP-proof-2`

Input: Matroid $(E, \mathcal{I})$ and privacy parameter ε
Noise sequence $y_1, \ldots, y_n \in \mathbb{R}^L$, user data $w = (w_1, \ldots, w_n) \in ([0,1]^L)^n$
Set $T_e \leftarrow 0, s_e \leftarrow -1, \tilde{w}_{e,T_e} \leftarrow 0, \mathcal{T}_e \leftarrow ()$, where $\varepsilon_0 = \varepsilon/K$ ▷ Initialization
for $t = 1, 2, \ldots$ **do**
  for $e \in E$ **do**
    $U_t(e) := \tilde{w}_{e,T_e} + \sqrt{\frac{3\ln(Kt)}{T_e}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e}$ ▷ Set to ∞ when $T_e = 0$
  end for
  Invoke Algorithm 1 with $U_t(e)$ for all $e \in E$ as input and A^t as output
  Incur reward for choosing super arm A^t
  Add $w_t(e)$ to $\mathcal{T}_e$ for all $e \in A^t$
  Find $B^t = \{e \in A^t : |\mathcal{T}_e| = 2^{s_e+1}\}$ ▷ Find all the base arms with the number of observations hitting 2^{s_e+1}
  for $e \in B^t$ **do**
    $T_e \leftarrow 2^{s_e+1}$, $\tilde{w}_{e,T_e} = \frac{\left(\sum_{s\in\mathcal{T}_e} s\right) + \cancel{\mathrm{Lap}(1/\varepsilon_0)}(y_t)_e}{T_e}$, $r_{t,e} = \left(\sum_{s\in\mathcal{T}_e} s\right) + (y_t)_e$
    $s_e \leftarrow s_e + 1$, $\mathcal{T}_e \leftarrow ()$ ▷ Doubling the "effective" number of observations and reset $\mathcal{T}_e$
  end for
end for
return $(r_{t,e})_{t\in[n],e\in[L]}$

where $\Delta_{e,\min} = \min_{k\in[K]:\Delta_{e,k}>0} \Delta_{e,k}$.

Discussion. Algorithm 2 can be viewed as a differentially private version of OMM of Kveton, Wen, Ashkan, Eydgahi, et al. (2014). When $\varepsilon \to \infty$, the differentially private matroid bandit problem boils down to the non-private matroid bandits. In this special setting, the regret bound shown in Theorem 9 matches both the regret upper and lower bounds presented in Kveton, Wen, Ashkan, Eydgahi, et al. (2014). This shows that we pay an additional $\frac{\min\{K,\log(Kn)\}\cdot\ln(Kn)}{\varepsilon/K}$ price for introducing differential privacy which goes to 0 as $\varepsilon \to \infty$. With a suitable privacy parameter ε, our regret bound improves the state-of-the-art regret bound presented in Chen et al. (2020) by removing an extra $\log(n)$ factor [1].

[1] For the discussion, we already translate their regret bound (Theorem 8) from combinatorial bandits to matroid bandits.

Algorithm 5 DPTS-MAT

Input: Matroid $M = (E, \mathcal{I})$ and privacy parameter ε
Set $T_e \leftarrow 0, s_e \leftarrow -1, \tilde{w}_{e,T_e} \leftarrow 0, \mathcal{T}_e \leftarrow ()$, where $\varepsilon_0 = \frac{\varepsilon}{K}$ ▷ Initialization
for $t = 1, 2, \ldots$ **do**
  for $e \in E$ **do**
    Set $w'_{e,T_e}(t) := \tilde{w}_{e,T_e} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e}$ ▷ Boost the parameters of the posterior distributions
    Sample $\theta_e(t) \sim \mathcal{N}\left(w'_{e,T_e}(t), \frac{1}{T_e}\right)$ ▷ Draw random samples from a Gaussian distribution
  end for
  Invoke Algorithm 1 with $\theta_e(t)$ for all $e \in E$ as input and A^t as output
  Incur reward for playing super arm A^t
  Add $w_t(e)$ to $\mathcal{T}_e$ for all $e \in A^t$
  Find $B^t = \{e \in A^t : |\mathcal{T}_e| = 2^{s_e+1}\}$ ▷ Find all the base arms with the number of observations hitting 2^{s_e+1}
  for $e \in B^t$ **do**
    $T_e \leftarrow 2^{s_e+1}$, $\tilde{w}_{e,T_e} = \frac{\left(\sum_{s \in \mathcal{T}_e} s\right) + \mathrm{Lap}(1/\varepsilon_0)}{T_e}$
    $s_e \leftarrow s_e + 1$, $\mathcal{T}_e \leftarrow ()$ ▷ Doubling the "effective" number of observations and reset $\mathcal{T}_e$
  end for
end for

4.1.2 Differentially Private Thompson Sampling for Matroid Bandits

Thompson Sampling-based algorithms assume a prior distribution on the unknown mean reward parameters of each arm. They also assume a likelihood function which is the probability of a reward given the mean reward parameter. The posterior is proportional to the prior times the likelihood. Gaussians prior and likelihood have the nice property that the posterior is also Gaussian. In each round, such algorithms sample a mean reward parameter from the posterior and pick the arm with the highest mean reward in the sample. As our goal is to design learning algorithms that have good regret guarantees with a finite time horizon, we can use Gaussian priors and likelihood[2].

Our Thompson Sampling differentially private algorithm for matroid bandits, DPTS-MAT, is shown in Algorithm 5. The general idea is to boost the parameter of the posterior distribution

[2]As proved in Agrawal et al. (2017), Beta-Bernoulli Thompson Sampling can be asymptotically optimal while Thompson Sampling with Gaussian prior and likelihood may not be asymptotically optimal.

from $\tilde{w}_{e,T_e(t-1)}(t-1)$ to $w'_{e,T_e(t-1)}(t)$, where $w'_{e,T_e(t-1)}(t) = \tilde{w}_{e,T_e(t-1)}(t-1) + 3\ln(Kt)/(\varepsilon_0 \cdot T_e(t-1))$ for each base arm $e \in E$. Then, DPTS-MAT draws a random sample $\theta_e(t) \sim \mathcal{N}\left(w'_{e,T_e(t-1)}(t), 1/T_e(t-1)\right)$ for all $e \in E$. We set $\theta_e(t) := \infty$ if $T_e(t-1) = 0$. After we have all these posterior samples $\theta_e(t)$ in hand, DPTS-MAT selects the best super arm A^t in a greedy way, i.e., invoking Algorithm 1 with all $\theta_e(t)$ as input and A^t as output. That is also to say, DPTS-MAT plays $A^t = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \theta_e(t)$. DPTS-MAT uses the same way as Algorithm 2 to process the revealed observations, i.e., it only updates the DP mean estimate of a base arm $e \in A^t$ if the number of observations in $\mathcal{T}_e$ hits 2^{s_e+1}.

Similar to Algorithm 2, when any of the $\theta_e(t)$'s are ∞, they are placed at the end after the sorting in Algorithm 1. If there are multiple ∞ values, ties are broken randomly. We now present theoretical guarantees for Algorithm 5.

Theorem 10. *Algorithm 5 is (2ε)-differentially private.*

We omit the proof of Theorem 10 as it is almost identical to Theorem 8.

Theorem 11. *The regret of Algorithm 5 is*

$$R_n = \sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K, \log(Kn)\} \cdot \ln(Kn)}{\varepsilon/K}\right) + \sum_{k \in [K] : \Delta_{\min,k} > 0} O\left(\frac{\ln(Kn)}{\Delta_{\min,k}}\right), \tag{4.4}$$

where $\Delta_{e,\min} = \min_{k \in [K] : \Delta_{e,k} > 0} \Delta_{e,k}$ *and* $\Delta_{\min,k} = \min_{e \in \bar{A}^* : \Delta_{e,k} > 0} \Delta_{e,k}$.

Discussion. Different from Theorem 9 where the regret bound only has one term which is linear in the size of the sub-optimal base arms, there are two terms in Theorem 11. The first term is the same as the regret bound shown in Theorem 9 and it captures the regret for introducing differential privacy. This term characterizes the regret in all the rounds when the posterior distributions of the sub-optimal base arms are not concentrated. It is not surprising that we have the second non-private term which is linear in the size of the optimal base arm set. The second term upper bounds the regret among all the rounds when the posterior distributions of the optimal base arms are not concentrated. As will be shown in Section 4.1.3, the core of our regret decomposition is to decompose a matroid bandit problem

into K stochastic bandit problems. For each of the K bandit problems, we get an additional regret of $O\left(\ln(n)/\Delta_{\min}\right)$ for all the rounds when the Gaussian posterior distributions of the optimal arm are not concentrated. To get this additional term, we modify the regret analysis in Agrawal et al. (2017) for the case when using Gaussian priors. In contrast, when using Beta priors, the additive term can be $\tilde{O}\left(1/\Delta_{\min}^4\right)$, where $\tilde{O}$ hides problem-dependent constants. Since the regret for a matroid bandit is composed of K different stochastic bandit problems, we have the second term in Theorem 11.

4.1.3 Regret Decomposition

In this section, we present an approach to decompose the regret of any index-based algorithm. For matroid bandits, an index-based algorithm computes some index $J_t(e)$ for all $e \in E$ then uses Algorithm 1 to compute A_t. Both `DPUCB-MAT` (Algorithm 2) and `DPTS-MAT` (Algorithm 5) are index-based. The core of regret decomposition is to introduce a round-dependent permutation π_t over $\{1, \ldots, K\}$. Using π_t, the regret can be decomposed into regret of K different stochastic bandit problems. The construction of π_t is inspired by Lemma 1 in Kveton, Wen, Ashkan, Eydgahi, et al. (2014).

Recall $A^t = \{a_1^t, a_2^t, \ldots, a_K^t\}$ with the base arm indices $J_t(a_1^t) \geq J_t(a_2^t) \geq \ldots \geq J_t(a_K^t)$. The indices $J_t(e)$ for $e \in E$ are UCBs in Algorithm 2 and posterior samples in Algorithm 5. Also recall $A^* = \{a_1^*, a_2^*, \ldots, a_K^*\}$ with $\bar{w}(a_1^*) \geq \bar{w}(a_2^*) \geq \ldots \geq \bar{w}(a_K^*)$. The purpose of introducing permutation $\pi_t : \{1, \ldots, K\} \to \{1, \ldots, K\}$ is to construct K ordered pairs between A^t and A^* so that we can get the regret of choosing a suboptimal base arm instead of an optimal one.

We construct π_t in a backward order as follows. We first focus on a_K^t. Fix $B_K = \left\{a_1^t, \ldots, a_{K-1}^t\right\}$. If $a_K^t \in A^*$, i.e., $a_K^t = a_i^*$ for some $i \in [K]$, we set $\pi_t(K) = i$, i.e., we pair a_K^t to itself. If $a_K^t \notin A^*$, due to the augmentation property of matroids (Property (3) in Section 2.1), we set $\pi_t(K) = \min\left\{i : a_i^* \in A^* \setminus B_K, B_K \cup \{a_i^*\} \in \mathcal{I}\right\}$, i.e., we can pair a_K^t with the optimal base arm with the smallest index that can be added to B_K to form a matroid basis. Now fix $B_{K-1} = \left\{a_1^t, \ldots, a_{K-2}^t, a_{\pi_t(K)}^*\right\}$. If $a_{K-1}^t = a_i^*$ for some $i \in [K]$, we set

$\pi_t(K-1) = i$. If $a^t_{K-1} \notin A^*$, we set $\pi_t(K-1) = \min\{i : a^*_i \in A^* \setminus B_{K-1}, B_{K-1} \cup \{a^*_i\} \in \mathcal{I}\}$. The same idea is applied to all the remaining base arms $a^t_{K-2}, \ldots, a^t_1$ in A^t.

The construction of π_t can be summarized as follows. For $k = K, K-1, \ldots, 2$:

1. $B_k = \left\{a^t_1, \ldots, a^t_{k-1}, a^*_{\pi_t(k+1)}, \ldots, a^*_{\pi_t(K)}\right\}$.

2. Let $j = \min\{i \in [K] : a^*_i \in A^* \setminus B_k, B_k \cup \{a^*_i\} \in \mathcal{I}\}$. Set

$$\pi_t(k) = \begin{cases} i, & \text{if } a^t_k = a^*_i \text{ for some } i \in [K] \textbf{ (case 1)} \\ j, & \text{otherwise } \textbf{(case 2)}. \end{cases} \tag{4.5}$$

Next we show that π_t is indeed a permutation on $[K]$. First, let $\tilde{B}_k := B_k \cup \left\{a^*_{\pi_t(k)}\right\}$. Note that $\tilde{B}_k \in \mathcal{I}$ by the augmentation property of matroids.

Claim 12. $|\tilde{B}_1| = |\tilde{B}_2| = \ldots = |\tilde{B}_K| = K$.

Proof. We show the claim by induction.

