Stability in Online Learning: From Random Perturbations in Bandit Problems to Differential Privacy

by

Baekjin Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2021

Doctoral Committee:

Associate Professor Ambuj Tewari, Chair
Professor Yaacov Ritov
Associate Professor Eric Schwartz
Professor Stilian A. Stoev

Baekjin Kim

baekjin@umich.edu

ORCID iD: 0000-0002-0175-3325

DEDICATION

For my wife, Keejeong

ACKNOWLEDGMENTS

I was incredibly fortunate to have a great advisor, Ambuj Tewari. He is amazingly smart with an outstanding insight and passion as well as a really sweet person who is always caring and cheerful. Without his intellectual guidance, emotional support, and patience I would never have completed this journey over last five years. I am also grateful to Yaacov Ritov, Eric Schwartz, and Stillian Stoev for serving on my committee.

Most importantly, Keejeong, I cannot thank you enough for your encouraging me and believing in me, and without you none of what follows would exist. This dissertation is the result of our joint endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

Online learning is an area of machine learning that studies algorithms that make sequential predictions on data arriving incrementally. In this thesis, we investigate *stability of online learning algorithms* in two different settings. First, we examine *random perturbation* methods as a source of stability in *bandit problems*. Second, we study stability as a key concept connecting online learning and *differential privacy*.

The first two chapters study the statistical properties of the perturbation technique in both stochastic and adversarial *multi-armed bandit problems*. We provide the first general analysis of perturbations for the stochastic multi-armed bandit problem. We also show that the open problem regarding minimax optimal perturbations for adversarial bandits cannot be solved in two ways that might seem very natural.

The next two chapters consider stationary and non-stationary stochastic *linear bandits* respectively. We develop two randomized exploration strategies: (1) by replacing optimism with a simple randomization when deciding a confidence level in optimism based algorithms, or (2) by directly injecting the random perturbations to current estimates to overcome the *conservatism* that optimism based algorithms generally suffer from. Furthermore, we study the statistical and computational aspects of both of these strategies.

While at a first glance it may seem that *online learning* and *differential privacy* have little in common, there is a strong connection between them via the notion of stability since the definition of differential-privacy is at its core, a form of stability. The final chapter investigates whether the recently established equivalence between online and private learnability in binary classification extends to multi-class classification and regression.

CHAPTER 1

Introduction

Online learning is an area of machine learning that studies algorithms that make sequential predictions on data arriving incrementally. This field is distinguished from standard *batch learning* where the entire training instances are given to optimize the model. It is more challenging to provide theoretical guarantees for online learning algorithms as the model keeps changing with the data observed in a sequential fashion. In this thesis, we study the *stability of online learning algorithms* in two different settings. First, we examine *random perturbation* methods as a source of stability in *bandit problems*. Bandit problems are a special and more challenging case of online learning in that no information is provided on the rewards of alternative options in bandit setting. Second, we study stability as a key concept connecting online learning and *differential privacy*.

Stability has been one of the major topics of interest in machine learning. The well-known "No Free Lunch" theorems in mathematical learning theory show that one cannot derive formal guarantees for learning algorithms without imposing some prior assumptions. The mathematical reason for this is most learning problems are *ill-posed*, i.e., whatever is being learned is not uniquely identified by the data. *Regularization* is a classical approach to stabilizing and solving ill-posed inverse problems by adding a penalty function to an optimization problem to encourage simpler solutions. The alternative stabilization technique, which has become essential in modern applications of machine learning is via controlled injection of *random perturbations* into the learning process. Especially in online learning, stability arises as a central motif in two major families of online learning algorithms, Follow-the-regularized-leader and Follow-the-perturbed-leader.

In standard methods in differential privacy such as Exponential and Gaussian mechanisms, random perturbations also play a key role in designing differential private algorithms as they act as a source of stability in optimal online learning algorithms. *Differential privacy* was introduced to study data analysis mechanism that do not reveal too much information on any single sample in *batch learning*. Although two subjects originated from

essentially different learning frameworks, *stability* has been recently regarded as the concept of connecting online learning and differential privacy in the sense that the definition of differential privacy is in and of itself a form of stability.

1.1 Multi-armed bandits

Beginning with the seminal work of Hannan (1957), researchers have been interested in algorithms that use *random perturbations* to generate a distribution over available actions. Kalai and Vempala (2005) showed that the perturbation idea leads to efficient algorithms for many online learning problems with large action sets. Due to the *Gumbel lemma* (Hazan et al., 2017), the well known exponential weights algorithm (Freund and Schapire, 1997) also has an interpretation as a perturbation based algorithm that uses Gumbel distributed perturbations.

There have been several attempts to analyze the regret of perturbation based algorithms with specific distributions such as uniform, double-exponential, drop-out and random walk (see, e.g., (Kalai and Vempala, 2005; Kujala and Elomaa, 2005; Devroye et al., 2013; Van Erven et al., 2014)). These works provided rigorous guarantees but the techniques they used did not generalize to general perturbations. Recent work (Abernethy et al., 2014) provided a general framework to understand general perturbations and clarified the relation between regularization and perturbation by understanding them as different ways to smooth an underlying non-smooth potential function.

Abernethy et al. (2015) extended the analysis of general perturbations to the partial information setting of the adversarial multi-armed bandit problem. They isolated *bounded hazard rate* as an important property of a perturbation and gave several examples of perturbations that lead to the near optimal regret bound of $O(\sqrt{KT \log K})$. Since Tsallis entropy regularization can achieve the minimax regret of $O(\sqrt{KT})$ (Audibert and Bubeck, 2009, 2010), the question of whether perturbations can match the power of regularizers remained open for the adversarial multi-armed bandit problem.

1.2 Linear bandits

A multi-armed bandit is the simplest model of decision making that involves the exploration versus exploitation trade-off (Lai and Robbins, 1985). Linear bandits are an extension of multi-armed bandits where the reward has linear structure with a finite-dimensional feature associated with each arm (Abe et al., 2003; Dani et al., 2008). Two standard exploration strategies in stochastic linear bandits are upper confidence bound algorithm (Lin-

UCB) (Abbasi-Yadkori et al., 2011) and linear Thomson sampling (LinTS) (Agrawal and Goyal, 2013b). The former relies on optimism in face of uncertainty and is a deterministic algorithm built upon the construction of a high-probability confidence ellipsoid for the unknown parameter vector. The latter is a Bayesian solution that maximizes the expected rewards according to a parameter sampled from the posterior distribution. Chapelle and Li (2011) showed that linear Thompson sampling empirically performs better and is more robust to corrupted or delayed feedback than LinUCB. From a theoretical perspective, it enjoys a regret bound that is a factor of $\sqrt{d}$ worse than minimax-optimal regret bound $\tilde{\Theta}(d\sqrt{T})$ that LinUCB enjoys. However, the minimax optimality of optimism comes at a cost: implementing UCB type algorithms can lead to NP-hard optimization problems even for convex action sets (Agrawal, 2019).

Abeille et al. (2017) viewed linear Thompson sampling as a perturbation based algorithm, characterized a family of perturbations whose regrets can be analyzed, and raised an open problem to find a minimax-optimal perturbation. In addition to its significant role in smartly balancing exploration with exploitation, a perturbation based approach to linear bandits also reduces the problem to one call to the offline optimization oracle in each round. Recent works (Kveton et al., 2019, 2020) have proposed randomized algorithms that use perturbation as a means to achieve oracle-efficient computation as well as better theoretical guarantee than LinTS, but there is still a gap between their regret bounds and the lower bound of $\Omega(d\sqrt{T})$. This gap is logarithmic in the number of actions which can introduce extra dependence on dimension for large action spaces.

A new randomized exploration scheme was proposed in the recent work of Vaswani et al. (2020). In contrast to Hannan's perturbation approach that injects perturbation directly into an estimate, they replace optimism with random perturbation when using confidence sets for action selection in optimism based algorithms. This approach can be broadly applied to multi-armed bandit and structured bandit problems, and the resulting algorithms are theoretically optimal and empirically perform well since overall conservatism of optimism based algorithms can be tackled by randomizing the confidence level.

Linear bandit problems were originally motivated by applications such as online ad placement with features extracted from the ads and website users. However, users' preferences often evolve with time, which leads to interest in the non-stationary variant of linear bandits. Accordingly, adaptive algorithms that accommodate time-variation of environments have been studied in a rich line of works in both multi-armed bandit (Besbes et al., 2014) and linear bandit. In the pioneering work of Cheung et al. (2019), the authors proposed the SW-LinUCB algorithm to deal with changing environment via optimism based approach with sliding-window least square estimator, and proved its dynamic regret bound

of order $\tilde{O}(d^{2/3}B_T^{1/3}T^{2/3})$. In later studies, Russac et al. (2019) developed D-LinUCB based on the weighted least square estimator, and Zhao et al. (2020) designed Restart-LinUCB and showed that this simple restarted strategy is sufficient to achieve the same dynamic regret guarantee. Recently, Zhao and Zhang (2021) pointed out that the proof of a key lemma in previous analysis (Cheung et al. (2019), Lemma 3) has serious technical flaw, provided a fix for the analysis, and proved a new and slightly suboptimal dynamic regret of order $\tilde{O}(d^{7/8}B_T^{1/4}T^{3/4})$ for the three main algorithms (Cheung et al., 2019; Russac et al., 2019; Zhao et al., 2020).

Luo et al. (2018) and Chen et al. (2019) studied fully adaptive and oracle-efficient algorithms assuming access to an optimization oracle when total variation is unknown for the learner. It is still open problem to design a practically simple, oracle-efficient and statistically optimal algorithm for non-stationary linear bandits. Since existing non-stationary linear bandit algorithms are suboptimal in dynamic regret and non-randomized built upon the optimism in the face of uncertainty, it remains open to construct the randomized exploration methods, which might be able to enjoy computational efficiency as well as statistical guarantee.

1.3 Differential privacy

Online learning and *differentially-private (DP) learning* have been well-studied in the machine learning literature. While these two subjects are seemingly unrelated, recent papers have revealed a strong connection between online and private learnability via the notion of *stability* (Abernethy et al., 2019; Agarwal and Singh, 2017; Gonen et al., 2019). The notion of differential privacy is, at its core, less about privacy and more about algorithmic stability since the output distribution of a DP algorithm should be robust to small changes in the input. Stability also plays a key role in developing online learning algorithms such as follow-the-perturbed-leader (FTPL) and follow-the-regularized-leader (FTRL) (Abernethy et al., 2014).

Recently Alon et al. (2019) and Bun et al. (2020) showed that online learnability and private PAC learnability are equivalent in binary classification. Alon et al. (2019) showed that private PAC learnability implies finite Littlestone dimension (Ldim) in two steps; (i) every approximately DP learner for a class with Ldim d requires $\Omega(\log^* d)$ thresholds (see Section 2.6 for the definition of $\log^*$), and (ii) the class of thresholds over $\mathbb{N}$ cannot be learned in a private manner. Bun et al. (2020) proved the converse statement via a notion of algorithmic stability, called *global stability*. They showed (i) every class with finite Ldim can be learned by a globally-stable learning algorithm and (ii) they use global stability to

derive a DP algorithm. In this work, we investigate whether this equivalence extends to multi-class classification (MC) and regression, which is one of open questions raised by Bun et al. (2020).