Base case: In the base case, we have $B_K = \left\{a^t_1, \ldots, a^t_{K-1}\right\}$ and $|B_K| = K-1$. If we are in **case 1** of 4.5, we have $a^t_K = a^*_i$ for some $i \in [K]$. Now, since $a^t_K \notin \left\{a^t_1, \ldots, a^t_{k-1}\right\}$ and $\tilde{B}_K = A^t$, we have $|\tilde{B}_K| = |A^t| = K$. If we are in **case 2** of 4.5, since $a^*_{\pi_t(K)} \notin B_K$, we have $|\tilde{B}_K| = \tilde{B_K} + 1 = K$.

Induction step: Assume that the claim holds for $k = K, K-1, \ldots, l+1$, i.e., $|\tilde{B}_K| = |\tilde{B}_{K-1}| = \ldots = |\tilde{B}_{l+1}| = K$ (induction hypothesis). We now show that it also holds for $k = l$. Recall $\tilde{B}_l = \left\{a^t_1, \ldots, a^t_{l-1}, a^*_{\pi_t(l+1)}, \ldots, a^*_{\pi_t(K)}\right\}$.

If we are in **case 1** of 4.5, we know that $a^t_l = a^*_i$ for some $i \in [K]$. Now we have, $\tilde{B}_l = B_l \cup \left\{a^*_{\pi_t(l)}\right\} = \tilde{B}_{l+1}$. Therefore by the induction hypothesis, $|\tilde{B}_l| = |\tilde{B}_{l+1}| = K$.

If we are in **case 2** of 4.5, we know that $\tilde{B}_l = B_l \cup \left\{a^*_{\pi_t(l)}\right\}$ and $a^*_{\pi_t(l)} \notin B_l$ by construction of π_t. Note that $B_l = \tilde{B}_{l+1} \setminus \left\{a^*_{\pi_t(l)}\right\}$. By the induction hypothesis, since $|\tilde{B}_{l+1}| = K$, we have $|B_l| = K-1$. Since $a^*_{\pi_t(l)} \notin B_l$, we have $|\tilde{B}_l| = K - 1 + 1 = K$.

■

Corollary 13. *π_t is a permutation on $[K]$.*

Proof. By Claim 12, we have $|\tilde{B}_1| = \left|\left\{a^*_{\pi_t(1)}, \ldots, a^*_{\pi_t(K)}\right\}\right| = K$. This implies that π_t is a permutation on $[K]$. ■

After applying π_t, all the optimal base arms in A^* will be ordered as

$$A^*_{\pi_t} = \left(a^*_{\pi_t(1)}, \ldots, a^*_{\pi_t(k)}, \ldots, a^*_{\pi_t(K)}\right)$$

Still, the order of all the base arms in A^t is $(a^t_1, \ldots, a^t_k, \ldots, a^t_K)$. Now, we can construct the following ordered pairs

$$\left(a^t_1, a^*_{\pi_t(1)}\right), \ldots, \left(a^t_k, a^*_{\pi_t(k)}\right), \ldots, \left(a^t_K, a^*_{\pi_t(K)}\right) .$$

It is not hard to verify that each $\left(a^t_k, a^*_{\pi_t(k)}\right)$ has the following properties.

1. If the selected base arm a^t_k in round t is an optimal base arm, i.e., $a^t_k \in A^*$, then its pair is itself which gives $\Delta_{a^t_k, \pi_t(k)} = 0$ (recall $\Delta_{e,k} = \bar{w}(a^*_k) - \bar{w}(e)$ for any two base arms a^*_k and e).

2. Given $\{a^t_1, a^t_2, \ldots, a^t_{k-1}\}$, both a^t_k and its pair $a^*_{\pi_t(k)}$ can be added to $\{a^t_1, a^t_2, \ldots, a^t_{k-1}\}$ without breaking matroid independence using the construction of π_t and the fact that a subset of an independent set is independent. In other words, both $\{a^t_1, \ldots, a^t_{k-1}, a^t_k\}$ and $\{a^t_1, \ldots, a^t_{k-1}, a^*_{\pi_t(k)}\}$ are in $\mathcal{I}$.

3. If the selected base arm a^t_k in round t is a suboptimal base arm, i.e., $a^t_k \notin A^*$, we have $J_t(a^t_k) \geq J_t(a^*_{\pi_t(k)})$. Intuitively, this is because if this was not the case then the super-arm selection algorithm, Algorithm 1, would have chosen $a^*_{\pi_t(k)}$ instead of a^t_k. Recall that J_t would be the UCBs or the posterior samples in round t for Algorithm 2 and Algorithm 5 respectively.

To permute $A^*_{\pi_t}$ back to the original order $(a^*_1, \ldots, a^*_K)$, we can use $\pi_t^{-1} : \{1, \ldots, K\} \to \{1, \ldots, K\}$, the inverse permutation of π_t. Note that for each $k \in [K]$, we have $\pi_t^{-1}(\pi_t(k)) = k$. We apply π_t^{-1} to permute the ordered sets $A^*_{\pi_t}$ and A^t separately. Applying π_t^{-1}

over $A^*_{\pi_t}$ gives us the original order, which is $(a^*_1, \ldots, a^*_K)$. Applying π_t^{-1} over A^t gives $\left(a^t_{\pi_t^{-1}(1)}, \ldots, a^t_{\pi_t^{-1}(k)}, \ldots, a^t_{\pi_t^{-1}(K)}\right)$. Based on the new order, we construct the following ordered pairs

$$\left(a^t_{\pi_t^{-1}(1)}, a^*_1\right), \ldots, \left(a^t_{\pi_t^{-1}(k)}, a^*_k\right), \ldots, \left(a^t_{\pi_t^{-1}(K)}, a^*_K\right) . \tag{4.6}$$

Since a single π_t^{-1} is used to permute both the ordered sets, π_t^{-1} can be viewed as a permutation that permutes

$$\left(a^t_1, a^*_{\pi_t(1)}\right), \ldots, \left(a^t_k, a^*_{\pi_t(k)}\right), \ldots, \left(a^t_K, a^*_{\pi_t(K)}\right)$$

to

$$\left(a^t_{\pi_t^{-1}(1)}, a^*_1\right), \ldots, \left(a^t_{\pi_t^{-1}(k)}, a^*_k\right), \ldots, \left(a^t_{\pi_t^{-1}(K)}, a^*_K\right) .$$

Recall $\bar{A}^* = E \setminus A^*$. Now, we are ready to decompose the regret:

$$\begin{aligned}
R_n &= \sum_{t=1}^{n} \mathbb{E}\left[\sum_{e \in A^*} \bar{w}(e) - \sum_{e \in A^t} \bar{w}(e)\right] \\
&\overset{(a)}{=} \sum_{t=1}^{n} \mathbb{E}\left[\sum_{k=1}^{K} \left(\bar{w}(a^*_k) - \bar{w}\left(a^t_{\pi_t^{-1}(k)}\right)\right)\right] \\
&\leq \sum_{k=1}^{K} \sum_{t=1}^{n} \mathbb{E}\left[\Delta_{a^t_{\pi_t^{-1}(k)},k} \cdot \mathbb{1}\left\{\Delta_{a^t_{\pi_t^{-1}(k)},k} > 0\right\}\right] \\
&= \sum_{k=1}^{K} \underbrace{\sum_{e \in \bar{A}^* : \Delta_{e,k} > 0} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right] \cdot \Delta_{e,k}}_{I_k} .
\end{aligned} \tag{4.7}$$

Equality (a) uses the ordered pairs shown in (4.6). Note that I_k can be viewed as the regret of a stochastic bandit problem with a^*_k as the optimal arm and $\left\{e \in \bar{A}^* : \Delta_{e,k} > 0\right\}$ as the set of sub-optimal arms. The regret bounds for both `DPUCB-MAT` (Theorem 9) and `DPTS-MAT` (Theorem 11) can be derived based on the regret decomposition shown in (4.7). We defer the details of the proof to Chapter 5.

Chapter 5

Regret Analysis

For the regret analysis, we will need the following concentration inequalities.

5.1 Concentration Inequalities

Lemma 14. *(Hoeffding's inequality). Let $X_1, \ldots, X_n$ be independent random variables with each $X_i \in [a_i, b_i]$. Then, for any $\epsilon > 0$, we have*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}X_i)\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right). \tag{5.1}$$

Lemma 15. *((Dwork and Roth, 2014, Fact 3.7); tail probability for Laplace distribution). If $Y \sim \mathrm{Lap}(b)$, for any $0 < \delta \leq 1$, we have*

$$\mathbb{P}\left\{|Y| \geq b\ln(1/\delta)\right\} = \delta\,. \tag{5.2}$$

Lemma 16. *(Gaussian tail bound). Let X be a Gaussian distributed random variable with mean $\mathbb{E}[X]$ and variance σ^2, then for any $t > 0$ we have*

$$\mathbb{P}\left\{X - \mathbb{E}[X] > t\right\} \leq e^{-\frac{t^2}{2\sigma^2}}\,. \tag{5.3}$$

In this chapter, we provide proofs for Theorem 9 and Theorem 11. Before we begin, let us recall the definition of the regret for matroid bandits:

$$R_n = \sum_{t=1}^{n}\mathbb{E}\left[\sum_{e\in A^*}\bar{w}(e) - \sum_{e\in A^t}\bar{w}(e)\right].$$

5.2 Regret Upper Bound Proof for DPUCB-MAT

Notation. Throughout this chapter, we will write $\mathbb{1}\{A \cap B\}$ as $\mathbb{1}\{A, B\}$ for sets A and B.

We seek to prove the regret bound for DPUCB-MAT (Algorithm 2) as given in Theorem 9:

$$R_n = \sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K, \log(Kn)\} \cdot \ln(Kn)}{\varepsilon/K}\right).$$

The proof of Theorem 9 makes use of the following two claims.

Lemma 17. *The regret of DPUCB-MAT (Algorithm 2) is*

$$\sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{K \ln(Kn)}{\varepsilon/K}\right). \tag{5.4}$$

Lemma 18. *The regret of DPUCB-MAT (Algorithm 2) is*

$$\sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\ln^2(Kn)}{\varepsilon/K}\right). \tag{5.5}$$

Proof of Theorem 9. The proof follows directly by combining Lemmas 17 and 18. ■

5.2.1 Proofs of Lemmas 17 and 18

To prove Lemmas 17 and 18, we will need Lemmas 20, 21, 22 and 23.

First, to prove Lemma 20, we need the following lemma from Kveton, Wen, Ashkan, Eydgahi, et al. (2014).

Lemma 19. *(Kveton, Wen, Ashkan, Eydgahi, et al., 2014, Lemma 3). Let $\Delta_1 \geq \ldots \geq \Delta_K$ be a sequence of reals in $(0, 1]$. Then we have*

$$\Delta_1 \frac{1}{\Delta_1^2} + \sum_{k=2}^{K} \Delta_k \left(\frac{1}{\Delta_k^2} - \frac{1}{\Delta_{k-1}^2}\right) \leq \frac{2}{\Delta_K}. \tag{5.6}$$

We now state and prove Lemma 20.

Lemma 20. *Let $\Delta_1 \geq \ldots \geq \Delta_K$ be a sequence of reals in $(0, 1]$. For any $\varepsilon_0 > 0$, we have*

$$\Delta_1 \frac{1}{\Delta_1 \cdot \min\{\Delta_1, \varepsilon_0\}} + \sum_{k=2}^{K} \Delta_k \left(\frac{1}{\Delta_k \cdot \min\{\Delta_k, \varepsilon_0\}} - \frac{1}{\Delta_{k-1} \cdot \min\{\Delta_{k-1}, \varepsilon_0\}}\right) \leq \frac{2}{\Delta_K} + \frac{K}{\varepsilon_0}. \tag{5.7}$$

Proof of Lemma 20.

Case 1: When $\varepsilon_0 \geq \Delta_1$, we have

$$\text{LHS of (5.7)} = \Delta_1 \frac{1}{\Delta_1^2} + \sum_{k=2}^{K} \Delta_k \left(\frac{1}{\Delta_k^2} - \frac{1}{\Delta_{k-1}^2} \right) \leq \frac{2}{\Delta_K} \leq \frac{2}{\Delta_K} + \frac{K}{\varepsilon_0}, \tag{5.8}$$

where the first inequality uses Lemma 19.