In general, online learning and private learning for MC and regression have been less studied. In binary classification without considering privacy, the Vapnik-Chervonenkis dimension (VCdim) of hypothesis classes yields tight sample complexity bounds in the batch learning setting, and Littlestone (1988) defined Ldim as a combinatorial parameter that was later shown to fully characterize hypothesis classes that are learnable in the online setting Ben-David et al. (2009). Until recently, however, it was unknown what complexity measures for MC or regression classes characterize online or private learnability. Daniely et al. (2015) extended the Ldim to the MC setting, and Rakhlin et al. (2015) proposed the sequential fat- shattering dimension, an online counterpart of the fat-shattering dimension in the batch setting (Bartlett et al., 1996).

Related works DP has been extensively studied in the machine learning literature (Dwork and Lei, 2009; Dwork et al., 2014; Sarwate and Chaudhuri, 2013). Private PAC and agnostic learning were formally studied in the seminal work of Kasiviswanathan et al. (2011), and the sample complexities of private learners were characterized in the later work of Beimel et al. (2013).

Dwork et al. (2014) identified stability as a common factor of learning and differential privacy. Abernethy et al. (2019) proposed a DP-inspired stability-based methodology to design online learning algorithms with excellent theoretical guarantees, and Agarwal and Singh (2017) showed that stabilization techniques such as regularization or perturbation in online learning preserve DP. Feldman and Xiao (2014) relied on communication complexity to show that every purely DP learnable class has a finite Ldim. Purely DP learnability is a stronger condition than online learnability, which means that there exist online learnable classes that are not purely DP learnable. More recently, Alon et al. (2019) and Bun et al. (2020) established the equivalence between online and private learnability in a non-constructive manner. Gonen et al. (2019) derived an efficient black-box reduction from purely DP learning to online learning.

1.4 Thesis structure and contribution

The core material in this thesis is contained in five chapters. Each of these chapters has been adapted from publications:

- The results of Chapter 3 and 4 have been published in the electronic proceedings of the *Neural Information Processing Systems Conference* (Kim and Tewari, 2019).
- The results of Chapter 5 and 6 have been published in the electronic proceedings of the *Conference on Uncertainty in Artificial Intelligence* (Kim and Tewari, 2020).
- The results of Chapter 7 have been published in the electronic proceedings of the *Neural Information Processing Systems Conference* (Jung et al., 2020).

We build Chapter 3 and 4 upon previous works (Abernethy et al., 2014, 2015) in two distinct but related directions.

In Chapter 3, we provide the first unified regret analysis for perturbation algorithms in the *stochastic* multi-armed bandit problem. Our regrets are instance optimal for sub-Weibull perturbations with parameter 2 (with a matching lower tail bound), and all bounded support perturbations where there is sufficient probability mass at the extremes of the support. Our analysis relies on the simple but powerful observation that Thompson sampling with Gaussian priors and rewards can also be interpreted as a perturbation algorithm with Gaussian perturbations. We are able to generalize both the upper bound and lower bound of Agrawal and Goyal (2013a) in two respects; (1) from the special Gaussian perturbation to general sub-Weibull or bounded perturbations, and (2) from the special Gaussian rewards to general sub-Gaussian rewards.

In Chapter 4, we study the open problem of developing a perturbation based algorithm that gives us minimax optimality. We do not resolve it but provide rigorous proofs that *there are barriers to two natural approaches to solving the open problem*. (A) One cannot simply find a perturbation that is exactly equivalent to Tsallis entropy. This is surprising since Shannon entropy does have an exact equivalent perturbation, viz. Gumbel. (B) One cannot simply do a better analysis of perturbations used before (Abernethy et al., 2015) and plug the results into their general regret bound to eliminate the extra $O(\sqrt{\log K})$ factor. In proving the first barrier, we use a fundamental result in discrete choice theory. For the second barrier, we rely on tools from extreme value theory.

In Chapter 5, we explicate, in the simpler stationary setting, the role of two perturbation approaches in overcoming conservatism that UCB-type algorithms chronically suffer from in practice. In one approach, we replace optimism with a simple randomization when using confidence sets. In the other, we add random perturbations to the current estimate before maximizing the expected reward. These two approaches result in randomized LinUCB and Gaussian linear Thompson sampling for stationary linear bandits. We highlight the statistical optimality versus oracle efficiency trade-off between them.

In Chapter 6, we study the non-stationary environment and present two randomized algorithms with exponential discounting weights, discounted randomized LinUCB (D-RandLinUCB) and discounted linear Thompson sampling (D-LinTS) to gracefully adjust to the time-variation in the true parameter. We explain the trade-off between statistical guarantee and oracle efficiency in that the former asymptotically achieves the same dynamic regret bound as optimism based algorithms, but the latter enjoys computational efficiency due to sole reliance on an offline optimization oracle for large or infinite action set.

We build Chapter 7 upon previous works (Alon et al., 2019; Bun et al., 2020) which recently showed that online learnability and private PAC learnability are equivalent in binary classification. In Chapter 7, we investigate whether this equivalence extends to multi-class classification and regression. First, we show that private learnability implies online learnability in both settings. Our extension involves studying a novel variant of the Littlestone dimension that depends on a tolerance parameter and on an appropriate generalization of the concept of threshold functions beyond binary classification. Second, we show that while online learnability continues to imply private learnability in multi-class classification, current proof techniques encounter significant hurdles in the regression setting. While the equivalence for regression remains open, we provide non-trivial sufficient conditions for an online learnable class to also be privately learnable.

CHAPTER 2

Preliminaries

In this chapter, we present all preliminary materials including basic problem setups for several bandit problems and foundations for learning theory.

2.1 Multi-armed bandit problems

In every round t starting at 1, a learner chooses an action $A_t \in [K] \triangleq \{1, \cdots, K\}$ out of K arms and the environment picks a response in the form of a real-valued reward vector $\mathbf{g}_t \in [0,1]^K$. While the entire reward vector $\mathbf{g}_t$ is revealed to the learner in the full information setting, the learner only receives a reward associated with his choice in the bandit setting, while any information on other arms is not provided. Thus, we denote the reward corresponding to his choice A_t as $X_t = g_{t,A_t}$.

In *stochastic* multi-armed bandit, the rewards $g_{t,i}$ are sampled i.i.d from a fixed, but unknown distribution with mean μ_i. *Adversarial* multi-armed bandit is more general in that all assumptions on how rewards are assigned to arms are dropped. It only assumes that rewards are assigned by an adversary before the interaction begins. Such an adversary is called an *oblivious adversary*. In both environments, the learner makes a sequence of decisions A_t based on history $\mathcal{H}_{t-1} = (A_1, X_1, \cdots, A_{t-1}, X_{t-1})$ to maximize the cumulative reward, $\sum_{t=1}^{T} X_t$.

As a measure of evaluating a learner, *Regret* is the difference between rewards the learner would have received had he played the best in hindsight, and the rewards he actually received. Therefore, minimizing the regret is equivalent to maximizing the expected cumulative reward. We consider the expected regret in adversarial setting and the pseudo regret in stochastic setting, respectively as

$$\mathrm{R}(T) = \mathrm{E}[\max_{i\in[K]} \sum_{t=1}^{T} g_{t,i} - \sum_{t=1}^{T} g_{t,A_t}], \text{ and } \mathrm{R}'(T) = T \cdot \max_{i\in[K]} \mu_i - \mathrm{E}[\sum_{t=1}^{T} X_t].$$

Note that two regrets are the same where an oblivious adversary is considered. An online algorithm is called a *no-regret algorithm* if for every adversary, the expected regret with respect to every action A_t is sub-linear in T. Thus, it is of main interest in online learning to study the rate of growth of regret for various algorithms in various environments.

We use follow-the-perturbed-leader (FTPL) to denote families of algorithms for both stochastic and adversarial settings. The common core of FTPL algorithms consists in adding random perturbations to the estimates of rewards of each arm prior to computing the current "the best arm" (or "leader"). However, the estimates used are different in the two settings: stochastic setting uses sample means and adversarial setting uses inverse probability weighted estimates.

For details for bandit problems, there are several references on bandit theory, and we provide a few; Bubeck et al. (2012); Slivkins (2019) and Lattimore and Szepesvári (2020).

2.2 Stochastic linear bandit problems

Linear bandits are an extension of multi-armed bandits where the reward has linear structure with a finite-dimensional feature associated with each arm. Then an agent repeatedly makes decisions based on user or patient information with the goal of maximizing cumulative rewards. Such problems have numerous applications including online personalization for recommendation systems (Li et al., 2010) and advertisement placement (Chapelle et al., 2014), mobile health (Tewari and Murphy, 2017), adaptive clinical trials (Woodroofe, 1979), and dynamic pricing (Besbes and Zeevi, 2009). For instance, in online personalization problems, we might serve content based on user history and demographic information with the goal of maximizing user engagement with this recommendation service.

In this section, we introduce both stationary and non-stationary setup for stochastic linear bandit problems.

2.2.1 Stationary setup

In *stationary* stochastic linear bandit, a learner chooses an action X_t from a given action set $\mathcal{X}_t \subset \mathbb{R}^d$ in every round t, and he subsequently observes a reward $Y_t = \langle X_t, \theta^\star \rangle + \eta_t$ where $\theta^\star \in \mathbb{R}^d$ is an unknown parameter and η_t is a conditionally 1-subGaussian random variable. For simplicity, assume that $\|\theta^\star\|_2 \leq 1$ and, for all $x \in \mathcal{X}_t$, $\|x\|_2 \leq 1$, and thus $|\langle x, \theta^\star \rangle|_2 \leq 1$.

As a measure of evaluating a learner, the regret is defined as the difference between rewards the learner would have received had it played the best in hindsight, and the rewards

actually received. Therefore, minimizing the regret is equivalent to maximizing the expected cumulative reward. Denote the best action in a round t as $x_t^\star = \arg\max_{x \in \mathcal{X}_t} \langle x, \theta^\star \rangle$ and the expected regret as

$$E[R(T)] = \mathbb{E}\big[\sum_{t=1}^{T} [\langle x_t^\star, \theta^\star \rangle - \langle X_t, \theta^\star \rangle]\big].$$

2.2.2 Non-stationary setup

In each round $t \in [T]$, an action set $\mathcal{X}_t \in \mathbb{R}^d$ is given to the learner and it has to choose an action $X_t \in \mathcal{X}_t$. Then, the reward $Y_t = \langle X_t, \theta_t^\star \rangle + \eta_t$ is observed to the learner where $\theta_t^\star \in \mathbb{R}^d$ is an unknown time-varying parameter and η_t is a conditionally 1-subGaussian random variable. The *non-stationary* assumption allows unknown parameter $\theta_t^\star$ to be time-variant within total variation budget $B_T = \sum_{t=1}^{T-1} \|\theta_t^\star - \theta_{t+1}^\star\|_2$. It is a nice way of quantifying time-variations of $\theta_t^\star$ in that it covers both slowly-changing and abruptly-changing environments. For simplicity, assume $\|\theta_t^\star\|_2 \leq 1$, for all $x \in \mathcal{X}_t$, $\|x\|_2 \leq 1$, and thus $|\langle x, \theta_t^\star \rangle|_2 \leq 1$.

In a similar way to stationary setting, denote the best action in a round t as $x_t^\star = \arg\max_{x \in \mathcal{X}_t} \langle x, \theta_t^\star \rangle$ and denote the expected dynamic regret as

$$E[R(T)] = \mathbb{E}\big[\sum_{t=1}^{T} [\langle x_t^\star, \theta_t^\star \rangle - \langle X_t, \theta_t^\star \rangle]\big]$$

where X_t is chosen action at time t. The goal of the learner is to minimize the expected dynamic regret.

2.3 PAC learning

Though we consider the settings beyond binary classification such as multi-class classification and regression problems in chapter 7, the literature on PAC learning is vast and includes several books (Shalev-Shwartz and Ben-David, 2014; Vapnik, 2013). They provided details on learning framework, VC dimension, and PAC learnability for binary classification.

In multi-class classification problems with $K \geq 2$ classes, we let $\mathcal{X}$ be the input space and $\mathcal{Y} = [K] \triangleq \{1, 2, \cdots, K\}$ be the output space, and the *standard zero-one loss* $\ell^{0-1}(\hat{y}; y) = \mathbb{I}(\hat{y} \neq y)$ is considered.

The regression problem is similar to the classification problem, except that the label becomes continuous, $\mathcal{Y} = [-1, 1]$, and the goal is to learn a real-valued function $f : \mathcal{X} \to \mathcal{Y}$ that approximates well labels of future instances. We consider the *absolute*

loss $\ell^{abs}(\hat{y}; y) = |\hat{y} - y|$ in this setting. Results under the absolute loss can be generalized to any other Lipschitz losses with modified rates.

Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ be an output space, and $\mathcal{D}$ be an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. A *hypothesis* is a function mapping from $\mathcal{X}$ to $\mathcal{Y}$. The *population loss* of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ with respect to a loss function ℓ is defined by $\text{loss}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\ell\big(h(x); y\big)\big]$. We also define the *empirical loss* of a hypothesis h with respect to a loss function ℓ and a sample $S = \big((x_i, y_i)\big)_{1:n}$ as $\text{loss}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(h(x_i); y_i\big)$. The distribution $\mathcal{D}$ is said to be *realizable* with respect to $\mathcal{H}$ if there exists $h^\star \in \mathcal{H}$ such that $\text{loss}_{\mathcal{D}}(h^\star) = 0$.

Definition 2.3.1 (PAC learning)**.** A hypothesis class $\mathcal{H}$ is PAC learnable with sample complexity $m(\alpha, \beta)$ if there exists an algorithm $\mathcal{A}$ such that for any $\mathcal{H}$-realizable distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, an accuracy and confidence parameters $\alpha, \beta \in (0, 1)$, if $\mathcal{A}$ is given input samples $S = \big((x_i, y_i)\big)_{1:m} \sim \mathcal{D}^m$ such that $m \geq m(\alpha, \beta)$, then it outputs a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ satisfying $\text{loss}_{\mathcal{D}}(h) \leq \alpha$ with probability at least $1 - \beta$. A learner which always returns hypotheses inside the class $\mathcal{H}$ is called a proper learner, otherwise is called an improper learner.

2.4 Differential privacy

Differential privacy (DP), a standard notion of statistical data privacy, was introduced to study data analysis mechanism that do not reveal too much information on any single sample in a dataset. There are several literature on differential privacy, and they include the following books; Dwork et al. (2014), and Li et al. (2016).

Definition 2.4.1 (Differential privacy (Dwork et al., 2014))**.** Data samples $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$ are called neighboring if they differ by exactly one example. A randomized algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}$ is (ϵ, δ)-differentially private if for all neighboring data samples $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$, and for all measurable sets T of outputs,

$$\mathbb{P}\big(\mathcal{A}(S) \in T\big) \leq e^{\epsilon} \cdot \mathbb{P}\big(\mathcal{A}(S') \in T\big) + \delta.$$

The probability is taken over the randomness of $\mathcal{A}$. When $\delta = 0$ we say that $\mathcal{A}$ preserves pure differential privacy, otherwise (when $\delta > 0$) we say that $\mathcal{A}$ preserves approximate differential privacy.

Combining the requirements of PAC and DP learnability yields the definition of private PAC learner.

Definition 2.4.2 (Private PAC learning (Kasiviswanathan et al., 2011))**.** A hypothesis class $\mathcal{H}$ is (ϵ, δ)-differentially private PAC learnable with sample complexity $m(\alpha, \beta)$ if it is PAC learnable with sample complexity $m(\alpha, \beta)$ by an algorithm $\mathcal{A}$ which is (ϵ, δ)-differentially private.

2.5 Online learning

The online learning problem can be viewed as a repeated game between a learner and an adversary. The literature on this subject is vast and includes several books, e.g. Cesa-Bianchi and Lugosi (2006); Shalev-Shwartz and Ben-David (2014), and Hazan (2019).

Let T be a time horizon and $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ be a class of predictors over a domain $\mathcal{X}$. At time t, the adversary chooses a pair $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, and the learner observes the instance x_t, predicts a label $\hat{y}_t \in \mathcal{Y}$, and finally observes the loss $\ell\big(\hat{y}_t; y_t\big)$. This work considers the *full-information setting* where the learner receives the true label information y_t. The goal is to minimize the *regret*, namely the cumulative loss that the learner actually observed compared to the best prediction in hindsight:

$$\sum_{t=1}^{T} \ell\big(\hat{y}_t; y_t\big) - \min_{h^\star \in \mathcal{H}} \sum_{t=1}^{T} \ell\big(h^\star(x_t); y_t\big).$$

A class $\mathcal{H}$ is *online learnable* if for every T, there is an algorithm that achieves sub-linear regret $o(T)$ against any sequence of T instances.

The *Littlestone dimension* is a combinatorial parameter that exactly characterizes online learnability for binary hypothesis classes (Ben-David et al., 2009; Littlestone, 1988). Daniely et al. (2015) further extended this to the multi-class setting. We need the notion of mistake trees to define this complexity measure. A *mistake tree* is a binary tree whose internal nodes are labeled by elements of $\mathcal{X}$. Given a node x, its descending edges are labeled by distinct $k, k' \in \mathcal{Y}$. Then any root-to-leaf path can be expressed as a sequence of instances $\big((x_i, y_i)\big)_{1:d}$, where x_i represents the i-th internal node in the path, and y_i is the label of its descending edge in the path. We say that a tree T is *shattered* by $\mathcal{H}$ if for any root-to-leaf path $\big((x_i, y_i)\big)_{1:d}$ of T, there is $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $i \leq d$. The Littlestone dimension of multi-class hypothesis class $\mathcal{H}$, $\mathrm{Ldim}(\mathcal{H})$, is the maximal depth of any $\mathcal{H}$-shattered mistake tree. Just like binary classification, a set of MC hypotheses $\mathcal{H}$ is online learnable if and only if $\mathrm{Ldim}(\mathcal{H})$ is finite.

The (sequential) *fat-shattering dimension* is the scale-sensitive complexity measure for real-valued function classes (Rakhlin et al., 2015). A mistake tree for real-valued function

class $\mathcal{F}$ is a binary tree whose internal nodes are labeled by $(x, s) \in \mathcal{X} \times \mathcal{Y}$, where s is called a *witness to shattering*. Any root-to-leaf path in a mistake tree can be expressed as a sequence of tuples $\big((x_i, \epsilon_i)\big)_{1:d}$, where x_i is the label of the i-th internal node in the path, and $\epsilon_i = +1$ if the $(i+1)$-th node is the right child of the i-th node, and otherwise $\epsilon_i = -1$ (for the leaf node, ϵ_d can take either value). A tree T is γ-shattered by $\mathcal{F}$ if for any root-to-leaf path $\big((x_i, \epsilon_i)\big)_{1:d}$ of T, there exists $f \in \mathcal{F}$ such that $\epsilon_i \left(f(x_i) - s_i\right) \geq \gamma/2$ for all $i \leq d$. The fat-shattering dimension at scale γ, denoted by $\mathrm{fat}_\gamma(\mathcal{F})$, is the largest d such that $\mathcal{F}$ γ-shatters a mistake tree of depth d. For any function class $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$, $\mathcal{F}$ is online learnable in the supervised setting under the absolute loss if and only if $\mathrm{fat}_\gamma(\mathcal{F})$ is finite for any $\gamma > 0$ (Rakhlin et al., 2015).

The (sequential) *Pollard pseudo-dimension* is a scale-free fat-shattering dimension for real-valued function classes. For every $f \in \mathcal{F}$, we define a binary function $B_f : \mathcal{X} \times \mathcal{Y} \to \{-1, +1\}$ by $B_f(x, s) = \mathrm{sign}\left(f(x) - s\right)$ and let $\mathcal{F}^+ = \{B_f \mid f \in \mathcal{F}\}$. Then we define the Pollard pseudo-dimension by $\mathrm{Pdim}(\mathcal{F}) = \mathrm{Ldim}(\mathcal{F}^+)$. It is easy to check that $\mathrm{fat}_\gamma(\mathcal{F}) \leq \mathrm{Pdim}(\mathcal{F})$ for all γ. That being said, finite Pollard pseudo-dimension is a sufficient condition for online learnability but not a necessary condition (e.g., bounded Lipschitz functions on [0,1] separate the two notions).

2.6 Additional notation

We define a few functions in a recursive manner. The *tower function* twr_t and the *iterated logarithm* $\log^{(m)}$ are defined respectively as

$$\mathrm{twr}_t(x) = \begin{cases} x & \text{if } t = 0, \\ 2^{\mathrm{twr}_{t-1}(x)} & \text{if } t > 0, \end{cases} \qquad \log^{(m)} x = \begin{cases} \log x & \text{if } m = 1, \\ \log^{(m-1)} \log x & \text{if } m > 1. \end{cases}$$

Lastly, we use $\log^* x$ to denote the minimal number of recursions for the iterated logarithm to return the value less than or equal to one:

$$\log^* x = \begin{cases} 0 & \text{if } x \leq 1, \\ 1 + \log^* \log x & \text{if } x > 1. \end{cases}$$

CHAPTER 3

Perturbation Method in Stochastic Multi-armed Bandit Problems

In this chapter we investigate the optimality of perturbation based algorithms in the stochastic multi-armed bandit problems. We propose FTPL algorithms for stochastic bandits and provide a unified regret analysis for both sub-Weibull and bounded perturbations when rewards are sub-Gaussian. Our bounds are instance optimal for sub-Weibull perturbations with parameter 2 that also have a matching lower tail bound, and all bounded support perturbations where there is sufficient probability mass at the extremes of the support.

Since the Uniform and Rademacher distribution are instances of these bounded support perturbations, one of our results is a regret bound for a randomized version of UCB where the algorithm picks a random number in the confidence interval or randomly chooses between lower and upper confidence bounds instead of always picking the upper bound.

This chapter is mainly motivated by Thompson sampling (Thompson, 1933), one of the standard algorithms in stochastic settings. Especially, our analysis relies on the simple but powerful observation that Thompson sampling with Gaussian priors and rewards can also be interpreted as a perturbation algorithm with Gaussian perturbations. We are able to generalize both the upper bound and lower bound of Agrawal and Goyal (2013a) in two respects; (1) from the special Gaussian perturbation to general sub-Weibull or bounded perturbations, and (2) from the special Gaussian rewards to general sub-Gaussian rewards. We also provide a lower bound for the regret of this FTPL algorithm.