Case 2: When $\Delta_K \geq \varepsilon_0$, we have

$$\text{LHS of (5.7)} = \frac{1}{\varepsilon_0} + \frac{1}{\varepsilon_0} \sum_{k=2}^{K} \left(1 - \frac{\Delta_k}{\Delta_{k-1}} \right) \leq \frac{1}{\varepsilon_0} + \frac{1}{\varepsilon_0} \cdot (K-1) \leq \frac{K}{\varepsilon_0} + \frac{2}{\Delta_K}.$$

Case 3: When $\Delta_1 \geq \ldots \Delta_j \geq \varepsilon_0 \geq \Delta_{j+1} \geq \ldots \geq \Delta_K$ for some $j \in [K-1]$. We rewrite the LHS of (5.7) as

$$\begin{aligned}
\sum_{k=1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k \cdot \min\{\Delta_k, \varepsilon_0\}} + \frac{1}{\min\{\Delta_K, \varepsilon_0\}} &= \sum_{k=1}^{j} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k \cdot \varepsilon_0} + \sum_{k=j+1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k^2} + \frac{1}{\Delta_K} \\
&\leq \sum_{k=1}^{j} \left[\frac{1}{\varepsilon_0} - \frac{\Delta_{k+1}}{\varepsilon_0 \Delta_k} \right] + \sum_{k=j+1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k \cdot \Delta_{k+1}} + \frac{1}{\Delta_K} \\
&= \left[\frac{j}{\varepsilon_0} - \sum_{k=1}^{j} \frac{\Delta_{k+1}}{\varepsilon_0 \Delta_k} \right] + \left[\frac{1}{\Delta_K} - \frac{1}{\Delta_{j+1}} \right] + \frac{1}{\Delta_K} \\
&\leq \frac{j}{\varepsilon_0} + \frac{2}{\Delta_K} \\
&\leq \frac{K}{\varepsilon_0} + \frac{2}{\Delta_K},
\end{aligned} \tag{5.9}$$

which concludes the proof. ■

For the next lemmas, let $l_{e,k} = O\left(\frac{\ln(Kn)}{\Delta_{e,k} \cdot \min\{\Delta_{e,k}, \varepsilon_0\}} \right)$.

Lemma 21. *For any $\varepsilon_0 > 0$, we have*

$$0.5^0 \cdot l_{e,1} + \sum_{r=2}^{r_{\max,e}} 0.5^{r-1} \cdot (l_{e,r} - l_{e,r-1}) = O(\ln(Kn)) \cdot \left(\frac{2}{\Delta_{e,\min}} + \frac{\log(Kn)}{\varepsilon_0} \right). \tag{5.10}$$

Proof of Lemma 21. We have that LHS in (5.10) is

$$
\begin{aligned}
&0.5^0 \cdot l_{e,1} + \sum_{r=2}^{r_{\max,e}} 0.5^{r-1} \cdot (l_{e,r} - l_{e,r-1}) \\
&= O(\ln(Kn)) \cdot \left(0.5^0 \frac{1}{0.5^0 \cdot \min\{0.5^0, \varepsilon_0\}} + \right. \\
&\left. \sum_{r=2}^{r_{\max,e}} 0.5^{r-1} \left(\frac{1}{0.5^{r-1} \cdot \min\{0.5^{r-1}, \varepsilon_0\}} - \frac{1}{0.5^r \cdot \min\{0.5^r, \varepsilon_0\}} \right) \right) \\
&= O(\ln(Kn)) \cdot \left(\frac{2}{\Delta_{e,\min}} + \frac{\log(Kn)}{\varepsilon_0} \right) \quad ,
\end{aligned}
\tag{5.11}
$$

where the last step uses Lemma 20.

Note that $0.5^0 \geq 0.5^1 \geq \ldots \geq 0.5^{r_{\max,e}}$ and from $r_{\max,e} = \min\{\log(1/\Delta_{e,\min}), \log(Kn)\}$, we have $2^{r_{\max,e}} \leq \Delta_{e,\min}$ and $r_{\max,e} \leq \log(Kn)$. Combining the previous two facts, we get that 5.11 can be written as

$$
\begin{aligned}
&\left(0.5^0 \frac{1}{0.5^0 \cdot \min\{0.5^0, \varepsilon_0\}} + \sum_{r=2}^{r_{\max,e}} 0.5^{r-1} \left(\frac{1}{0.5^{r-1} \cdot \min\{0.5^{r-1}, \varepsilon_0\}} - \frac{1}{0.5^r \cdot \min\{0.5^r, \varepsilon_0\}} \right) \right) \\
&\leq \frac{2}{0.5^{r_{\max,e}}} + \frac{r_{\max,e}}{\varepsilon_0} \,,
\end{aligned}
$$

which concludes the proof. ■

To prove Lemma 23, we need the Lemma 22 which shows that the optimal base arm a_k^* was available when Algorithm 2 or Algorithm 5 selected $a^t_{\pi_t^{-1}(k)}$.

Lemma 22. *Fix $k \in [K]$. Let $A^t = (a_1^t, \ldots, a_K^t)$ be the super arm chosen by Algorithm 2 in round t and $A^t_{\pi_t^{-1}} = \left(a^t_{\pi_t^{-1}(1)}, \ldots, a^t_{\pi_t^{-1}(k)}, \ldots, a^t_{\pi_t^{-1}(K)} \right)$ be a permutation of A^t according to π_t^{-1}. We have $\left(a^t_{\pi_t^{-1}(1)}, \ldots, a^t_{\pi_t^{-1}(k-1)}, a_k^*, a^t_{\pi_t^{-1}(k+1)}, \ldots, a^t_{\pi_t^{-1}(K)} \right) \in \mathcal{I}$.*

Proof of Lemma 22. Let $\pi_t^{-1}(k) = i$ so that $\pi_t(i) = k$. Now, permuting $A^t_{\pi_t^{-1}}$ to the original order, we have $A^t = (a_1^t, \ldots, a_i^t, \ldots, a_K^t) \in \mathcal{I}$. Since a subset of an independent set is independent, we have $(a_1^t, \ldots, a_{i-1}^t) \in \mathcal{I}$. From the construction of π_t, we have $(a_1^t, \ldots, a_{i-1}^t, a^*_{\pi_t(i)}, \ldots, a^*_{\pi_t(K)}) \in \mathcal{I}$ and therefore $A_i^t := (a_1^t, \ldots, a_{i-1}^t, a^*_{\pi_t(i)}) \in \mathcal{I}$. Since $A^t, A_i^t \in \mathcal{I}$ and $|A^t| \geq |A_i^t|$, using the augmentation property of matroids, we can add elements from A^t to A_i^t such that the result is still independent. We add $a_{i+1}^t, \ldots, a_K^t \in A^t$ to

A_i^t such that $(a_1^t, \ldots, a_{i-1}^t, a^*_{\pi_t(i)}, a_{i+1}^t, \ldots, a_K^t) \in \mathcal{I}$. Denote $(a_1^t, \ldots, a_{i-1}^t, a^*_{\pi_t(i)}, a_{i+1}^t, \ldots, a_K^t)$ as $\tilde{A}^t = (\tilde{a}_1^t, \ldots, \tilde{a}_K^t)$ such that $\tilde{a}_i^t = a^*_{\pi_t(i)}$ and $\tilde{a}_j^t = a_j^t$ for $j \neq i$. Permuting $\tilde{A}^t$ using π_t^{-1}, we get $\left(\tilde{a}^t_{\pi_t^{-1}(1)}, \ldots, \tilde{a}^t_{\pi_t^{-1}(k-1)}, \tilde{a}^t_{\pi_t^{-1}(k)}, \tilde{a}^t_{\pi_t^{-1}(k+1)}, \ldots, \tilde{a}^t_{\pi_t^{-1}(K)}\right)$ which is the same as $\left(\tilde{a}^t_{\pi_t^{-1}(1)}, \ldots, \tilde{a}^t_{\pi_t^{-1}(k-1)}, \tilde{a}_i^t, \tilde{a}^t_{\pi_t^{-1}(k+1)}, \ldots, \tilde{a}^t_{\pi_t^{-1}(K)}\right)$. Since $\tilde{a}_i^t = a^*_{\pi_t(i)} = a_k^*$ and $\tilde{a}_j^t = a_j^t$ for $j \neq i$, we get the result. ■

The next lemma is adapted from Hu, Huang, et al. (2021). An alternate proof can be found in Azize et al. (2022).

Lemma 23. *Fix $k \in [K]$. For a suboptimal base arm $e \in \bar{A}^*$ and an optimal base arm $a_k^* \in A^*$, we have*

$$\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, T_e(t-1) > l_{e,k}\right\}\right] \Delta_{e,k} = O\left(\frac{1}{K^2}\right). \tag{5.12}$$

Proof of Lemma 23.

$$\begin{aligned}
\text{L.H.S} &= \sum_{t=1}^{n} \mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = e, T_e(t-1) > l_{e,k}\right\} \cdot \Delta_{e,k} \\
&= \sum_{t=1}^{n} \mathbb{P}\left\{U_t(e) \geq U_t(a_k^*), T_e(t-1) > l_{e,k}\right\} \cdot \Delta_{e,k} \\
&= \sum_{s=\lfloor \log(l_{e,k}) \rfloor + 1}^{\log(n)} \sum_{t=2^s}^{2^{s+1}-1} \mathbb{P}\left\{\tilde{w}_{e,2^{s-1}}(t-1) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \geq \right. \\
&\qquad \left. \tilde{w}_{a_k^*, T_{a_k^*}(t-1)}(t-1) + \sqrt{\frac{3\ln(Kt)}{T_{a_k^*}(t-1)}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_{a_k^*}(t-1)}\right\} \cdot \Delta_{e,k} \\
&\leq \sum_{s=\lfloor \log(l_{e,k}) \rfloor + 1}^{\log(n)} \sum_{t=2^s}^{2^{s+1}-1} \sum_{\tau=0}^{\log(t)} \mathbb{P}\left\{\tilde{w}_{e,2^{s-1}}(t-1) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \geq \right. \\
&\qquad \left. \tilde{w}_{a_k^*, 2^\tau}(t-1) + \sqrt{\frac{3\ln(Kt)}{2^\tau}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^\tau}\right\} \cdot \Delta_{e,k}\,,
\end{aligned}$$

where the second equality above used the fact that the base arm a_k^* was available from Lemma 22, but instead the suboptimal base arm e was chosen.

Now, when $\tilde{w}_{e,2^{s-1}}(t-1) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \geq \tilde{w}_{a_k^*,2^\tau}(t-1) + \sqrt{\frac{3\ln(Kt)}{2^\tau}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^\tau}$, at least

one of the following events must happen:

$$\tilde{w}_{e,2^{s-1}}(t-1) \geq \bar{w}(e) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \tag{5.13}$$

$$\tilde{w}_{a_k^*,2^{\tau}}(t-1) \leq \bar{w}(a_k^*) - \sqrt{\frac{3\ln(Kt)}{2^{\tau}}} - \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{\tau}} \tag{5.14}$$

$$\frac{\Delta_{e,k}}{2} < \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \quad . \tag{5.15}$$

We first bound the probability of the first event (5.13) using the concentration bounds for Laplace random variables (Lemma 15) and Hoeffding's inequality (Lemma 14).

$$\begin{aligned}
&\mathbb{P}\left\{\tilde{w}_{e,2^{s-1}}(t-1) \geq \bar{w}(e) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}}\right\} \\
&\leq \mathbb{P}\left\{\tilde{w}_{e,2^{s-1}}(t-1) \geq \hat{w}_{e,2^{s-1}}(t-1) + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}}\right\} + \\
&\qquad \mathbb{P}\left\{\hat{w}_{e,2^{s-1}}(t-1) \geq \bar{w}(e) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}}\right\} \\
&= O\left(\frac{1}{K^2t^2}\right) .
\end{aligned}$$

Similarly, for the second event (5.14) we have

$$\mathbb{P}\left\{\tilde{w}_{a_k^*,2^{\tau}}(t-1) \leq \bar{w}(a_k^*) - \sqrt{\frac{3\ln(Kt)}{2^{\tau}}} - \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{\tau}}\right\} \leq O\left(\frac{1}{K^2t^2}\right) .$$

We show that the third event (5.15) cannot happen using contradiction. From the R.H.S, we have

$$\begin{aligned}
\sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} &\leq \sqrt{\frac{3\ln(Kt)}{2^{\log(l_{e,k})}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{\log(l_{e,k})}} \\
&= \sqrt{\frac{3\ln(Kt)}{\frac{27\ln(Kn)}{\Delta_{e,k}\cdot\min\{\Delta_{e,k},\varepsilon_0\}}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot \frac{27\ln(Kn)}{\Delta_{e,k}\cdot\min\{\Delta_{e,k},\varepsilon_0\}}} \\
&< \frac{\Delta_{e,k}}{2} ,
\end{aligned}$$

which cannot happen since the L.H.S is $\frac{\Delta_{e,k}}{2}$. Therefore, we have

$$\begin{aligned}
&\sum_{t=1}^{n} \mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = e, T_e(t-1) > l_{e,k}\right\} \cdot \Delta_{e,k} \\
&\leq \sum_{s=\lfloor \log(l_{e,k})\rfloor+1}^{\log(n)} \sum_{t=2^s}^{2^{s+1}-1} \sum_{\tau=0}^{\log(t)} \left(\mathbb{P}\left\{ \tilde{w}_{e,2^{s-1}}(t-1) \geq \bar{w}(e) + \sqrt{\frac{3\ln(Kt)}{2^{s-1}}} + \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^{s-1}} \right\} + \right. \\
&\qquad \left. \mathbb{P}\left\{ \tilde{w}_{a_k^*,2^\tau}(t-1) \leq \bar{w}(a_k^*) - \sqrt{\frac{3\ln(Kt)}{2^\tau}} - \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^\tau} \right\} \right) \\
&= \sum_{s=\lfloor \log(l_{e,k})\rfloor+1}^{\log(n)} \sum_{t=2^s}^{2^{s+1}-1} \sum_{\tau=0}^{\log(t)} O\left(\frac{1}{K^2t^2}\right) \\
&= \sum_{t=1}^{n} O\left(\frac{\log(t)}{K^2t^2}\right) \\
&= \int_{t=1}^{n} O\left(\frac{1}{K^2t^2}\right) \\
&= O\left(\frac{1}{K^2}\right).
\end{aligned}$$

■

We are now ready to prove our first main lemma, Lemma 17.

Proof of Lemma 17. Recall from the regret decomposition in Chapter 4 (Equation 4.7) that the regret can be expressed as

$$R_n \leq \sum_{k=1}^{K} \sum_{e \in \bar{A}^* : \Delta_{e,k} > 0} \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right] \cdot \Delta_{e,k}}_{I_{e,k}} \quad . \tag{5.16}$$

The indicator function $\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}$ will be 1 when the learner selects a sub-optimal base arm e instead of the optimal base arm a_k^*. This further implies the differentially private upper confidence bound of e is no smaller than that of a_k^* in that round, i.e. $U_t(e) \geq U_t(a_k^*)$, which we use to prove Lemma 23. We decompose $I_{e,k}$ in (5.16) as

$$\begin{aligned}
I_{e,k} = &\underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, T_e(t-1) \leq l_{e,k}\right\}\right]}_{\Gamma_{1,k,e}} \Delta_{e,k} \\
&+ \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, T_e(t-1) > l_{e,k}\right\}\right]}_{\Gamma_{2,k,e}} \Delta_{e,k} \quad .
\end{aligned} \tag{5.17}$$

From Lemma 23, $\Gamma_{2,k,e} = O(1/K^2)$. Now, we use similar arguments as the one shown by Kveton, Wen, Ashkan, Eydgahi, et al. (2014) to complete the proof for Lemma 17. As for a fixed $e \in \bar{A}^*$, we maintain a counter $T_e(t-1)$ during learning. The counter counts the number of observations that are used to compute the differentially private empirical mean and the value of the counter doubles each time we update the mean. Note that since the gaps are ordered with $\Delta_{e,1} \geq \Delta_{e,2} \geq \ldots \geq \Delta_{e,K}$, we have that $l_{e,1} \leq l_{e,2} \leq \ldots \leq l_{e,K}$. Now, for all the rounds when the values of the counter are in the range of $[0, l_{e,1}]$, the total regret among all these rounds $\Delta_{e,1} \cdot O(l_{e,1})$. Similarly, for all the rounds when the values of the counter are in the range of $[l_{e,1}+1, l_{e,2}]$, the total regret among all these rounds is $\Delta_{e,2} \cdot O\left(l_{e,2} - l_{e,1}\right) + \Gamma_{2,1,e} = \Delta_{e,2} \cdot O\left(l_{e,2} - l_{e,1}\right) + O(1/K)$. Finally, for all the rounds when the values of the counter are in the range of $[l_{e,K-1}+1, l_{e,K}]$, the total regret among all these rounds is at most $\Delta_{e,K} \cdot O\left(l_{e,K} - l_{e,K-1}\right) + \sum_{k=1}^{K-1} \Gamma_{2,k,e} = \Delta_{e,K} \cdot O\left(l_{e,K} - l_{e,K-1}\right) + O(1/K)$. For all the rounds after the counter hits $l_{e,K}$, the total regret is at most $\sum_{k=1}^{K} \Gamma_{2,k,e} = O(1/K)$.

Using the above reasoning and Lemma 20, we have

$$\sum_{k=1}^{K} I_{e,k} = \Delta_{e,1} \cdot O(l_{e,1}) + \sum_{k=2}^{K} \Delta_{e,k} \cdot O\left(l_{e,k} - l_{e,k-1}\right) + \sum_{k=2}^{K+1} \sum_{q=1}^{k-1} \Gamma_{2,q,e} \tag{5.18}$$

$$= O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{K \ln(Kn)}{\varepsilon_0}\right), \tag{5.19}$$

which yields the result in the first claim, i.e., the regret is

$$\sum_{e \in \bar{A}^* : \Delta_{e,\min} > 0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{K \ln(Kn)}{\varepsilon / K}\right).$$

■

Now we prove our second main lemma, Lemma 18.

Proof of Lemma 18. Let $r_{\max,e} := \min\left\{\log\left(\frac{1}{\Delta_{e,\min}}\right), \log(Kn)\right\}$. For any $1 \leq r \leq r_{\max,e}$, let $\Phi_r := \{k \in [K] : \Delta_{e,k} \in [0.5^r, 0.5^{r-1})\}$ and $l_{e,r} = O\left(\frac{\ln(Kn)}{0.5^r \cdot \min\{0.5^r, \varepsilon_0\}}\right)$.

The algorithm chooses random super arms for the first L rounds since it needs to select all the base arms once. In this initialization phase, the maximum regret suffered is LK. As touched upon in Chapter 4, this can be improved using the matroid partitioning algorithm

by Edmonds (1965). However this only contributes a lower order term in our regret as we will see. Therefore we do not need the improved initialization for our purposes. We have the following regret decomposition.

$$\begin{aligned}
R_n &= \sum_{k=1}^{K} \sum_{e\in\bar{A}^*:\Delta_{e,k}>0} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right]\cdot\Delta_{e,k} \\
&\le LK + \sum_{e\in\bar{A}^*} \sum_{k\in[K]:\Delta_{e,k}\ge\frac{1}{Kn}} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right]\cdot\Delta_{e,k} \\
&\le LK + \sum_{e\in\bar{A}^*} \sum_{r=1}^{r_{\max,e}} \sum_{k\in\Phi_r} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right]\cdot\Delta_{e,k} \\
&\le LK + \sum_{e\in\bar{A}^*} \sum_{r=1}^{r_{\max,e}} \sum_{k\in\Phi_r} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e\right\}\right]\cdot 0.5^r \\
&\le LK + \sum_{e\in\bar{A}^*} \sum_{r=1}^{r_{\max,e}} \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\exists k\in\Phi_r : a^t_{\pi_t^{-1}(k)} = e\right\}\right]\cdot 0.5^r}_{I_{e,r}} .
\end{aligned} \tag{5.20}$$

Let $F_{e,r}$ denote the event that $\left\{\exists k\in\Phi_r : a^t_{\pi_t^{-1}(k)} = e\right\}$.

Now, we decompose $I_{e,r}$ as

$$\begin{aligned}
I_{e,r} = &\underbrace{\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}\left\{F_{e,r}\cap\left\{T_e(t-1)\le l_{e,r}\right\}\right\}]\cdot 0.5^r}_{\Gamma_{1,r,e}} \\
&+ \underbrace{\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}\left\{F_{e,r}\cap\left\{T_e(t-1)\le l_{e,r}\right\}\right\}]\cdot 0.5^r}_{\Gamma_{2,r,e}} .
\end{aligned} \tag{5.21}$$

Lemma 23 can be adapted to show that $\Gamma_{2,r,e} = O(1/K^2)$. So the total regret is

$$\begin{aligned}
&\sum_{e\in\bar{A}^*} \left(0.5^0\cdot O\left(l_{e,1}\right) + \sum_{r=2}^{r_{\max,e}} 0.5^{r-1}\cdot O\left(l_{e,r}-l_{e,r-1}\right) + O(1)\right) \\
&= \sum_{e\in\bar{A}^*} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\ln^2(Kn)}{\varepsilon_0}\right) ,
\end{aligned} \tag{5.22}$$

where we use Lemma 21. ■

In the next section, we prove Theorem 11.

5.3 Regret Upper Bound Proof for DPTS-MAT

First, let's recall the regret bound for DPTS-MAT as given in Theorem 11 that we seek to prove:

$$R_n = \sum_{e\in\bar{A}^*:\Delta_{e,\min}>0} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K,\log(Kn)\}\cdot\ln(Kn)}{\varepsilon/K}\right) + \sum_{k\in[K]:\Delta_{\min,k}>0} O\left(\frac{\ln(Kn)}{\Delta_{\min,k}}\right).$$

Now recall from the regret decomposition (Equation 4.7) the regret can be expressed as

$$R_n \le \sum_{k=1}^{K}\sum_{e\in\bar{A}^*:\Delta_{e,k}>0}\sum_{t=1}^{n}\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)}=e\right\}\right]\cdot\Delta_{e,k}\,.$$

Similar to the DPUCB-MAT case, the indicator function $\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)}=e\right\}$ will be 1 when the learner selects a sub-optimal base arm e instead of the optimal base arm a^*_k. This implies that the posterior sample of e is no smaller than the posterior sample of a^*_k in that round. For a fixed $k\in[K]$ and $e\in\bar{A}^*$, we define the event

$$\mathcal{E}^{\theta}_{e,k}(t) := \{\theta_e(t)\le y_{e,k}\}\,,$$

where $y_{e,k} := \bar{w}(a^*_k) - \frac{1}{3}\Delta_{e,k}$, to decompose the regret. By introducing $\mathcal{E}^{\theta}_{e,k}(t)$, the regret can be decomposed as

$$\begin{aligned} R_n &\le \sum_{k=1}^{K}\sum_{e\in\bar{A}^*}\sum_{t=1}^{n}\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)}=e\right\}\right]\cdot\Delta_{e,k} \\ &= \underbrace{\sum_{k=1}^{K}\sum_{e\in\bar{A}^*}\underbrace{\sum_{t=1}^{n}\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)}=e,\overline{\mathcal{E}^{\theta}_{e,k}(t)}\right\}\right]\cdot\Delta_{e,k}}_{V_{e,k}}}_{V} \\ &\quad + \underbrace{\sum_{k=1}^{K}\sum_{e\in\bar{A}^*}\underbrace{\sum_{t=1}^{n}\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)}=e,\mathcal{E}^{\theta}_{e,k}(t)\right\}\right]\cdot\Delta_{e,k}}_{U_{e,k}}}_{U}\,. \end{aligned} \tag{5.23}$$

We give the following two lemmas to provide bounds on V and U in 5.23, which in turn gives us the final regret bound.

Lemma 24. *The value of as V as defined in 5.23 is*

$$V = \sum_{e \in \bar{A}^*} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K, \ln(Kn)\} \cdot \ln(Kn)}{\varepsilon_0}\right).$$

Lemma 25. *The value of as U as defined in 5.23 is*

$$U = \sum_{k=1}^{K} O\left(\frac{\ln(Kn)}{\Delta_{k,\min}}\right).$$

Proof of Theorem 11. The proof follows directly by combining Lemmas 24 and 25. ■

5.3.1 Proof of Lemma 24

To show our first main lemma, Lemma 24, we will need the following result.

We first define two events. Let

$$C_e(t) := \left\{|\hat{w}_{e,T_e(t-1)}(t-1) - \bar{w}(e)| \leq \sqrt{\frac{3\ln(Kt)}{T_e(t-1)}}\right\}$$

and

$$G_e(t) := \left\{|\tilde{w}_{e,T_e(t-1)}(t-1) - \hat{w}_{e,T_e(t-1)}(t-1)| \leq \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)}\right\}.$$

So C_e is the event that the mean reward of $e \in E$ is within the confidence interval in round t and G_e is the event that the noise added is not too much in round t. Let $\overline{C_e(t)}$ and $\overline{G_e(t)}$ denote the complements of events $C_e(t)$ and $G_e(t)$, respectively.

Lemma 26. *We have*

$$\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{G_e(t)}\right\}\right] = O\left(\frac{1}{K^2n^2}\right),$$

and

$$\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{C_e(t)}\right\}\right] = O\left(\frac{1}{K^2n^2}\right).$$

Proof of Lemma 26. We have

$$\begin{aligned}
\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{G_e(t)}\right\}\right] &= \sum_{t=1}^{n} \mathbb{P}\left\{|\tilde{w}_{e,T_e(t-1)}(t-1) - \hat{w}_{e,T_e(t-1)}(t-1)| > \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)}\right\} \\
&\le \sum_{t=1}^{n} \sum_{s=0}^{\lfloor \log(t) \rfloor} \mathbb{P}\left\{|\tilde{w}_{e,2^s}(t-1) - \hat{w}_{e,2^s}(t-1)| > \frac{3\ln(Kt)}{\varepsilon_0 \cdot 2^s}\right\} \\
&\le \sum_{t=1}^{n} \sum_{s=0}^{\lfloor \log(t) \rfloor} \mathbb{P}\left\{|2^s \cdot \tilde{w}_{e,2^s}(t-1) - 2^s \cdot \hat{w}_{e,2^s}(t-1)| > \frac{3\ln(Kt)}{\varepsilon_0}\right\} \\
&= \sum_{t=1}^{n} \sum_{s=0}^{\lfloor \log(t) \rfloor} e^{-3\ln(Kt)} \\
&= O\left(\frac{1}{K^2 n^2}\right),
\end{aligned}$$

where the second inequality used the concentration bound of a Laplace random variable (Lemma 15).