For our analysis, we assume, without loss of generality, that arm 1 is optimal, $\mu_1 = \max_{i\in[K]} \mu_i$, and the sub-optimality gap is denoted as $\Delta_i = \mu_1 - \mu_i$. Let $\hat{\mu}_i(t)$ be the average reward received from arm i after round t written formally as $\hat{\mu}_i(t) = \sum_{s=1}^{t} \mathrm{I}\{A_s = i\}X_s/T_i(t)$ where $T_i(t) = \sum_{s=1}^{t} \mathrm{I}\{A_s = i\}$ is the number of times arm i has been pulled

after round t. The regret for stochastic bandits can be decomposed into

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

The reward distributions are generally assumed to be sub-Gaussian with parameter 1 (Lattimore and Szepesvári, 2020).

Definition 3.0.1 (sub-Gaussian)**.** A random variable Z with mean $\mu = \mathrm{E}[Z]$ is sub-Gaussian with parameter $\sigma > 0$ if it satisfies $\mathrm{P}(|Z - \mu| \geq t) \leq \exp(-t^2/(2\sigma^2))$ for all $t \geq 0$.

Lemma 3.0.1 (Hoeffding bound of sub-Gaussian (Hoeffding, 1994))**.** *Suppose Z_i, $i \in [n]$ are i.i.d. random variables with $\mathrm{E}(Z_i) = \mu$ and sub-Gaussian with parameter σ. Then $\mathrm{P}(\bar{Z}_n - \mu \geq t) \vee \mathrm{P}(\bar{Z}_n - \mu \leq -t) \leq \exp(-nt^2/(2\sigma^2))$ for all $t \geq 0$, where $\bar{Z}_n = \sum_{i=1}^{n} Z_i/n$.*

3.1 Upper confidence bound and Thompson sampling

The standard algorithms in stochastic bandit are upper confidence bound (UCB1) (Auer, 2002) and Thompson sampling (Thompson, 1933). The former algorithm is constructed to compare the largest plausible estimate of mean for each arm based on the optimism in the face of uncertainty so that it would be deterministic in contradistinction to the latter one. At time $t+1$, UCB1 chooses an action A_{t+1} by maximizing upper confidence bounds, $UCB_i(t) = \hat{\mu}_i(t) + \sqrt{2 \log T / T_i(t)}$. Regarding the instance-dependent regret of UCB1, there exists some universal constant $C > 0$ such that

$$\mathrm{R}(T) \leq C \sum_{i:\Delta_i > 0} (\Delta_i + \log T / \Delta_i).$$

Thompson sampling is a Bayesian solution based on randomized probability matching approach (Scott, 2010). Given the prior distribution Q_0, at time $t+1$, it computes posterior distribution Q_t based on observed data, samples ν_t from posterior Q_t, and then chooses the arm $A_{t+1} = \arg\max_{i \in [k]} \mu_i(\nu_t)$. In Gaussian Thompson sampling where the Gaussian rewards $\mathcal{N}(\mu_i, 1)$ and the Gaussian prior distribution for each μ_i with mean μ_0 and infinite variance are considered, the policy from Thompson sampling is to choose an index that maximizes $\theta_i(t)$ randomly sampled from Gaussian posterior distribution, $\mathcal{N}(\hat{\mu}_i(t), 1/T_i(t))$ as stated in Alg.1-(1). Also, its regret bound is restated in Theorem 3.1.1.

Algorithm 1 Randomized probability matching algorithms via perturbation

Initialize: $T_i(0) = 0, \hat{\mu}_i(0) = 0$ for all $i \in [K]$
for $t = 1$ **to** T **do**
  for $i = 1$ **to** K **do**
    (1) Gaussian Thompson sampling :$\theta_i(t-1) \sim \mathcal{N}\big(\hat{\mu}_i(t-1), \frac{1}{1 \vee T_i(t-1)}\big)$
    (2) FTPL via unbounded perturbation : $\theta_i(t-1) = \hat{\mu}_i(t-1) + \frac{1}{\sqrt{1 \vee T_i(t-1)}} \cdot Z_{it}$
      where Z_{it}s are randomly sampled from unbounded Z.
    (3) FTPL via bounded perturbation : $\theta_i(t-1) = \hat{\mu}_i(t-1) + \sqrt{\frac{(2+\epsilon)\log T}{1 \vee T_i(t-1)}} \cdot Z_{it}$
      where Z_{it}s are randomly sampled from $Z \in [-1, 1]$.
  end for
  Learner chooses $A_t = \arg\max_{i \in [K]} \theta_i(t-1)$ and receives the reward of $X_t \in [0, 1]$
  Update : $\hat{\mu}_{A_t}(t) = \frac{\hat{\mu}_{A_t}(t-1) \cdot T_{A_t}(t-1) + X_t}{T_{A_t}(t-1)+1}, T_{A_t}(t) = T_{A_t}(t-1) + 1.$
end for

Theorem 3.1.1 (Theorem 3 (Agrawal and Goyal, 2013a))**.** *Assume that reward distribution of each arm i is Gaussian with mean μ_i and unit variance. Thompson sampling policy via Gaussian prior defined in Alg.1-(1) has the following instance-dependent and independent regret bounds, for $C' > 0$,*

$$\mathrm{R}(T) \leq C' \sum_{\Delta_i > 0} \Big(\log(T\Delta_i^2)/\Delta_i + \Delta_i \Big), \quad \mathrm{R}(T) \leq \mathcal{O}(\sqrt{KT \log K}).$$

3.1.1 Viewpoint of follow-the-perturbed-leader

The more generic view of Thompson sampling is via the idea of perturbation. We bring an interpretation of viewing this Gaussian Thompson sampling as follow-the-perturbed-leader (FTPL) algorithm via Gaussian perturbation (Lattimore and Szepesvári, 2020). If Gaussian random variables $\theta_i(t)$ are decomposed into the average mean reward of each arm $\hat{\mu}_i(t)$ and scaled Gaussian perturbation $\eta_{it} \cdot Z_{it}$ where $\eta_{it} = 1/\sqrt{T_i(t)}, Z_{it} \sim N(0, 1)$. In a round $t+1$, the FTPL algorithm chooses the action according to

$$A_{t+1} = \arg\max_{i \in [K]} \hat{\mu}_i(t) + \eta_{it} \cdot Z_{it}.$$

3.2 Follow-the-perturbed-leader

We show that the FTPL algorithm with Gaussian perturbation under Gaussian reward setting can be extended to sub-Gaussian rewards as well as families of sub-Weibull and bounded perturbations. The sub-Weibull family is an interesting family in that it includes

well known families like sub-Gaussian and sub-Exponential as special cases. We propose perturbation based algorithms via sub-Weibull and bounded perturbation in Alg.1-(2), (3), and their regrets are analyzed in Theorem 3.2.1 and 3.2.2.

Definition 3.2.1 (sub-Weibull (Wong et al., 2019))**.** A random variable Z with mean $\mu = \mathrm{E}[Z]$ is sub-Weibull (p) with $\sigma > 0$ if it satisfies $\mathrm{P}(|Z - \mu| \geq t) \leq C_a \exp(-t^p/(2\sigma^p))$ for all $t \geq 0$.

Theorem 3.2.1 (FTPL via sub-Weibull perturbation, Proof in Section A.1)**.** *Assume that reward distribution of each arm i is 1-sub-Gaussian with mean μ_i, and the sub-Weibull (p) perturbation Z with parameter σ and $\mathrm{E}[Z] = 0$ satisfies the following anti-concentration inequality,*

$$\mathrm{P}(|Z| \geq t) \geq \exp(-t^q/2\sigma^q)/C_b, \quad \textit{for } t \geq 0 \tag{3.1}$$

Then the follow-the-perturbed-leader algorithm via Z in Alg.1-(2) has the following instance-dependent and independent regret bounds, for $p \leq q \leq 2$ (if $q = 2$, $\sigma \geq 1$) and $C'' = C(\sigma, p, q) > 0$,

$$\mathrm{R}(T) \leq C'' \sum_{\Delta_i > 0} \Big(\big[\log(T\Delta_i^2) \big]^{2/p} / \Delta_i + \Delta_i \Big), \quad \mathrm{R}(T) \leq \mathcal{O}(\sqrt{KT} (\log K)^{1/p}). \tag{3.2}$$

Note that the parameters p and q can be chosen from any values $p \leq q \leq 2$, and the algorithm can achieve smaller regret bound as p becomes larger. For nice distributions such as Gaussian and double-exponential, the parameters p and q can be matched by 2 and 1, respectively.

Corollary 3.2.1 (FTPL via Gaussian perturbation)**.** *Assume that reward distribution of each arm i is 1-sub-Gaussian with mean μ_i. The follow-the-perturbed-leader algorithm via Gaussian perturbation Z with parameter σ and $\mathrm{E}[Z] = 0$ in Alg.1-(2) has the following instance-dependent and independent regret bounds, for $C'' = C(\sigma, 2, 2) > 0$ and $\sigma \geq 1$,*

$$\mathrm{R}(T) \leq C'' \sum_{\Delta_i > 0} \big(\log(T\Delta_i^2)/\Delta_i + \Delta_i \big), \quad \mathrm{R}(T) \leq \mathcal{O}(\sqrt{KT \log K}). \tag{3.3}$$

Failure of bounded perturbation Any perturbation with bounded support cannot yield an optimal FTPL algorithm. For example, in a two-armed bandit setting with $\mu_1 = 1$ and $\mu_2 = 0$, rewards of each arm i are generated from Gaussian distribution with mean μ_i and unit variance and perturbation is uniform with support $[-1, 1]$. In the case where we have $T_1(10) = 1, T_2(10) = 9$ during first 10 times, and average mean rewards are

$\hat{\mu}_1 = -1$ and $\hat{\mu}_2 = 1/3$, then perturbed rewards are sampled from $\theta_1 \sim \mathcal{U}[-2, 0]$ and $\theta_2 \sim \mathcal{U}[0, 2/3]$. This algorithm will not choose the first arm and accordingly achieve a linear regret. To overcome this limitation of bounded support, we suggest another FTPL algorithm via bounded perturbation by adding an extra logarithmic term in T as stated in Alg.1-(3).

Theorem 3.2.2 (FTPL algorithm via bounded support perturbation, Proof in Section A.3)**.** *Assume that reward distribution of each arm i is 1-sub-Gaussian with mean μ_i, the perturbation distribution Z with $\mathrm{E}[Z] = 0$ lies in $[-1, 1]$ and for any $\epsilon > 0$, there exists $0 < M_{Z,\epsilon} < \infty$ s.t. $\mathrm{P}\big(Z \le \sqrt{2/(2+\epsilon)}\big)/\mathrm{P}\big(Z \ge \sqrt{2/(2+\epsilon)}\big) = M_{Z,\epsilon}$. Then the follow-the-perturbed-leader algorithm via Z in Alg.1-(3) has the following instance-dependent and independent regret bounds, for $C''' > 0$ independent of T, K and Δ_i,*

$$\mathrm{R}(T) \le C''' \sum_{\Delta_i > 0} \Big(\log(T)/\Delta_i + \Delta_i \Big), \quad \mathrm{R}(T) \le \mathcal{O}(\sqrt{KT \log T}). \tag{3.4}$$

Randomized confidence bound algorithm Theorem 3.2.2 implies that the optimism embedded in UCB can be replaced by simple randomization. Instead of comparing upper confidence bounds, our modification is to compare a value randomly chosen from confidence interval or between lower and upper confidence bounds by introducing uniform $\mathcal{U}[-1, 1]$ or Rademacher perturbation $\mathcal{R}\{-1, 1\}$ in UCB1 algorithm with slightly wider confidence interval,

$$A_{t+1} = \arg\max_{i \in [K]} \hat{\mu}_i(t) + \sqrt{(2+\epsilon) \log T / T_i(t)} \cdot Z_{it}.$$

These FTPL algorithms via Uniform and Rademacher perturbations can be regarded as a randomized version of UCB algorithm, which we call the RCB (randomized confidence bound) algorithm, and they also achieve the same regret bound as that of UCB1. The RCB algorithm is meaningful in that it can be arrived at from two different perspectives, either as a randomized variant of UCB or by replacing the Gaussian distribution with Uniform in Gaussian Thompson Sampling.