Similarly, we have

$$\begin{aligned}
\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{C_e(t)}\right\}\right] &= \sum_{t=1}^{n} \mathbb{P}\left\{|\hat{w}_{e,T_e(t-1)}(t-1) - \bar{w}(e)| > \sqrt{\frac{3\ln(Kt)}{T_e(t-1)}}\right\} \\
&\le \sum_{t=1}^{n} \sum_{s=0}^{\lfloor \log(t) \rfloor} \mathbb{P}\left\{|\hat{w}_{e,2^s}(t-1) - \bar{w}(e)| > \sqrt{\frac{3\ln(Kt)}{2^s}}\right\} \\
&\le \sum_{t=1}^{n} \sum_{s=0}^{\lfloor \log(t) \rfloor} 2e^{-2\cdot 2^s \cdot \frac{3\ln(Kt)}{2^s}} \\
&= O\left(\frac{1}{K^2 n^2}\right),
\end{aligned}$$

where the second inequality uses Hoeffding's inequality (Lemma 14). ■

Now we are ready to prove our first main lemma, Lemma 24.

Proof of Lemma 24. The proof is very similar to the regret analysis of `DPUCB-MAT` (Algorithm 2).

Let $l_{e,k} := \frac{72\ln(nK)}{\min\left\{\Delta_{e,k}^2, \varepsilon_0 \cdot \Delta_{e,k}\right\}}$. Then, $V_{e,k}$ can be further decomposed as

$$\begin{aligned} V_{e,k} = & \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, T_e(t-1) \leq l_{e,k}\right\}\right] \cdot \Delta_{e,k}}_{\Gamma_{1,k,e}} \\ & + \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, T_e(t-1) > l_{e,k}\right\}\right] \cdot \Delta_{e,k}}_{\Gamma_{2,k,e}} \quad . \end{aligned} \tag{5.24}$$

We further decompose $\Gamma_{2,k,e}$ as

$$\begin{aligned} \Gamma_{2,k,e} &= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, T_e(t-1) > l_{e,k}\right\}\right] \cdot \Delta_{e,k} \\ &\leq \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, C_e(t), G_e(t), T_e(t-1) > l_{e,k}\right\}\right] \cdot \Delta_{e,k}}_{\gamma} \\ &+ \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{C_e(t)}\right\}\right] + \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{G_e(t)}\right\}\right]}_{=O\left(\frac{1}{K^2n^2}\right), \text{ Lemma 26}} . \end{aligned} \tag{5.25}$$

Let $d_{e,k} := \lceil \log(l_{e,k}) \rceil$. Let τ_s denote the round by the end of which there are exactly 2^s fresh observations that will be used to compute the differentially private empirical mean of a sub-optimal base arm e. Also, let $\mathcal{F}_{t-1}$ denote the history till the end of round $t-1$ that contains the selected base arms, the weights observed corresponding to those arms, and the noise added for differential privacy. Now we upper bound γ. We have

$$\begin{aligned} \gamma &= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, C_e(t), G_e(t), T_e(t-1) > l_{e,k}\right\}\right] \cdot \Delta_{e,k} \\ &\leq \sum_{s=d_e}^{\log n} \mathbb{E}\left[\sum_{t=\tau_s+1}^{\tau_{s+1}} \mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, C_e(t), G_e(t), T_e(t-1) > l_{e,k}\right\}\right] \cdot \Delta_{e,k} \\ &\leq \sum_{s=d_e}^{\log n} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, C_e(t), G_e(t), T_e(t-1) = 2^s\right\}\right] \cdot \Delta_{e,k} \\ &= \sum_{s=d_e}^{\log n} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = e, \overline{\mathcal{E}_{e,k}^{\theta}(t)}, C_e(t), G_e(t), T_e(t-1) = 2^s\right\} \mid \mathcal{F}_{t-1}\right]\right] \cdot \Delta_{e,k} \\ &\leq \sum_{s=d_e}^{\log n} \sum_{t=1}^{n} \underbrace{\mathbb{E}\left[\mathbb{1}\left\{C_e(t), G_e(t), T_e(t-1) = 2^s\right\} \underbrace{\mathbb{E}\left[\mathbb{1}\left\{\overline{\mathcal{E}_{e,k}^{\theta}(t)}\right\} \,\middle|\, \mathcal{F}_{t-1}\right]}_{\lambda}\right]}_{\Lambda} \cdot \Delta_{e,k} \quad . \end{aligned} \tag{5.26}$$

Now we consider two cases based on whether the event $\mathcal{E} = C_e(t) \cap G_e(t) \cap \{T_e(t) = 2^s\}$ happens or not.

Case 1: When $\mathcal{E}$ does not happen, we have $\Lambda = 0$.

Case 2: When $\mathcal{E}$ happens, we construct an upper bound for λ to upper bound Λ. To do so, recall that $\theta_e(t) \sim \mathcal{N}\left(w'_{e,T_e(t-1)}(t), \frac{1}{T_e(t-1)}\right)$, where $w'_{e,T_e(t-1)}(t) = \tilde{w}_{e,T_e(t-1)}(t-1) + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)}$ and $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. We have

$$
\begin{aligned}
w'_{e,T_e(t-1)}(t) &= \tilde{w}_{e,T_e(t-1)}(t-1) + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)} \\
&\le \hat{w}_{e,T_e(t-1)}(t-1) + \frac{6\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)} \qquad (\text{since } \mathbb{1}\{G_e(t-1)\} = 1) \\
&\le \bar{w}(e) + \frac{6\ln(Kt)}{\varepsilon_0 \cdot T_e(t-1)} + \sqrt{\frac{3\ln(Kt)}{T_e(t-1)}} \qquad (\text{since } \mathbb{1}\{C_e(t-1)\} = 1) \\
&= \bar{w}(e) + \frac{6\ln(Kt)}{\varepsilon_0 \cdot 2^s} + \sqrt{\frac{3\ln(Kt)}{2^s}} \qquad (\text{since } \mathbb{1}\{T_e(t-1) = 2^s\} = 1) \\
&\le \bar{w}(e) + \frac{\Delta_{e,k}}{12} + \frac{\Delta_{e,k}}{\sqrt{18}} \qquad \left(\text{for } s \ge \left\lceil \log \frac{72\ln(nK)}{\min\left\{\Delta_{e,k}^2, \varepsilon_0 \cdot \Delta_{e,k}\right\}} \right\rceil\right) \\
&\le \bar{w}(e) + \frac{1}{3}\Delta_{e,k}\,.
\end{aligned}
$$

We say that a Gaussian random variable with mean μ is stochastically dominated by a Gaussian random variable with mean μ' if $\mu' \ge \mu$. Therefore, by the last display, a $\mathcal{N}\left(w'_{e,T_e(t-1)}(t), \frac{1}{T_e(t-1)}\right)$ distributed random variable is stochastically dominated by a $\mathcal{N}\left(\bar{w}(e) + \frac{1}{3}\Delta_{e,k}, \frac{1}{T_e(t-1)}\right)$ distributed random variable.

Next, we introduce the tail function $\mathcal{L}_{\mu,\sigma^2}(x)$ that maps $x \in \mathbb{R}$ to the tail probability $\mathbb{P}\{X > x\}$ of $X \sim \mathcal{N}(\mu, \sigma^2)$. Now, we upper bound $\mathbb{E}\left[\mathbb{1}\left\{\overline{\mathcal{E}^{\theta}_{e,k}(t)}\right\} \,\middle|\, \mathcal{F}_{t-1}\right]$ using stochastic dominance between two Gaussian distributed random variables. We have

$$
\begin{aligned}
\mathbb{E}\left[\mathbb{1}\left\{\overline{\mathcal{E}^{\theta}_{e,k}(t)}\right\}\,\middle|\,\mathcal{F}_{t-1}\right] &= \mathbb{P}\left\{\theta_e(t) > \bar{w}(e) + \frac{2}{3}\Delta_{e,k}\,\middle|\,\mathcal{F}_{t-1}\right\} \\
&= \mathcal{L}_{w'_{e,T_e(t-1)}(t),\frac{1}{T_e(t-1)}}\left(\bar{w}(e) + \frac{2}{3}\Delta_{e,k}\right) \\
&\leq \mathcal{L}_{\bar{w}(e)+\frac{1}{3}\Delta_{e,k},\frac{1}{T_e(t-1)}}\left(\bar{w}(e) + \frac{2}{3}\Delta_{e,k}\right) \\
&= \mathcal{L}_{\bar{w}(e),\frac{1}{T_e(t-1)}}\left(\bar{w}(e) + \frac{1}{3}\Delta_{e,k}\right) \\
&\overset{(\clubsuit)}{\leq} \exp\left(-\frac{1}{18}\cdot\Delta^2_{e,k}\cdot\frac{72\ln(nK)}{\min\left\{\Delta^2_{e,k},\varepsilon_0\cdot\Delta_{e,k}\right\}}\right) \\
&= O\left(\frac{1}{(nK)^4}\right),
\end{aligned}
\tag{5.27}
$$

where ($\clubsuit$) uses the concentration bounds for a normally distributed random variable that is shown in Lemma 16, and the fact that $T_e(t-1) = 2^s \geq \frac{72\ln(nK)}{\min\left\{\Delta^2_{e,k},\varepsilon_0\cdot\Delta_{e,k}\right\}}$.

Putting all the pieces together, we have $\Gamma_{2,k,e} = O\left(\frac{1}{K^2}\right)$. Now, we use similar arguments that have been used for the proofs of Theorem 9 to upper bound term V in (5.23). Recall that $T_e(t-1)$ is a counter that counts the number of fresh observations that have been used to compute the differentially private empirical mean of a sub-optimal base arm e. For all the rounds when the counter is in the range of $[0, l_{e,1}]$, the total regret among all these rounds is at most $\Delta_{e,1}\cdot O\left(l_{e,1}\right)$. For all the rounds when the values of the counter are in the range of $[l_{e,1}+1, l_{e,2}]$, the total regret among all these rounds is upper bounded by $\Delta_{e,2}\cdot O\left(l_{e,2}-l_{e,1}\right)+\Gamma_{2,1,e} \leq \Delta_{e,2}\cdot O\left(l_{e,2}-l_{e,1}\right)+O(1/K)$. Finally, for all the rounds when the values of the counter are in the range of $[l_{e,K-1}+1, l_{e,K}]$, the total regret among all these rounds is at most $\Delta_{e,K}\cdot O\left(l_{e,K}-l_{e,K-1}\right)+\sum_{k=1}^{K-1}\Gamma_{2,k,e} = \Delta_{e,K}\cdot O\left(l_{e,K}-l_{e,K-1}\right)+O(1/K)$. For all the rounds after the counter hits $l_{e,K}$, the total regret is at most $\sum_{k=1}^{K}\Gamma_{2,k,e} = O(1/K)$.

By using Lemma 20, we have

$$
\begin{aligned}
V &= \sum_{e\in\bar{A}^*}\left(\Delta_{e,1}\cdot O\left(l_{e,1}\right) + \sum_{k=2}^{K}\Delta_{e,k}\cdot O\left(l_{e,k}-l_{e,k-1}\right) + \sum_{k=2}^{K+1}\sum_{q=1}^{k-1}\Gamma_{2,q,e}\right) \\
&= \sum_{e\in\bar{A}^*} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{K\ln(Kn)}{\varepsilon_0}\right).
\end{aligned}
\tag{5.28}
$$

Similarly, by using Lemma 21, we have

$$V = \sum_{e \in \bar{A}^*} O(\ln(Kn)) \cdot \left(\frac{2}{\Delta_{e,K}} + \frac{\log(Kn)}{\varepsilon_0} \right) \tag{5.29}$$

$$= \sum_{e \in \bar{A}^*} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\ln^2(Kn)}{\varepsilon_0} \right). \tag{5.30}$$

Combining (5.28) and (5.29), we have

$$V = \sum_{e \in \bar{A}^*} O(\ln(Kn)) \cdot \left(\frac{2}{\Delta_{e,K}} + \frac{\log(Kn)}{\varepsilon_0} \right)$$

$$= \sum_{e \in \bar{A}^*} O\left(\frac{\ln(Kn)}{\Delta_{e,\min}} + \frac{\min\{K, \ln(Kn)\} \cdot \ln(Kn)}{\varepsilon_0} \right).$$

■

5.3.2 Proof of Lemma 25

To prove Lemma 25, we need the following results. The proof of Lemma 28 is omitted and can be found in Agrawal et al. (2017).

For the next lemma, let $Y^{\theta}_{e,k}(t) := \mathbb{P}\left\{ \theta_{a^*_k}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1} \right\}$.

Lemma 27. *For all $t \in \mathbb{N}$ we have*

$$\mathbb{P}\left\{ a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t) \,\middle|\, \mathcal{F}_{t-1} \right\}$$

$$\leq \frac{1 - Y^{\theta}_{e,k}(t)}{Y^{\theta}_{e,k}(t)} \mathbb{P}\left\{ a^t_{\pi_t^{-1}(k)} = a^*_k, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t) \,\middle|\, \mathcal{F}_{t-1} \right\}. \tag{5.31}$$

Proof of Lemma 27. We start by noting that the event $G_{a^*_k}(t)$ is determined by the history $\mathcal{F}_{t-1}$.

Case 1: If F_{t-1} is the one such that $G_{a^*_k}(t)$ is false, then both sides of the inequality shown in (5.31) are zero, and the inequality trivially holds.

Case 2: If F_{t-1} is the one such that $G_{a^*_k}(t)$ is true, we can omit $G_{a^*_k}$ in both sides in (5.31). Let $\pi_t^{-1}(k) = i$ and $A^t_{i-1} = \{a^t_1, \ldots, a^t_{i-1}\}$ be the set of the first $i-1$ base arms selected greedily by Algorithm 5. To complete the proof, it suffices to show

$$\mathbb{P}\left\{ a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t) \,\middle|\, \mathcal{F}_{t-1}, A^t_{i-1} \right\}$$

$$\leq \frac{1 - Y^{\theta}_{e,k}(t)}{Y^{\theta}_{e,k}(t)} \mathbb{P}\left\{ a^t_{\pi_t^{-1}(k)} = a^*_k, \mathcal{E}^{\theta}_{e,k}(t) \,\middle|\, \mathcal{F}_{t-1}, A^t_{i-1} \right\}. \tag{5.32}$$

Let $\sigma(A_{i-1}^t) = \left\{e : e \in E \setminus A_{i-1}^t, A_{i-1}^t \cup \{e\} \in \mathcal{I}\right\}$ be the set of base arms that can be added to the current solution set A_{i-1}^t. Note that $a_k^* \in \sigma(A_{i-1}^t)$ and $a_k^* \notin A_{i-1}^t$.

We first construct an upper bound for the LHS of (5.32). We have

$$\begin{aligned}
&\mathbb{P}\left\{a_{\pi_t^{-1}(k)}^t = e, \mathcal{E}_{e,k}^\theta(t) \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\le \mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\overset{(\spadesuit)}{=} \mathbb{P}\left\{\theta_{a_k^*}(t) \le y_{e,k} \,\middle|\, A_{i-1}^t, \mathcal{F}_{t-1}\right\} \cdot \\
&\mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&= \mathbb{P}\left\{\theta_{a_k^*}(t) \le y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\} \cdot \mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&= (1 - Y_{e,k}^\theta(t)) \cdot \mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\},
\end{aligned} \tag{5.33}$$

where ($\spadesuit$) uses the fact that $\theta_{a_k^*}(t)$ and all base arms in $\sigma(A_{i-1}^t)$ are independent.

Similarly, the RHS of (5.32) is lower bounded by

$$\begin{aligned}
&\mathbb{P}\left\{a_{\pi_t^{-1}(k)}^t = a_k^*, \mathcal{E}_{e,k}^\theta(t) \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\qquad \ge \mathbb{P}\left\{\theta_{a_k^*}(t) > y_{e,k} \ge \theta_j(t), \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\qquad = \mathbb{P}\left\{\theta_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1} A_{i-1}^t\right\} \cdot \\
&\qquad\quad \mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\qquad = Y_{e,k}^\theta(t) \cdot \mathbb{P}\left\{\theta_j(t) \le y_{e,k}, \forall j \in \sigma(A_{i-1}^t) \setminus \{a_k^*\} \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\}.
\end{aligned} \tag{5.34}$$

Combining (5.33) and (5.34) gives

$$\begin{aligned}
&\mathbb{P}\left\{a_{\pi_t^{-1}(k)}^t = e, \mathcal{E}_{e,k}^\theta(t) \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\} \\
&\qquad \le \frac{1 - Y_{e,k}^\theta(t)}{Y_{e,k}^\theta(t)} \mathbb{P}\left\{a_{\pi_t^{-1}(k)}^t = a_k^*, \mathcal{E}_{e,k}^\theta(t) \,\middle|\, \mathcal{F}_{t-1}, A_{i-1}^t\right\}.
\end{aligned} \tag{5.35}$$

To get the stated result, we use the law of total expectation and the fact that $Y_{e,k}^\theta$ is deter-

mined by $\mathcal{F}_{t-1}$. We have

$$\begin{aligned}
&\mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^\theta_{e,k}(t) \,\middle|\, \mathcal{F}_{t-1}\right\} \\
=&\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^\theta_{e,k}(t)\right\} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right] \\
=&\mathbb{E}\left[\mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^\theta_{e,k}(t) \middle| A^t_{i-1}, \mathcal{F}_{t-1}\right\} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right] \\
\leq&\mathbb{E}\left[\frac{1 - Y^\theta_{e,k}(t)}{Y^\theta_{e,k}(t)}\mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = a^*_k, \mathcal{E}^\theta_{e,k}(t) \,\middle|\, A^t_{i-1}, \mathcal{F}_{t-1}\right\} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right] \\
=&\frac{1 - Y^\theta_{e,k}(t)}{Y^\theta_{e,k}(t)}\mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = a^*_k, \mathcal{E}^\theta_{e,k}(t) \,\middle|\, \mathcal{F}_{t-1}\right\},
\end{aligned} \tag{5.36}$$

where the only inequality uses (5.35).

■

For the next lemma, let $\hat{\theta} \sim \mathcal{N}\left(\hat{w}_{a^*_k, T_{a^*_k}(t-1)}(t-1), \frac{1}{T_{a^*_k}(t-1)}\right)$.

Lemma 28. *(Agrawal et al., 2017, Lemma 2.13). Let τ_s be the round by the end of which we use fresh 2^s observations to update the empirical mean of an optimal base arm a^*_k. Then, we have*

$$\mathbb{E}\left[\frac{1 - Y^{\hat{\theta}}_{r,k}(\tau_s + 1)}{Y^{\hat{\theta}}_{r,k}(\tau_s + 1)}\right] \leq \begin{cases} O(1) & \forall s\,, \\ O\left(\frac{1}{L^2 n}\right) & s \geq \log\left(l^*_{e,k}\right)\,, \end{cases}$$

where $l^*_{e,k} := \left\lceil \log\left(\frac{288 \ln\left(L^2\left(n + e^{32}\right)\right)}{\Delta^2_{e,k}}\right)\right\rceil$.

Now we are ready to prove the second main lemma for the regret bound of `DPTS-MAT`, Lemma 25.

Proof of Lemma 25.

Recall

$$U = \sum_{k=1}^{K} \sum_{e \in \bar{A}^*} \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^\theta_{e,k}(t)\right\}\right] \cdot \Delta_{e,k}}_{U_{e,k}}\,. \tag{5.37}$$

Let $l^*_{e,k} := \left\lceil \frac{288 \ln\left(L^2\left(n+e^{32}\right)\right)}{\Delta^2_{e,k}} \right\rceil$ and $d^*_{e,k} = \log\left(l^*_{e,k}\right)$. We first decompose $U_{e,k}$ as

$$
\begin{aligned}
U_{e,k} &= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t)\right\}\right] \cdot \Delta_{e,k} \\
&\leq \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t)\right\}\right] + \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{\overline{G_{a^*_k}(t)}\right\}\right]}_{O\left(\frac{1}{K^2}\right), \text{ Lemma 26}} \\
&= \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t), T_{a^*_k}(t-1) \leq l^*_{e,k}\right\}\right]}_{\Gamma_{1,k,e}} \cdot \Delta_{e,k} \\
&\quad + \underbrace{\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t), T_{a^*_k}(t-1) > l^*_{e,k}\right\}\right] \cdot \Delta_{e,k} + O\left(\frac{1}{K^2}\right)}_{\Gamma_{2,k,e}} \quad .
\end{aligned}
\tag{5.38}
$$

Recall $Y^{\theta}_{e,k}(t) := \mathbb{P}\left\{\theta_{a^*_k}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}$. Then, we have

$$
\begin{aligned}
\Gamma_{1,k,e} &= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t), T_{a^*_k}(t-1) \leq l^*_{e,k}\right\}\right] \\
&= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t), T_{a^*_k}(t-1) \leq l^*_{e,k}\right\} \,\middle|\, \mathcal{F}_{t-1}\right]\right] \\
&= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t)\right\} \cdot \mathbb{1}\left\{T_{a^*_k}(t-1) \leq l^*_{e,k}\right\} \,\middle|\, \mathcal{F}_{t-1}\right]\right] \\
&= \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{T_{a^*_k}(t-1) \leq l^*_{e,k}\right\} \cdot \mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = e, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t) \,\middle|\, \mathcal{F}_{t-1}\right\}\right] \\
&\overset{(a)}{\leq} \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\left\{T_{a^*_k}(t-1) \leq l^*_{e,k}\right\} \cdot \frac{1 - Y^{\theta}_{e,k}(t)}{Y^{\theta}_{e,k}(t)} \mathbb{P}\left\{a^t_{\pi_t^{-1}(k)} = a^*_k, \mathcal{E}^{\theta}_{e,k}(t), G_{a^*_k}(t) \,\middle|\, \mathcal{F}_{t-1}\right\}\right] \\
&\leq \sum_{t=1}^{n} \mathbb{E}\left[\underbrace{\mathbb{1}\left\{T_{a^*_k}(t-1) \leq l^*_{e,k}\right\} \cdot \frac{1 - Y^{\theta}_{e,k}(t)}{Y^{\theta}_{e,k}(t)} \mathbb{1}\left\{a^t_{\pi_t^{-1}(k)} = a^*_k, G_{a^*_k}(t)\right\}}_{\eta}\right] \quad ,
\end{aligned}
\tag{5.39}
$$

where inequality (a) uses Lemma 27.

We now reduce the proof to the non-private setting. First, we divide all the realizations F_{t-1} of $\mathcal{F}_{t-1}$ into two groups depending on whether $\mathbb{1}\left\{G_{a^*_k}(t), T_{a^*_k}(t-1) \leq l^*_{e,k}\right\}$ is 1 or 0.

Case 1: For F_{t-1} such that $\mathbb{1}\left\{G_{a_k^*}(t), T_{a_k^*}(t-1) \leq l_{e,k}^*\right\} = 0$, we have $\eta = 0$.

Case 2: For F_{t-1} such that $\mathbb{1}\left\{G_{a_k^*}(t), T_{a_k^*}(t-1) \leq l_{e,k}^*\right\} = 1$, we have $w'_{a_k^*, T_{a_k^*}(t-1)}(t) = \tilde{w}_{a_k^*, T_{a_k^*}(t-1)}(t-1) + \frac{3\ln(Kt)}{\varepsilon_0 \cdot T_{a_k^*}(t-1)} \geq \hat{w}_{a_k^*, T_{a_k^*}(t-1)}(t-1)$.

Since a random variable drawn from $\mathcal{N}(\mu, \sigma^2)$ stochastically dominates a random variable drawn from $\mathcal{N}(\mu', \sigma^2)$ if $\mu \geq \mu'$, we have

$$\frac{1 - Y_{e,k}^{\theta}(t)}{Y_{e,k}^{\theta}(t)} = \frac{\mathbb{P}\left\{\theta_{a_k^*}(t) \leq y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}}{\mathbb{P}\left\{\theta_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}} \leq \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) \leq y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}},$$

where $\hat{\theta}_{a_k^*}(t) \sim \mathcal{N}\left(\hat{w}_{a_k^*, T_{a_k^*}(t-1)}(t-1), \frac{1}{T_{a_k^*}(t-1)}\right)$.

From these two cases, for any F_{t-1}, we have

$$\eta \leq \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) \leq y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}} \mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = a_k^*, G_{a_k^*}(t)\right\} \quad . \tag{5.40}$$

Now, the proof is reduced to the non-private setting. We divide all the n rounds depending on when the empirical mean of a_k^* changes, i.e., whether $\hat{w}_{a_k^*, T_{a_k^*}(t-1)}(t-1)$ changes. Let τ_s denote the round by the end of which we use fresh 2^s observations to update the empirical mean of a_k^*. Note that τ_s is random. Then, we have

$$\begin{aligned}
(5.39) &\leq \sum_{s=0}^{d_{e,k}^*} \mathbb{E}\left[\sum_{t=\tau_s+1}^{\tau_{s+1}} \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) \leq y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}} \cdot \mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = a_k^*\right\} \cdot \mathbb{1}\left\{G_{a_k^*}(t)\right\}\right] \\
&\leq \sum_{s=0}^{d_{e,k}^*} \mathbb{E}\left[\sum_{t=\tau_s+1}^{\tau_{s+1}} \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) \leq y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(t) > y_{e,k} \,\middle|\, \mathcal{F}_{t-1}\right\}} \cdot \mathbb{1}\left\{a_{\pi_t^{-1}(k)}^t = a_k^*\right\}\right] \\
&\leq \sum_{s=0}^{d_{e,k}^*} \mathbb{E}\left[2^{s+1} \cdot \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(\tau_s+1) \leq y_{e,k} \,\middle|\, \mathcal{F}_{\tau_s} = F_{\tau_s}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(\tau_s+1) > y_{e,k} \,\middle|\, \mathcal{F}_{\tau_s} = F_{\tau_s}\right\}}\right] \\
&= O\left(\frac{\ln(Kn)}{\Delta_{e,k}^2}\right) \quad ,
\end{aligned} \tag{5.41}$$

where the last inequality uses Lemma 28.