3.2.1 Regret lower bound

The regret lower bound of the FTPL algorithm in Alg.1-(2) is built on the work of Agrawal and Goyal (2013a). Theorem 3.2.3 states that the regret lower bound depends on the lower bound of the tail probability of perturbation. As special cases, FTPL algorithms via Gaussian ($q = 2$) and Double-exponential ($q = 1$) make the lower and upper regret bounds

matched, $\Theta(\sqrt{KT}(\log K)^{1/q})$.

Theorem 3.2.3 (Regret lower bound, Proof in Section A.4)**.** *If the perturbation Z with $\mathrm{E}[Z] = 0$ has the lower bound of tail probability as $\mathrm{P}(|Z| \geq t) \geq \exp[-t^q/(2\sigma^q)]/C_b$ for $t \geq 0, \sigma > 0$, the follow-the-perturbed-leader algorithm via Z has the lower bound of expected regret, $\Omega(\sqrt{KT}(\log K)^{1/q})$.*

3.3 Numerical experiments

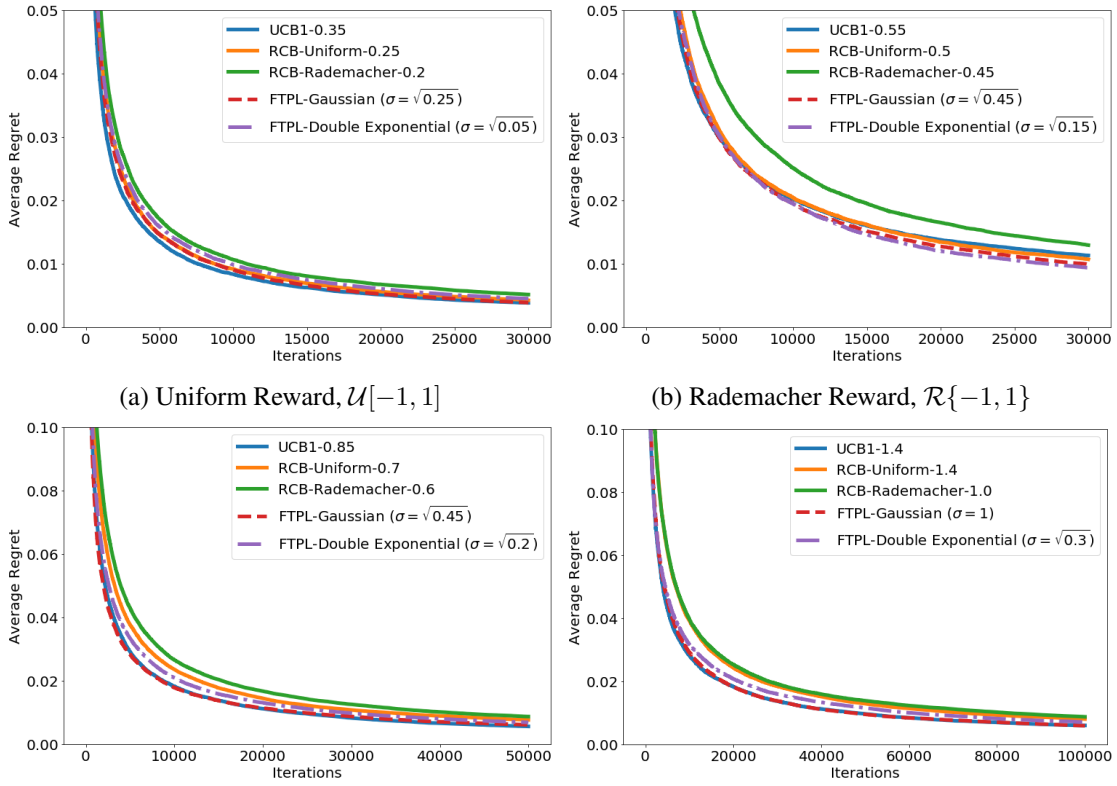
We present some experimental results with perturbation based algorithms (Alg.1-(2),(3)) and compare them to the UCB1 algorithm in the simulated stochastic K-armed bandit. In all experiments, the number of arms (K) is 10, the number of different episodes is 1000, and true mean rewards (μ_i) are generated from $\mathcal{U}[0, 1]$ (Kuleshov and Precup, 2014). We consider the following four examples of 1-sub-Gaussian reward distributions that will be shifted by true mean μ_i; (a) Uniform, $\mathcal{U}[-1, 1]$, (b) Rademacher, $\mathcal{R}\{-1, 1\}$, (c) Gaussian, $\mathcal{N}(0, 1)$, and (d) Gaussian mixture, $W \cdot \mathcal{N}(-1, 1) + (1 - W) \cdot \mathcal{N}(1, 1)$ where $W \sim \mathrm{Bernoulli}(1/2)$. Under each reward setting, we run five different algorithms; UCB1, RCB with Uniform and Rademacher, and FTPL via Gaussian $\mathcal{N}(0, \sigma^2)$ and Double-exponential (σ) after we use grid search to tune confidence levels for confidence based algorithms and the parameter σ for FTPL algorithms. All tuned confidence level and parameter are specified in Figure 3.1. We compare the performance of perturbation based algorithms to UCB1 in terms of average regret $R(t)/t$, which is expected to more rapidly converge to zero if the better algorithm is used.[1]

The average regret plots in Figure 3.1 have the similar patterns that FTPL algorithms via Gaussian and Double-exponential consistently perform the best after parameters tuned, while UCB1 algorithm works as well as them in all rewards except for Rademacher reward. The RCB algorithms with Uniform and Rademacher perturbations are slightly worse than UCB1 in early stages, but perform comparably well to UCB1 after enough iterations. In the Rademacher reward case, which is discrete, RCB with Uniform perturbation slightly outperforms UCB1.

Note that the main contribution of this chapter is to establish theoretical foundations for a large family of perturbation based algorithms (including those used in this section). Our numerical results are not intended to show the superiority of perturbation methods but to demonstrate that they are competitive with Thompson Sampling and UCB. Note that in more complex bandit problems, sampling from the posterior and optimistic optimization

[1] https://github.com/Kimbaekjin/Perturbation-Methods-StochasticMAB

can prove to be computationally challenging. Accordingly, this chapter paves the way for designing efficient perturbation methods in complex settings, such as stochastic linear bandits and stochastic combinatorial bandits, that have both computational advantages and low regret guarantees. Furthermore, perturbation approaches based on the Double-exponential distribution are of special interest from a privacy viewpoint since that distribution figures prominently in the theory of differential privacy (Dwork et al., 2014; Tossou and Dimitrakakis, 2016, 2017).



(a) Uniform Reward, $\mathcal{U}[-1,1]$

(b) Rademacher Reward, $\mathcal{R}\{-1,1\}$

(c) Gaussian Reward, $\mathcal{N}(0,1)$

(d) Gaussian Mixture, $0.5\cdot\mathcal{N}(-1,1)+0.5\cdot\mathcal{N}(1,1)$

Figure 3.1: Average regret for stochastic bandit algorithms in four reward settings

CHAPTER 4

Perturbation Method in Adversarial Multi-armed Bandit Problems

In this chapter we investigate the optimality of perturbation based algorithms in the adversarial multi-armed bandit problems. We study two major families of online learning, follow-the-Regularized-leader (FTRL) and follow-the-perturbed-leader (FTPL), as ways of smoothings and introduce the gradient based prediction algorithm (GBPA) family for solving the adversarial multi-armed bandit problem. Then, we mention an important open problem regarding existence of an optimal FTPL algorithm.

The main contributions of this chapter are theoretical results showing that two natural approaches to solving the open problem are not going to work. (A) One cannot simply find a perturbation that is exactly equivalent to Tsallis entropy. This is surprising since Shannon entropy does have an exact equivalent perturbation, viz. Gumbel. (B) One cannot simply do a better analysis of perturbations used before (Abernethy et al., 2015) and plug the results into their general regret bound to eliminate the extra $O(\sqrt{\log K})$ factor. In proving the first barrier, we use a fundamental result in discrete choice theory. For the second barrier, we rely on tools from extreme value theory. We also make some conjectures on what alternative ideas might work.

4.1 FTRL and FTPL as two types of smoothings and an open problem

Following previous work (Abernethy et al., 2015), we consider a general algorithmic framework, Alg.2. There are two main ingredients of GBPA. The first ingredient is the smoothed potential $\tilde{\Phi}$ whose gradient is used to map the current estimate of the cumulative reward vector to a probability distribution $\mathbf{p}_t$ over arms. The second ingredient is the construction of an unbiased estimate $\hat{\mathbf{g}}_t$ of the rewards vector using the reward of the pulled arm only

by inverse probability weighting. This step reduces the bandit setting to full-information setting so that any algorithm for the full-information setting can be immediately applied to the bandit setting.

Algorithm 2 Gradient-based prediction algorithm in bandit setting

Input: GBPA($\tilde{\Phi}$) is a differentiable convex function such that $\nabla\tilde{\Phi} \in \Delta_{K-1}$ and $\nabla_i\tilde{\Phi} > 0, \forall i \in [K]$
Initialize $\hat{\mathbf{G}}_0 = \mathbf{0}$
for $t = 1$ **to** T **do**
  A reward vector $\mathbf{g}_t \in [0,1]^K$ is chosen by environment
  Learner chooses A_t randomly sampled from the distribution $\mathbf{p}_t = \nabla\tilde{\Phi}(\hat{\mathbf{G}}_{t-1})$
  Learner receives the reward of chosen arm g_{t,A_t}
  Learner estimates reward vector $\hat{\mathbf{g}}_t = \frac{g_{t,A_t}}{p_{t,A_t}}\mathbf{e}_{A_t}$
  Update : $\hat{\mathbf{G}}_t = \hat{\mathbf{G}}_{t-1} + \hat{\mathbf{g}}_t$
end for

If we did not use any smoothing and directly used the baseline potential $\Phi(\mathbf{G}) = \max_{w\in\Delta_{K-1}}\langle w, \mathbf{G}\rangle$, we would be running follow-the-leader (FTL) as our full information algorithm. It is well known that FTL does not have good regret guarantees (Hazan et al., 2016). Therefore, we need to smooth the baseline potential to induce stability in the algorithm. It turns out that two major algorithm families in online learning, namely follow-the-regularized-leader (FTRL) and follow-the-perturbed-leader (FTPL) correspond to two different types of smoothings.