Similarly, we will have

$$\begin{aligned}
\Gamma_{2,k,e} &\leq \sum_{s=d_{e,k}^*+1}^{\log(n)} \mathbb{E}\left[2^{s+1} \cdot \frac{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(\tau_s+1) \leq y_{e,k} \,\middle|\, \mathcal{F}_{\tau_s} = F_{\tau_s}\right\}}{\mathbb{P}\left\{\hat{\theta}_{a_k^*}(\tau_s+1) > y_{e,k} \,\middle|\, \mathcal{F}_{\tau_s} = F_{\tau_s}\right\}}\right] \cdot \Delta_{e,k} \\
&= O\left(\frac{1}{L^2}\right) \quad .
\end{aligned} \tag{5.42}$$

We use the following arguments to complete the proof. Now, as we are tracking the number of observations of the optimal base arm a_k^*, i.e., we are tracking $T_{a_k^*}(t-1)$. For a fixed k, we arrange all the mean reward gaps $\Delta_{e,k}$ for all $e \in \bar{A}^*$ in a descending order $\Delta_{e_1,k} \geq \Delta_{e_2,k} \geq \ldots \geq \Delta_{e_{L-K},k} =: \Delta_{e_{\min},k}$.

For all the rounds when the counter is in the range of $\left[0, l^*_{e_1,k}\right]$, the total regret among all these rounds is at most $\Delta_{e_1,k} \cdot O\left(l^*_{e_1,k}\right)$. When the counter is in the range of $\left[l^*_{e_1,k}+1, l^*_{e_2,k}\right]$, the total regret is at most $\Delta_{e_2,k} \cdot O\left(l^*_{e_2,k} - l^*_{e_1,k}\right) + \Gamma_{2,k,e_1} = \Delta_{e_2,k} \cdot O\left(l^*_{e_2,k} - l^*_{e_1,k}\right) + O(1/K)$. Finally, when the counter is in the range of $\left[l^*_{e_{L-K-1},k}+1, l^*_{e_{L-K},k}\right]$, the total regret among all these rounds is at most $\Delta_{e_{L-K-1},k} \cdot O\left(l^*_{e_{L-K+1},k} - l^*_{e_{L-K},k}\right) + \sum_{q=1}^{L-K-1} \Gamma_{2,k,e_q} \leq \Delta_{e_{L-K-1},k} \cdot O\left(l^*_{e_{L-K},k} - l^*_{e_{L-K-1},k}\right) + O(1/K)$. For all the rounds after the counter hits $l^*_{e_{L-K},k}$, the total regret is at most $\sum_{k=1}^{K} O(1/L^2) = O(1/K)$.

By combining all these pieces together and using Lemma 19, we have

$$U = \sum_{k=1}^{K} O\left(\frac{\ln(Kn)}{\Delta_{k,\min}}\right) \quad . \tag{5.43}$$

■

Chapter 6

Experiments

6.1 Overview

We perform experiments in two different settings. In Section 6.1.1, we evaluate our algorithms on a synthetic dataset and in Section 6.1.2, we use a real-world movie-rating dataset with the purpose of recommending diverse and popular movies to users in a differentially private manner. To measure the performance of our algorithms we use the *expected per-round return* as suggested in Kveton, Wen, Ashkan, Eydgahi, et al. (2014). The expected per-round return in round s is computed as $\frac{1}{s}\sum_{t=1}^{s}\sum_{e\in A^t}\bar{w}(e)$.

For `DPUCB-MAT`, we compare its empirical performance with two baselines. The first baseline is the optimal return $f(A^*, \bar{w}) = \sum_{e\in A^*}\bar{w}(e)$ denoted as Optimal Policy in the plots. The second baseline is Optimistic Matroid Maximization (OMM) from Kveton, Wen, Ashkan, Eydgahi, et al. (2014), which is the non-private UCB1-based algorithm for matroid bandits. Similarly, for `DPTS-MAT`, we compare its empirical performance with the optimal return and the non-private Combinatorial Thompson Sampling (CTS) in Wang et al. (2018). Note that CTS was developed for the more general combinatorial bandits but can be easily adapted to matroid bandits. To have a fair empirical performance comparison, we adapt CTS to the matroid bandit setting by enforcing the matroid constraints. Also, we show the performance of our algorithms with different values of $\varepsilon \in \{10^5, 2, 10^{-4}\}$. Additionally, we show asymptotic regret growth and the variation of regret with $1/\varepsilon$ for both our algorithms.

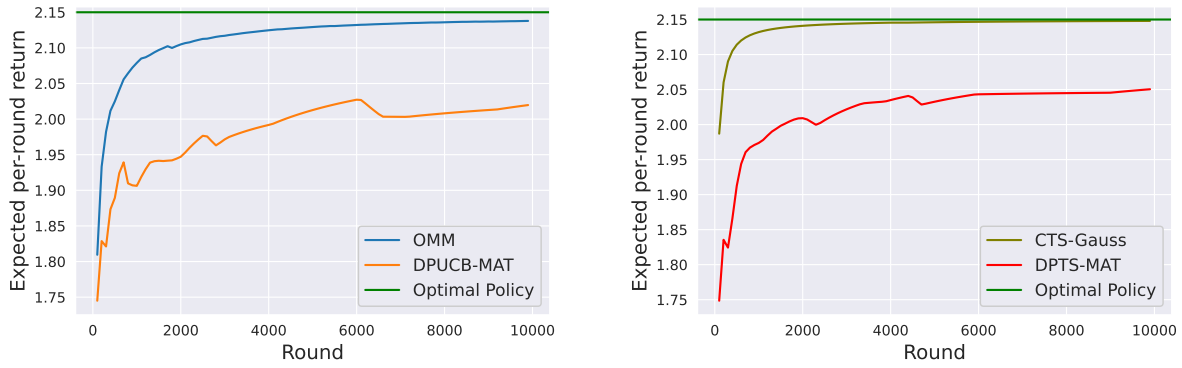
Base arm e	Mean reward $\bar{w}(e)$
(1, 0, 0)	0.80
(0, 1, 0)	0.75
(0, 0, 1)	0.60
(1, 0, 1)	0.20
(0, 1, 1)	0.30
(2, 0, 0)	0.40
(0, 0, 0)	0.70

Table 6.1: Synthetic dataset.

6.1.1 Synthetic Dataset

In this section, we report the experimental results of our proposed algorithms on a set of 3-dimensional vectors taken from Neel et al. (2009). We set $\varepsilon = 2$. The base arm set is $E = \{e_1, \ldots, e_7\}$ which is shown in the first column of Table 6.1. Here each base arm is a 3-dimensional vector in the Euclidean space. Table 6.1 also shows the mean rewards of all the base arms. Matroid independence is defined by the linear independence of the vectors. Since for a 3-dimensional space, all bases are of size 3, we have $|A^*| = |A^t| = 3$ for all t. From Table 6.1, we can also see that $A^* = \{e_1, e_2, e_3\}$. Since the total privacy budget is ε, the privacy budget for each $e \in A^t$ is $\varepsilon_0 = \varepsilon/3 = 2/3$. The reward $w_t(e)$ for each $e \in E$ is generated from a Bernoulli distribution with mean $\bar{w}(e)$. The total number of rounds of interaction is $n = 10,000$. The experimental results for `DPUCB-MAT`, OMM, and Optimal Policy are presented in Figure 6.1a. From the results, we can see that `DPUCB-MAT` and OMM have a similar growth rate in terms of expected per-round return. The results for `DPTS-MAT`, CTS, and Optimal Policy also show similar trends and are shown in Figure 6.1b.

Effect of large and small ε's. Figure 6.2 shows the expected per-round return for different values of the privacy parameter $\varepsilon \in \{10^5, 2, 10^{-4}\}$. In Figure 6.2a, we show the performance of `DPUCB-MAT` (Algorithm 2). We observe that when ε decreases, the performance

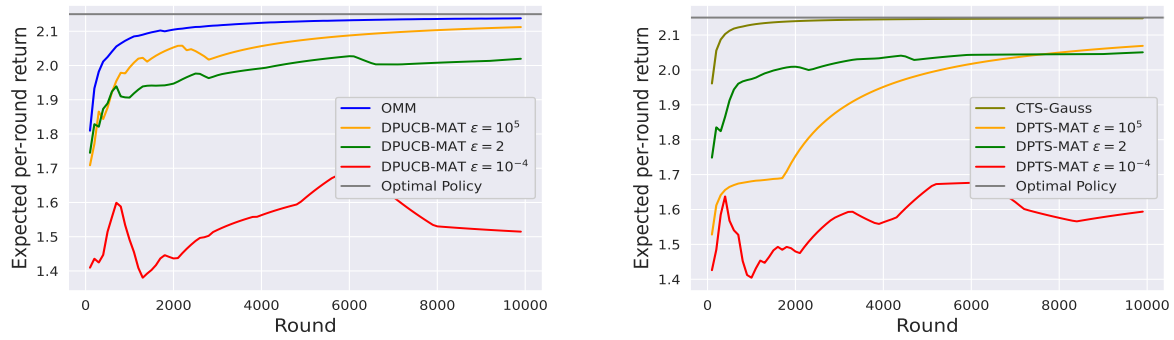


(a) `DPUCB-MAT`. (b) `DPTS-MAT`.

Figure 6.1: **Performance on the synthetic dataset.** Figure 6.1a compares the performance of `DPUCB-MAT` (Algorithm 2) against the non-private OMM and the Optimal Policy. Figure 6.1b shows the performance of `DPTS-MAT` (Algorithm 5 against the non-private CTS (with Gaussian prior and likelihood) and the Optimal Policy. Even after adding differential privacy, both our algorithms do not perform much worse than their non-private counterparts.

of `DPUCB-MAT` deteriorates, and when ε increases, the performance of `DPUCB-MAT` becomes better. This is expected as a good differentially private learning algorithm should balance the privacy and regret guarantees. When ε is too small (10^{-4}) the scale of the Laplace noise added becomes too large. Hence, too much noise is added which deteriorates the algorithm's performance. We also observe that when the privacy parameter ε is large (10^5), i.e., in the non-private regime, the performance of `DPUCB-MAT` approaches that of the non-private OMM. This is also expected as in the non-private regime, we do not pay any price for preserving privacy. Similar trends can also be seen for `DPTS-MAT` (Algorithm 5) in Figure 6.2b.

Other experiments. In Figure 6.3a, we plot the asymptotic growth of the regret for the synthetic dataset. Figure 6.3a shows the accumulated regret R_n till round n over $\ln(n)$ for $n = 1, 2, \ldots, 10^6$ rounds for both our private algorithms `DPUCB-MAT` and `DPTS-MAT` on the synthetic dataset. The quantity $\lim_{n\to\infty} R_n/\ln(n)$ characterizes the asymptotic rate of growth of the regret (Lai et al., 1985). We observe that both `DPUCB-MAT` and `DPTS-MAT` converge but with different rates. In addition, we also observe that the Thompson Sampling-based algorithm, `DPTS-MAT`, empirically outperforms the UCB-based algorithm `DPUCB-MAT`, since



(a) `DPUCB-MAT`. (b) `DPTS-MAT`.

Figure 6.2: **Effect of large and small ε's**. Figure 6.2a shows the performance of `DPUCB-MAT` (Algorithm 2) for different values of ε on the synthetic dataset. We observe that the performance of `DPUCB-MAT` decreases as the value of ε decreases. We also observe that as ε increases, we approach the non-private regime and the performance of `DPUCB-MAT` approaches the performance of the non-private OMM. Figure 6.2b shows similar trends for `DPTS-MAT` (Algorithm 5).

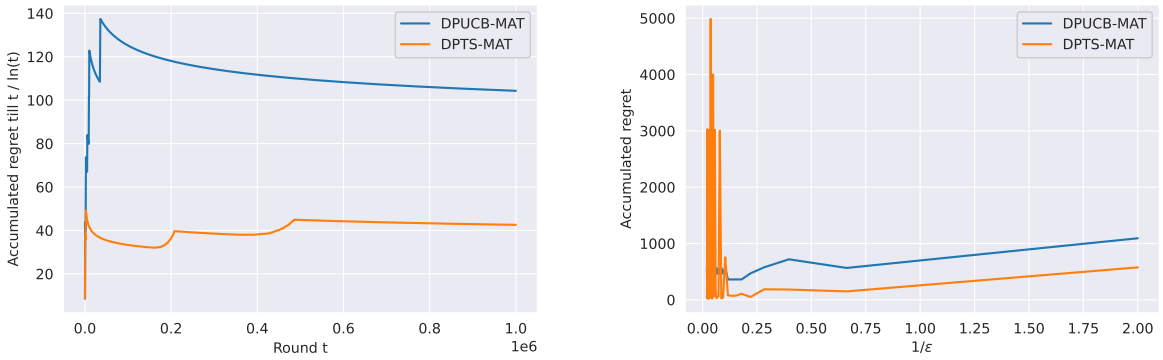
it converges to a smaller value.

Figure 6.3b studies the relationship between the accumulated regret R_n and $1/\varepsilon$ for `DPUCB-MAT` and `DPTS-MAT` on the synthetic dataset. The values of ε are 50 evenly spaced numbers between 0.5 and 50. We run our algorithms for $n = 10,000$ rounds for each value of ε and plot the accumulated regret. From the experimental results, we can see that for both the algorithms the regret trend is linear in $1/\varepsilon$ and the Thompson Sampling-based algorithm empirically outperforms the UCB-based one.