The smoothing used by FTRL is achieved by adding a strongly convex regularizer in the dual representation of the baseline potential. That is, we set $\tilde{\Phi}(\mathbf{G}) = \mathcal{R}^\star(\mathbf{G}) = \max_{w\in\Delta_{K-1}}\langle w, \mathbf{G}\rangle - \eta\mathcal{R}(w)$, where $\mathcal{R}$ is a strongly convex function. The well known exponential weights algorithm (Freund and Schapire, 1997) uses the Shannon entropy regularizer, $\mathcal{R}_S(w) = \sum_{i=1}^K w_i \log(w_i)$. GBPA with the resulting smoothed potential becomes the EXP3 algorithm (Auer et al., 2002) which achieves a near-optimal regret bound $\mathcal{O}(\sqrt{KT\log K})$ just logarithmically worse compared to the lower bound $\Omega(\sqrt{KT})$. This lower bound was matched by implicit normalized forecaster with polynomial function (Poly-INF algorithm) (Audibert and Bubeck, 2009, 2010) and later work (Abernethy et al., 2015) showed that Poly-INF algorithm is equivalent to FTRL algorithm via the Tsallis entropy regularizer, $\mathcal{R}_{T,\alpha}(w) = \frac{1-\sum_{i=1}^K w_i^\alpha}{1-\alpha}$ for $\alpha \in (0,1)$. It converges to Shannon entropy as α approaches to 1, which is why Tsallis Entropy is considered as a generalization of Shannon entropy. Therefore, FTRL via Tsallis entropy generalizes EXP3.

An alternate way of smoothing is *stochastic smoothing* which is what is used by FTPL algorithms. It injects stochastic perturbations to the cumulative rewards of each arm and

then finds the best arm. Given a perturbation distribution $\mathcal{D}$ and $\mathbf{Z} = (Z_1, \cdots, Z_K)$ consisting of i.i.d. draws from $\mathcal{D}$, the resulting stochastically smoothed potential is $\tilde{\Phi}(\mathbf{G}; \mathcal{D}) = \mathrm{E}_{Z_1,\cdots,Z_K \sim \mathcal{D}} \left[\max_{w \in \Delta_{K-1}} \langle w, \mathbf{G} + \eta \mathbf{Z} \rangle\right]$. Its gradient is

$$\mathbf{p}_t = \nabla \tilde{\Phi}(\mathbf{G}_t; \mathcal{D}) = \mathrm{E}_{Z_1,\cdots,Z_K \sim \mathcal{D}}[e_{i^\star}] \in \Delta_{K-1}$$

where $i^\star = \arg\max_i G_{t,i} + \eta Z_i$.

In Section 4.3, we recall the general regret bound proved by Abernethy et al. (2015) for distributions with bounded hazard rate. They showed that a variety of natural perturbation distributions can yield a near-optimal regret bound of $\mathcal{O}(\sqrt{KT \log K})$. However, none of the distributions they tried yielded the minimax optimal rate $\mathcal{O}(\sqrt{KT})$. Since FTRL with Tsallis entropy regularizer can achieve the minimax optimal rate in adversarial bandits, the following is an important unresolved question regarding the power of perturbations.

Open Problem *Is there a perturbation $\mathcal{D}$ such that GBPA with a stochastically smoothed potential using $\mathcal{D}$ achieves the optimal regret bound $\mathcal{O}(\sqrt{KT})$ in adversarial K-armed bandits?*

Given what we currently know, there are two very natural approaches to resolving the open question in the affirmative.

- **Approach 1:** Find a perturbation so that we get the exactly same choice probability function as the one used by FTRL via Tsallis entropy.
- **Approach 2:** Provide a tighter control on expected block maxima of random variables considered as perturbations by Abernethy et al. (2015).

4.2 Barrier against first approach: discrete choice theory

The first approach is motivated by a folklore observation in online learning theory, namely, that the exponential weights algorithm (Freund and Schapire, 1997) can be viewed as FTRL via Shannon entropy regularizer or as FTPL via a Gumbel-distributed perturbation. Thus, we might hope to find a perturbation which is an exact equivalent of the Tsallis entropy regularizer. Since FTRL via Tsallis entropy is optimal for adversarial bandits, finding such a perturbation would immediately settle the open problem.

The relation between regularizers and perturbations has been theoretically studied in discrete choice theory (McFadden, 1981; Hofbauer and Sandholm, 2002). For any perturbation, there is always a regularizer which gives the same choice probability function. The

converse, however, does not hold. The Williams-Daly-Zachary Theorem provides a characterization of choice probability functions that can be derived via additive perturbations.

Theorem 4.2.1 (Williams-Daly-Zachary Theorem (McFadden, 1981)). *Let* $\mathbf{C} : \mathbb{R}^K \to \mathcal{S}_K$ *be the choice probability function and derivative matrix*

$$\mathcal{D}\mathbf{C}(\mathbf{G}) = \left(\frac{\partial \mathbf{C}^\intercal}{\partial G_1}, \frac{\partial \mathbf{C}^\intercal}{\partial G_2}, \cdots, \frac{\partial \mathbf{C}^\intercal}{\partial G_K}\right)^\intercal.$$

The following 4 conditions are necessary and sufficient for the existence of perturbations Z_i *such that this choice probability function* $\mathbf{C}$ *can be written in* $C_i(\mathbf{G}) = \mathrm{P}(\arg\max_{j\in[K]} G_j + \eta Z_j = i)$ *for* $i \in [K]$.

1. $\mathcal{D}\mathbf{C}(\mathbf{G})$ *is symmetric*
2. $\mathcal{D}\mathbf{C}(\mathbf{G})$ *is positive definite*
3. $\mathcal{D}\mathbf{C}(\mathbf{G}) \cdot \mathbf{1} = 0$
4. *All mixed partial derivatives of* $\mathbf{C}$ *are positive,* $(-1)^j \frac{\partial^j C_{i_0}}{\partial G_{i_1} \cdots \partial G_{i_j}} > 0$ *for each* $j = 1, ..., K-1$.

We now show that if the number of arms is greater than three, there does not exist any perturbation exactly equivalent to Tsallis entropy regularization. Therefore, the first approach to solving the open problem is doomed to failure.

Theorem 4.2.2 (Proof in Section B.1). *When* $K \geq 4$*, there is no stochastic perturbation that yields the same choice probability function as the Tsallis entropy regularizer.*

4.3 Barrier against second approach: extreme value theory

The second approach is built on the work of Abernethy et al. (2015) who provided the-state-of-the-art perturbation based algorithm for adversarial multi-armed bandits. The framework proposed in this work covered all distributions with bounded hazard rate and showed that the regret of GBPA via perturbation $Z \sim \mathcal{D}$ with a bounded hazard is upper bounded by trade-off between the bound of hazard rate and expected block maxima as stated below.

Theorem 4.3.1 (Theorem 4.2 (Abernethy et al., 2015)). *Assume the support of* $\mathcal{D}$ *is unbounded in positive direction and hazard rate* $h_{\mathcal{D}}(x) = \frac{f(x)}{1-F(x)}$ *is bounded, then the expected regret of GBPA(*$\tilde{\Phi}$*) in adversarial bandit is bounded by* $\eta \cdot \mathrm{E}[M_K] + \frac{K \sup h_{\mathcal{D}}}{\eta} T$,

where $\sup h_{\mathcal{D}} = \sup_{x:f(x)>0} h_{\mathcal{D}}(x)$. *The optimal choice of* η *leads to the regret bound* $2\sqrt{KT \cdot \sup h_{\mathcal{D}} \cdot \mathrm{E}[M_K]}$ *where* $M_K = \max_{i\in[K]} Z_i$.

Abernethy et al. (2015) considered several perturbations such as Gumbel, Gamma, Weibull, Fréchet and Pareto. The best tuning of distribution parameters (to minimize *upper bounds* on the product $\sup h_{\mathcal{D}} \cdot \mathrm{E}[M_K]$) always leads to the bound $\mathcal{O}(\sqrt{KT \log K})$, which is tantalizingly close to the lower bound but does not match it. It is possible that some of their upper bounds on expected block maxima $\mathrm{E}[M_K]$ are loose and that we can get closer, or perhaps even match, the lower bound by simply doing a better job of bounding expected block maxima (we will not worry about supremum of the hazard since their bounds can easily be shown to be tight, up to constants, using elementary calculations in Appendix B.2). We show that this approach will also not work by *characterizing* the asymptotic (as $K \to \infty$) behavior of block maxima of perturbations using extreme value theory. The statistical behavior of block maxima, $M_K = \max_{i\in[K]} Z_i$, where Z_i's is a sequence of i.i.d. random variables with distribution function F can be described by one of three extreme value distributions: Gumbel, Fréchet and Weibull (Coles et al., 2001; Resnick, 2013). Then, the normalizing sequences $\{a_K > 0\}$ and $\{b_K\}$ are explicitly characterized (Leadbetter et al., 2012). Under the mild condition, $\mathrm{E}\big((M_K - b_K)/a_K\big) \to \mathrm{E}_{Z\sim G}[Z] = C$ as $K \to \infty$ where G is extreme value distribution and C is constant, and the expected block maxima behave asymptotically as $\mathrm{E}[M_K] = \Theta(C \cdot a_K + b_K)$. See Theorem B.2.1-B.2.3 in Section B.2.1 for more details.

Table 4.1: Asymptotic expected block maximum of five different distributions based on extreme value theory. Gumbel-type and Fréchet-type are denoted by Λ and Φ_α respectively. The Gamma function and the Euler-Mascheroni constant are denoted by $\Gamma(\cdot)$ and γ respectively.

Distribution	**Type**	**sup h**	$\mathbf{E[M_K]}$
Gumbel($\mu = 0, \beta = 1$)	Λ	1	$\log K + \gamma + o(1)$
Gamma($\alpha, 1$)	Λ	1	$\log K + \gamma + o(\log K)$
Weibull($\alpha \le 1$)	Λ	α	$(\log K)^{1/\alpha} + o((\log K)^{1/\alpha})$
Fréchet ($\alpha > 1$)	Φ_α	$\in (\frac{\alpha}{e-1}, 2\alpha)$	$\Gamma(1 - 1/\alpha) \cdot K^{1/\alpha}$
Pareto($x_m = 1, \alpha$)	Φ_α	α	$\Gamma(1 - 1/\alpha) \cdot (K^{1/\alpha} - 1)$

The asymptotically tight growth rates (with explicit constants for the leading term!) of expected block maximum of some distributions are given in Table 4.1. They match the upper bounds of the expected block maximum in Table 1 of Abernethy et al. (2015). That is, their upper bounds are asymptotically tight. Gumbel, Gamma and Weibull distribution are Gumbel-type (Λ) and their expected block maximum behave as $\mathcal{O}(\log K)$ asymptotically. It implies that Gumbel type perturbation can never achieve optimal regret bound

despite bounded hazard rate. Fréchet and Pareto distributions are Fréchet-type (Φ_α) and their expected block maximum grows as $K^{1/\alpha}$. Heuristically, if α is set optimally to $\log K$, the expected block maxima is independent of K while the supremum of hazard is upper bounded by $\mathcal{O}(\log K)$.