6.1.2 Movie Rating Dataset

In this experiment, we learn to recommend a set of *diverse* and *popular* movies with differential privacy from the MovieLens dataset (Harper et al., 2015). The experiment design is adopted from Kveton, Wen, Ashkan, Eydgahi, et al. (2014). We report experimental results with the total privacy budget $\varepsilon = 2$. The total number of rounds is $n = 20,000$.

The entire dataset contains 1 million ratings from 6040 users. The total number of movies is 3883 from 18 different genres. To recommend popular movies, we select 100 movies that have received the most ratings. These 100 movies constitute the base arm set E of a matroid.



(a) Asymptotic regret rate.

(b) Regret versus $1/\varepsilon$.

Figure 6.3: **Regret of `DPUCB-MAT` and `DPTS-MAT` on the synthetic dataset**. Figure 6.3a shows the accumulated regret R_n till round n divided by $\ln(n)$ for $n = 1, 2, \ldots, 10^6$ rounds for both `DPUCB-MAT` and `DPTS-MAT`. The expression $\lim_{n\to\infty} R_n/\ln(n)$ characterizes the asymptotic growth rate of the regret. Figure 6.3b shows the accumulated regret versus $1/\varepsilon$ for both `DPUCB-MAT` and `DPTS-MAT`.

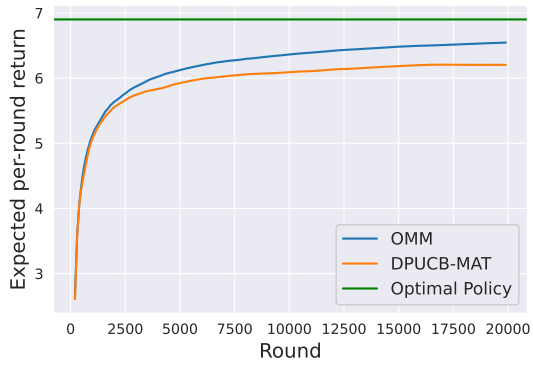
To form an independent set A of the matroid, we construct a binary feature vector u_e for each movie $e \in E$, which denotes the genres of that movie. If the feature vectors u_e for all $e \in A$ are linearly independent, all the movies in A form an independent set, which further indicates that these movies are *diverse*. The expected reward $\bar{w}(e)$ for each movie e is given by the total number of ratings for e divided by the total number of users in the dataset. The optimal solution A^* is computed greedily with respect to $\bar{w}$ using Algorithm 1. In each round, we recommend 17 movies, i.e., $|A^*| = |A^t| = 17$. Since $\varepsilon = 2$, the privacy budget for each $e \in A^t$ is $\varepsilon_0 = \varepsilon/17 = 2/17$. The randomness comes from the fact that in each round t, a random user is selected. For each movie $e \in A^t$, if e is rated by that selected random user, then the reward $w_t(e)$ is set to 1 otherwise 0.

Figure 6.4a shows the results for `DPUCB-MAT`, OMM, and Optimal Policy. We observe that the expected per-round return of `DPUCB-MAT` is comparable to that of the non-private baseline (OMM) and is also close to that of the Optimal Policy. We see similar results for `DPTS-MAT` as shown in Figure 6.4b. Table 6.2 lists the overlapping movies learned by `DPUCB-MAT` and Optimal Policy at the end of all the rounds of interaction. We can see that `DPUCB-MAT` recommends a lot of the same movies as the Optimal Policy and the movie genres

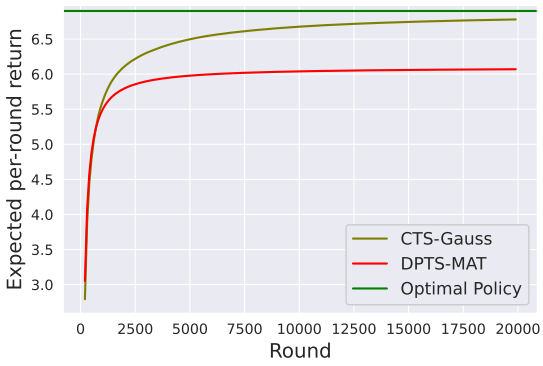
Movie e	$\bar{w}(e)$	Genres
American Beauty	0.568	Comedy, Drama
Star Wars: Episode IV	0.496	Action, Adventure, Fantasy, Sci-Fi
Star Wars: Episode VI	0.478	Action, Adventure, Romance, Sci-Fi, War
Saving Private Ryan	0.440	Action, Drama, War
Men in Black	0.420	Action, Adventure, Comedy, Sci-Fi
L.A. Confidential	0.379	Crime, Film-Noir, Mystery, Thriller
Ghostbusters	0.361	Comedy, Horror
The Wizard of Oz	0.285	Animation, Children's, Comedy, Musical

Table 6.2: Movies recommended by `DPUCB-MAT` that overlap with movies in A^* after 20k rounds.

of those movies appear to be diverse. This validates our modeling choice of using matroids to recommend movies with diverse movie genres.



(a) `DPUCB-MAT`



(b) `DPTS-MAT`

Figure 6.4: **Performance on the movie rating dataset.** Figure 6.4a and Figure 6.4b show the performance of both of our algorithms on the movie rating dataset to recommend diverse and popular movies. In Figure 6.4a, we can see that the `DPUCB-MAT` algorithm's performance is close to that of the non-private OMM and the Optimal Policy. Similar results can be seen for `DPTS-MAT` in Figure 6.4b.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we explored how we can efficiently and privately learn to choose items in a sequential and adaptive manner. In particular, we looked at the setting of matroid bandits, where the learner selects base arms or a super arm in each round of interaction with an environment, and improves on the selection based on the reward feedback. The items chosen at each step satisfy the matroid independence constraints. The goal of the learner was to maximize the total cumulative reward over all rounds of interaction, or equivalently, minimize regret. In applications like movie recommendations, the reward feedback might represent sensitive user data that we wish to keep private. For that purpose, we looked into differential privacy and defined differential privacy for matroid bandits.

We took differential privacy into account and showed that the problem of minimizing regret to learn a super arm (also known as the maximum weight basis) for matroid bandits can be solved efficiently. We proposed two simple differentially private algorithms `DPUCB-MAT` and `DPTS-MAT`. We showed differential privacy guarantees and logarithmic regret upper bounds for both algorithms. To achieve the regret bound, we decomposed the regret by introducing a round-dependent permutation π_t that helps in mapping a suboptimal base arm to an optimal base arm. The permutation π_t helps us decompose the regret into K stochastic bandit problems. Finally, we conducted experiments to evaluate our algorithms' empirical performance on a synthetic and a real-world movie rating dataset to recommend movies.

7.2 Future Directions

For future work, there are some open problems remaining in this line of work. To the best of our knowledge, there is no lower bound for matroid bandits with differential privacy. By modifying the regret lower bound for differentially private combinatorial bandits as shown by Chen et al. (2020, Theorem 9), we conjecture a regret lower bound of $\Omega\left(L\ln(n)/\Delta + LK\ln(n)/\varepsilon\right)$ for differentially private matroid bandits. However, our upper bound (Theorem 9) is still an extra $\min\{K, \ln(n)\}$ factor far from this conjectured regret lower bound. We are not sure yet whether our derived regret bound is not tight or whether a better regret lower bound exists for differentially private matroid bandits.

More broadly, for the general differentially private combinatorial bandits with no matroid constraints (Chen et al., 2020; Kveton, Wen, Ashkan, and Szepesvari, 2015), the regret upper bounds are still suboptimal (for example, Chen et al. (2020, Theorem 8)) and a tighter analysis is needed. Moreover, there are other similar settings like bandits with graph-structured feedback (Alon et al., 2017) where there are no differentially private algorithms yet. Ideas developed in this thesis can be applied to get differentially private algorithms for such combinatorial/graph-structured settings.

Bibliography

Agrawal, Shipra and Navin Goyal (2017). "Near-Optimal Regret Bounds for Thompson Sampling". In: *Journal of the ACM*, pp. 1–24.

Alon, Noga, Nicolo Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir (2017). "Nonstochastic Multi-Armed Bandits with Graph-Structured Feedback". In: *SIAM Journal on Computing*, pp. 1785–1826.

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer (2002). "Finite-Time Analysis of the Multiarmed Bandit Problem". In: *Machine learning*, pp. 235–256.

Azize, Achraf and Debabrota Basu (2022). "When Privacy Meets Partial Information: A Refined Analysis of Differentially Private Bandits". In: *Advances in Neural Information Processing Systems*, pp. 32199–32210.

Chan, T-H Hubert, Elaine Shi, and Dawn Song (2011). "Private and Continual Release of Statistics". In: *ACM Transactions on Information and System Security*, pp. 1–24.

Chapelle, Olivier and Lihong Li (2011). "An Empirical Evaluation of Thompson Sampling". In.

Chen, Xiaoyu, Kai Zheng, Zixin Zhou, Yunchang Yang, Wei Chen, and Liwei Wang (2020). "(Locally) Differentially Private Combinatorial Semi-bandits". In: *International Conference on Machine Learning*, pp. 1757–1767.

Dwork, Cynthia (2006). "Differential Privacy". In: *International Colloquium on Automata, Languages, and Programming*, pp. 1–12.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith (2006). "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*, pp. 265–284.

Dwork, Cynthia and Moni Naor (2010). "On the Difficulties of Disclosure Prevention in Statistical Databases or The Case for Differential Privacy". In: *Journal of Privacy and Confidentiality*, pp. 93–107.

Dwork, Cynthia, Moni Naor, Toniann Pitassi, and Guy N Rothblum (2010). "Differential Privacy Under Continual Observation". In: *ACM symposium on Theory of computing*, pp. 715–724.

Dwork, Cynthia and Aaron Roth (2014). "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science*, pp. 211–407.

Edmonds, Jack (1965). "Minimum Partition of a Matroid Into Independent Subsets". In: *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, pp. 67–72.

Garivier, Aurélien and Olivier Cappé (2011). "The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond". In: *Conference on Learning Theory*, pp. 359–376.

Geng, Quan and Pramod Viswanath (2014). "The Optimal Mechanism in Differential Privacy". In: *International Symposium on Information Theory*, pp. 2371–2375.

Harper, F. Maxwell and Joseph A. Konstan (2015). *The MovieLens Datasets: History and Context*.

Hu, Bingshan and Nidhi Hegde (2022). "Near-Optimal Thompson Sampling-based Algorithms for Differentially Private Stochastic Bandits". In: *Uncertainty in Artificial Intelligence*, pp. 844–852.

Hu, Bingshan, Zhiming Huang, and Nishant A Mehta (2021). "Optimal Algorithms for Private Online Learning in a Stochastic Environment". In: *arXiv preprint arXiv:2102.07929*.

Kairouz, Peter, Sewoong Oh, and Pramod Viswanath (2015). "The Composition Theorem for Differential Privacy". In: *International conference on machine learning*, pp. 1376–1385.

Kaufmann, Emilie, Nathaniel Korda, and Rémi Munos (2012). "Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis". In: *Algorithmic Learning Theory*, pp. 199–213.

Kveton, Branislav, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson (2014). "Matroid Bandits: Fast Combinatorial Optimization with Learning". In: *Uncertainty in Artificial Intelligence*, pp. 420–429.

Kveton, Branislav, Zheng Wen, Azin Ashkan, and Csaba Szepesvari (2015). "Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits". In: *International Conference on Artificial Intelligence and Statistics*, pp. 535–543.

Lai, T.L and Herbert Robbins (1985). "Asymptotically Efficient Adaptive Allocation Rules". In: *Advances in Applied Mathematics*, pp. 4–22.

Mishra, Nikita and Abhradeep Thakurta (2015). "(Nearly) Optimal Differentially Private Stochastic Multi-Arm Bandits". In: *Uncertainty in Artificial Intelligence*, pp. 592–601.

Narayanan, Arvind and Vitaly Shmatikov (2008). "Robust De-anonymization of Large Sparse Datasets". In: *IEEE Symposium on Security and Privacy*, pp. 111–125.

Neel, David L and Nancy Ann Neudauer (2009). "Matroids You Have Known". In: *Mathematics magazine*, pp. 26–41.

Talebi, Mohammad Sadegh and Alexandre Proutiere (2016). "An Optimal Algorithm for Stochastic Matroid Bandit Optimization". In: *International Conference on Autonomous Agents & Multiagent Systems*, pp. 548–556.

Wang, Siwei and Wei Chen (2018). "Thompson Sampling for Combinatorial Semi-Bandits". In: *International Conference on Machine Learning*, pp. 5114–5122.

Wasserman, Larry and Shuheng Zhou (2010). "A Statistical Framework for Differential Privacy". In: *Journal of the American Statistical Association*, pp. 375–389.

Whitney, Hassler (1935). "On the Abstract Properties of Linear Dependence". In: *American Journal of Mathematics*, pp. 509–533.