Conjecture *If there exists a perturbation that achieves minimax optimal regret in adversarial multi-armed bandits, it must be of Fréchet-type.*

Fréchet-type perturbations can still possibly yield the optimal regret bound in perturbation based algorithm if the expected block maximum is asymptotically bounded by a constant and the divergence term in regret analysis of GBPA algorithm can be shown to enjoy a tighter bound than what follows from the assumption of a bounded hazard rate.

The perturbation equivalent to Tsallis entropy (in two armed setting) is of Fréchet-type Further evidence to support the conjecture can be found in the connection between FTRL and FTPL algorithms that regularizer $\mathcal{R}$ and perturbation $Z \sim F_{\mathcal{D}}$ are bijective in two-armed bandit in terms of a mapping between $F_{\mathcal{D}^\star}$ and $\mathcal{R}$, $\mathcal{R}(w) - \mathcal{R}(0) = -\int_0^w F_{\mathcal{D}^\star}^{-1}(1-z)dz$, where Z_1, Z_2 are i.i.d random variables with distribution function, $F_{\mathcal{D}}$, and then $Z_1 - Z_2 \sim F_{\mathcal{D}^\star}$. The difference of two i.i.d. Fréchet-type distributed random variables is conjectured to be Fréchet-type. Thus, Tsallis entropy in two-armed setting leads to Fréchet-type perturbation, which supports our conjecture about optimal perturbations in adversarial multi-armed bandits. See Section B.3 for more details.

CHAPTER 5

Randomized Exploration in Stationary Stochastic Linear Bandits

In this chapter we examine randomized algorithms in stationary stochastic linear bandits and explicate the role of two perturbation approaches in overcoming conservatism that UCB-type algorithms chronically suffer from in practice. In one approach, we replace optimism with a simple randomization when using confidence sets. In the other, we add random perturbations to the current estimate before maximizing the expected reward. These two approaches result in randomized LinUCB and Gaussian linear Thompson sampling for stationary linear bandits. We highlight the statistical optimality versus oracle efficiency trade-off between them.

A learner chooses an action X_t from a given action set $\mathcal{X}_t \subset \mathbb{R}^d$ in every round t, and he subsequently observes a reward $Y_t = \langle X_t, \theta^\star \rangle + \eta_t$ where $\theta^\star \in \mathbb{R}^d$ is an unknown parameter and η_t is a conditionally 1-subGaussian random variable. To learn about unknown parameter $\theta^\star$ from history $\mathcal{H}_{t-1} = \{(X_l, Y_l)_{1 \le l \le t-1}\}$, algorithms rely on l^2-regularized least-squares estimate of $\theta^\star$, $\hat{\theta}_t^{ls}$, and confidence ellipsoid centered from $\hat{\theta}_t^{ls}$. We define $\hat{\theta}_t^{ls} = V_{t,\lambda}^{-1} \sum_{l=1}^{t-1} X_l Y_l$, where $V_{t,\lambda} = \lambda I_d + \sum_{l=1}^{t-1} X_l X_l^T$ and λ is a positive regularization parameter.

5.1 Randomized exploration

The standard solutions in stationary stochastic linear bandit are optimism based algorithm (LinUCB, Abbasi-Yadkori et al. (2011)) and linear Thompson sampling (LinTS, Agrawal and Goyal (2013b)). While the former obtains the theoretically optimal regret bound $\tilde{\mathcal{O}}(d\sqrt{T})$ matched to lower bound $\Omega(d\sqrt{T})$, the latter empirically performs better in spite of its regret bound $\sqrt{d}$ worse than LinUCB (Chapelle and Li, 2011). In finite-arm setting, the regret bound of Gaussian linear Thompson sampling (Gaussian-LinTS) is improved

by $\sqrt{(\log K)/d}$ as a special case of follow-the-perturbed-leader-GLM (FPL-GLM, Kveton et al. (2020)). Also, a series of randomized algorithms for linear bandit were proposed in recent works: linear perturbed history exploration (LinPHE, Kveton et al. (2019)) and randomized linear UCB (RandLinUCB, Vaswani et al. (2020)). They are categorized in terms of regret bounds, randomness, and oracle access in Table 5.1, where we denote $K = \max_{t\in[T]} |\mathcal{X}_t|$ in finite-arm setting.

There are two families of randomized algorithms according to the way perturbations are used. The first algorithm family is designed to choose an action by maximizing the expected rewards after adding the random perturbation to estimates. Gaussian-LinTS, LinPHE, and FPL-GLM are in this family. But they are limited in that their regret bounds, $\tilde{\mathcal{O}}(d\sqrt{T\log K})$, depend on the number of arms, and lead to $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ regret bounds when the action set is infinite. The other family including RandLinUCB is constructed by replacing the optimism with simple randomization when choosing a confidence level to handle the chronic issue that UCB-type algorithms are too conservative. This randomized version of LinUCB matches optimal regret bounds of LinUCB as well as the empirical performance of LinTS.

Table 5.1: Comparison of algorithms in stationary stochastic linear bandits : regret bound, randomness, and oracle access

Algorithm	Regret bound	Randomness	Oracle access
LinUCB (Abbasi-Yadkori et al., 2011)	$\tilde{\mathcal{O}}(d\sqrt{T})$	No	No
LinTS (Agrawal and Goyal, 2013b)	$\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$	Yes	Yes
Gaussian LinTS (Kveton et al., 2020)	$\tilde{\mathcal{O}}(d\sqrt{T\log K})$	Yes	Yes
LinPHE (Kveton et al., 2019)	$\tilde{\mathcal{O}}(d\sqrt{T\log K})$	Yes	Yes
RandLinUCB (Vaswani et al., 2020)	$\tilde{\mathcal{O}}(d\sqrt{T})$	Yes	No

Oracle point of view We assume that the learner has access to an algorithm that returns a near-optimal solution to the offline problem, called an *offline optimization oracle*. It returns the optimal action that maximizes the expected reward from a given action space $\mathcal{X} \subset \mathbb{R}^d$ when a parameter $\theta \in \mathbb{R}^d$ is given as input.

Definition 5.1.1 (Offline optimization oracle)**.** There exists an algorithm, $\mathcal{A}.\mathcal{M}.\mathcal{O}.$, which when given a pair of action space $\mathcal{X} \subset \mathbb{R}^d$, and a parameter $\theta \in \mathbb{R}^d$, computes

$$\mathcal{A}.\mathcal{M}.\mathcal{O}.(\mathcal{X}, \theta) = \arg\max_{x\in\mathcal{X}} \langle x, \theta\rangle.$$

Both the non-randomized LinUCB and RandLinUCB are required to compute spectral norms of all actions $\|x\|_{V_{t,\lambda}^{-1}}$ in every round so that they cannot be efficiently implemented with an infinite set of arms. The main advantage of the algorithms in the first family such as Gaussian-LinTS, LinPHE, and FPL-GLM is that they rely on an offline optimization oracle in every round t so that the optimal action can be efficiently obtained within polynomial times from large or even infinite action set.

5.1.1 Improved regret bound of Gaussian LinTS

In FTL-GLM, it is required to generate perturbations and save d-dimensional feature vectors $\{X_l\}_{l=1}^{t-1}$ in order to obtain perturbed estimate $\tilde{\theta}_t$ in every round t, which causes computation burden and memory issue for storage. However, once perturbations are Gaussian in the linear model, adding univariate Gaussian perturbations to historical rewards is the same as perturbing the estimate $\hat{\theta}_t$ by a multivariate Gaussian perturbation because of its linear invariance property, and the resulting algorithm is approximately equivalent to Gaussian linear Thompson sampling (Agrawal and Goyal, 2013b) as follows.

$$\begin{aligned}\tilde{\theta}_t &= \hat{\theta}_t + V_{t,\lambda}^{-1} \sum_{l=1}^{t-1} X_l Z_l^{(t)}, \; Z_l^{(t)} \sim \mathcal{N}(0, a^2) \\ &\approx \hat{\theta}_t + V_{t,\lambda}^{-1/2} Z^{(t)}, \; Z^{(t)} \sim \mathcal{N}(0, a^2 I_d) \;\; : \text{Gaussian-LinTS}.\end{aligned}$$

It naturally implies the regret bound of Gaussian-LinTS is improved by $\sqrt{(\log K)/d}$ with finite action sets (Kveton et al., 2020).

5.1.2 Equivalence between Gaussian LinTS and RandLinUCB

Another perspective of Gaussian-LinTS algorithm is that it is equivalent to RandLinUCB with *decoupled* perturbations across arms due to linearly invariant property of Gaussian random variables:

$$\begin{aligned}\langle x, \tilde{\theta}_t \rangle &= \langle x, \hat{\theta}_t \rangle + x^T V_{t,\lambda}^{-1/2} Z^{(t)}, \; Z^{(t)} \sim \mathcal{N}(0, a^2 I_d) \\ &= \langle x, \hat{\theta}_t \rangle + Z_{t,x} \|x\|_{V_{t,\lambda}^{-1}}, \; Z_{t,x} \sim N(0, a^2) \;\; : \textit{Decoupled} \text{ RandLinUCB}.\end{aligned}$$

If perturbations are coupled, we compute the perturbed expected rewards of all actions using randomly chosen confidence level $Z_t \sim N(0, a^2)$ instead of $Z_{t,x}$. In the decoupled RandLinUCB where each arm has its own random confidence level, more variations are generated so that its regret bound have extra logarithmic gap that depends on the number

of decoupled actions.

In other words, the standard (*coupled*) RandLinUCB enjoys minimax-optimal regret bound due to coupled perturbations. However, there is a cost to its theoretical optimality: it cannot just rely on an offline optimization oracle and thus loses computational efficiency. We thus have a trade-off between efficiency and optimality described in two design principles of perturbation based algorithms.

CHAPTER 6

Randomized Exploration in Non-Stationary Stochastic Linear Bandits

This chapter considers non-stationary stochastic linear bandit problems. The origin of linear bandit problems were motivated by applications such as online ad placement with features extracted from the ads and website users. In practice, however, users' preferences often evolve with time, which leads to interest in the non-stationary variant of linear bandits. To accommodate time-variation of environments, the reward $Y_t = \langle X_t, \theta_t^\star \rangle + \eta_t$ is observed to the learner where $\theta_t^\star \in \mathbb{R}^d$ is an unknown time-varying parameter and η_t is a conditionally 1-subGaussian random variable. The non-stationary assumption allows unknown parameter $\theta_t^\star$ to be time-variant within total variation budget $B_T = \sum_{t=1}^{T-1} \|\theta_t^\star - \theta_{t+1}^\star\|_2$.

In this chapter, we present two randomized algorithms with exponential discounting weights for non-stationary environment, discounted randomized LinUCB (D-RandLinUCB) and discounted linear Thompson sampling (D-LinTS) to gracefully adjust to the time-variation in the true parameter. We explain the trade-off between statistical guarantee and oracle efficiency in that the former asymptotically achieves dynamic regret $\tilde{O}(d^{7/8} B_T^{1/4} T^{3/4})$, which is the same as that of three mainstream algorithms such as SW-LinUCB (Cheung et al., 2019), D-LinUCB (Russac et al., 2019), and Restart-LinUCB (Zhao et al., 2020), but the latter enjoys computational efficiency due to sole reliance on an offline optimization oracle for large or infinite action set. However it incurs an extra $(\log K)^{3/8}$ gap in its dynamic regret bound, where K is the number of actions.

In addition, we run multiple simulation studies based on Criteo live traffic data (Diemert et al., 2017) to evaluate the empirical performances of D-RandLinUCB and D-LinTS. We observe that when high dimension and a large set of actions are considered, the two show outstanding performance in tackling conservatism issue that the non-randomized D-LinUCB struggles with.

6.1 Optimism based algorithm

In a stationary stochastic environment where the reward has a linear structure, linear upper confidence bound algorithm (LinUCB) follows a principle of optimism in the face of uncertainty (OFU). Under this OFU principle, three recent works of Cheung et al. (2019); Russac et al. (2019); Zhao et al. (2020) proposed sliding window linear UCB (SW-LinUCB), discounted linear UCB (D-LinUCB), and restarting linear UCB (Restart-LinUCB) which are non-stationary variants of LinUCB to adapt to time-variation of $\theta_t^\star$. First two algorithms rely on weighted least-squares estimators with equal weights only given to recent w observations where w is length of a sliding-window, and exponentially discounting weights, respectively. The last algorithm proceeds in epochs, and is periodically restarted to be resilient to the drift of underlying parameter θ_t.

Three non-randomized algorithms based on three different approaches are known to achieve the dynamic regret bounds $\tilde{O}(d^{7/8}B_T^{1/4}T^{3/4})$ using Bandit-over-Bandit (BOB) mechanism (Cheung et al., 2019) without the prior information on B_T, but share inefficiency of implementation with LinUCB (Abbasi-Yadkori et al., 2011) in that the computation of spectral norms of all actions are required. Furthermore, they are built upon the construction of a high-probability confidence ellipsoid for the unknown parameter, and thus they are deterministic and their confidence ellipsoids become too wide when high dimensional features are available. In this section, randomization exploration algorithms, discounted randomized LinUCB (D-RandLinUCB) and discounted linear Thompson sampling (D-LinTS), are proposed to handle computational inefficiency and conservatism that both optimism-based algorithms suffer from. The dynamic regret bound, randomness, and oracle access of algorithms are reported in Table 6.1.

Table 6.1: Comparison of algorithms in non-stationary stochastic linear bandits : regret bound, randomness, and oracle access

Algorithm	Regret bound	Randomness	Oracle access
D-LinUCB (Russac et al., 2019)	$\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$	No	No
SW-LinUCB (Cheung et al., 2019)	$\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$	No	No
Restart-LinUCB (Zhao et al., 2020)	$\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$	No	No
D-RandLinUCB [Algorithm 3]	$\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$	Yes	No
D-LinTS [Algorithm 4]	$\mathcal{O}(d^{\frac{7}{8}}(\log K)^{\frac{3}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$	Yes	Yes

6.2 Weighted least-squares estimator

First, we study the weighted least-squares estimator with discounting factor $0 < \gamma < 1$. In the round t, the weighted least-squares estimator is obtained in a closed form,

$$\hat{\theta}_t^{wls} = \arg\max_{\theta} \sum_{l=1}^{t-1} \gamma^{t-l}(Y_l - \langle X_l, \theta \rangle)^2 + \frac{\lambda}{2}\|\theta\|_2^2 = W_{t,\lambda}^{-1} \sum_{s=1}^{t-1} \gamma^{-l} X_l Y_l$$

where $W_{t,\lambda} = \sum_{l=1}^{t-1} \gamma^{-l} X_l X_l^T + \lambda \gamma^{-(t-1)} I_d$.

Additionally, we define $\tilde{W}_{t,\lambda} = \sum_{l=1}^{t-1} \gamma^{-2l} X_l X_l^T + \lambda \gamma^{-2(t-1)} I_d$. This form is closely connected with the covariance matrix of $\hat{\theta}_t^{wls}$. For simplicity, we denote $V_t = W_{t,\lambda} \tilde{W}_{t,\lambda}^{-1} W_{t,\lambda}$.

Lemma 6.2.1 (Weighted least-sqaures confidence ellipsoid, Theorem 1 (Russac et al., 2019))**.** *Assume the stationary setting where $\theta_t^\star = \theta^\star$. For any $\delta > 0$,*

$$P\big(\forall t \geq 1, \|\hat{\theta}_t^{wls} - \theta^\star\|_{W_{t,\lambda}\tilde{W}_{t,\lambda}^{-1}W_{t,\lambda}} \leq \beta_t\big) \geq 1 - \delta$$

where $\beta_t = \sqrt{\lambda} + \sqrt{2\log(1/\delta) + d\log(1 + \frac{(1-\gamma^{2t})}{\lambda d(1-\gamma^2)})}$.

While Lemma 6.2.1 states that the confidence ellipsoid

$$\mathcal{C}_t = \{\theta \in \mathbb{R}^d : \|\theta - \theta_t^{wls}\|_{W_{t,\lambda}\tilde{W}_{t,\lambda}^{-1}W_{t,\lambda}} \leq \beta_t\}$$

contains true parameter $\theta_t^\star$ with high probability in stationary setting, the true parameter $\theta_t^\star$ is not necessarily inside the confidence ellipsoid $\mathcal{C}_t$ in the non-stationary setting because of variation in the parameters. We alternatively define a *surrogate parameter* $\bar{\theta}_t = W_{t,\lambda}^{-1}(\sum_{l=1}^{t-1} \gamma^{-l} X_l X_l^T \theta_l^\star + \lambda \gamma^{-(t-1)} \theta_t^\star)$, which belongs to $\mathcal{C}_t$ with probability at least $1 - \delta$, which is formally stated in Lemma 6.4.1.

6.3 Randomized exploration

In this section, we propose two randomized algorithms for non-stationary stochastic linear bandits, discounted randomized LinUCB (D-RandLinUCB) and discounted linear Thompson sampling (D-LinTS). To gracefully adapt to environmental variation, the weighted method with exponentially discounting factor is directly applied to both RandLinUCB and Gaussian-LinTS, respectively. The random perturbations are injected to D-RandLinUCB and D-LinTS in different fashions: either by replacing optimism with simple randomization

in deciding the confidence level or perturbing estimates before maximizing the expected rewards.

6.3.1 Discounted randomized linear UCB

Following the optimism in face of uncertainty principle, D-LinUCB (Russac et al., 2019) chooses an action by maximizing the upper confidence bound of expected reward based on $\hat{\theta}_t^{wls}$ and confidence level a. Motivated by the recent work of Vaswani et al. (2020), our first randomized algorithm in non-stationary linear bandit setting is constructed by replacing confidence level a with a random variable $Z_t \sim \mathcal{D}$ and this non-stationary variant of RandLinUCB algorithm is called discounted randomized LinUCB (D-RandLinUCB, Algorithm 3),

$$\text{D-LinUCB} : X_t = \arg\max_{x \in \mathcal{X}_t} \langle x, \hat{\theta}_t^{wls} \rangle + a \|x\|_{V_t^{-1}}$$
$$\text{D-RandLinUCB} : X_t = \arg\max_{x \in \mathcal{X}_t} \langle x, \hat{\theta}_t^{wls} \rangle + Z_t \|x\|_{V_t^{-1}}.$$

Algorithm 3 Discounted randomized linear UCB (D-RandLinUCB)

Input: $\lambda > 0$, $0 < \delta < 1$, $0 < \gamma < 1$, and $a > 0$
Initialize $W = \lambda I_d$, $\tilde{W} = \lambda I_d$, $\bar{b} = 0$, and $\hat{\theta} = 0$.
for $t = 1$ **to** T **do**
  Randomly sample Z_t from a distribution $\mathcal{D}(\delta, a)$
  Obtain $UCB(x) = x^T \hat{\theta} + Z_t \sqrt{x^T W^{-1} \tilde{W} W^{-1} x}$
  $X_t = \arg\max_{x \in \mathcal{X}_t} UCB(x)$
  Play action X_t and receive reward Y_t
  Update $W = \gamma W + X_t X_t^T + (1 - \gamma)\lambda I_d$,
  $\tilde{W} = \gamma^2 \tilde{W} + X_t X_t^T + (1 - \gamma^2)\lambda I_d$,
  $\bar{b} = \gamma \bar{b} + X_t Y_t$, $\hat{\theta} = W^{-1} \bar{b}$.
end for

6.3.2 Discounted linear Thompson sampling

The idea of perturbing estimates via random perturbation in LinTS algorithm can be directly applied to non-stationary setting by replacing $\hat{\theta}_t^{ls}$ and Gram matrix $V_{t,\lambda}$ with the weighted least-squares estimator $\hat{\theta}_t^{wls}$ and its corresponding matrix $V_t = W_{t,\lambda} \tilde{W}_{t,\lambda}^{-1} W_{t,\lambda}$. We call it discounted linear Thompson sampling (D-LinTS, Algorithm 4). The motivation of D-LinTS arises from its equivalence to D-RandLinUCB with *decoupled* perturbations $Z_{x,t}$ for all $x \in \mathcal{X}_t$ in round t as

$$\tilde{f}_t(x) = \langle x, \tilde{\theta}_t^{wls} \rangle = \langle x, \hat{\theta}_t^{wls} \rangle + x^T W_{t,\lambda}^{-1} \tilde{W}_{t,\lambda}^{1/2} Z^{(t)}$$
$$= \langle x, \hat{\theta}_t^{wls} \rangle + Z_{x,t} \|x\|_{V_t^{-1}}$$

where $Z^{(t)} \sim \mathcal{N}(0_d, a^2 I_d), Z_{x,t} \sim \mathcal{N}(0, a^2)$. Perturbations above are decoupled in that random perturbation are not shared across every arm, and thus they obtain more variation and accordingly $(\log K)^{3/8}$ larger regret bound than that of D-RandLinUCB algorithm that is associated with *coupled* perturbations Z_t. By paying a logarithmic regret gap in terms of K at a cost, the innate perturbation of D-LinTS allows itself to have an offline optimization oracle access in contrast to D-LinUCB and D-RandLinUCB. Therefore, D-LinTS algorithm can be efficient in computation even with an infinite action set.

Algorithm 4 Discounted linear Thompson sampling (D-LinTS)

Input: $\lambda > 0$, $0 < \gamma < 1$, and $a > 0$
Initialize $W = \lambda I_d$, $\tilde{W} = \lambda I_d$, $\bar{b} = 0$ and $\hat{\theta} = 0$.
for $t = 1$ **to** T **do**
  Obtain $\tilde{\theta} = \hat{\theta} + W^{-1} \tilde{W}^{1/2} Z$, $Z \sim \mathcal{N}(0, a^2 I_d)$
  Oracle : $X_t = \arg\max_{x \in \mathcal{X}_t} \langle x, \tilde{\theta} \rangle$
  Play action X_t and receive reward Y_t
  Update $W = \gamma W + X_t X_t^T + (1 - \gamma) \lambda I_d$,
  $\tilde{W} = \gamma^2 \tilde{W} + X_t X_t^T + (1 - \gamma^2) \lambda I_d$,
  $\bar{b} = \gamma \bar{b} + X_t Y_t$, $\hat{\theta} = W^{-1} \bar{b}$.
end for

6.4 Analysis

We construct a general regret bound for linear bandit algorithm on the top of prior work of Kveton et al. (2019). The difference from their work is that an action set $\mathcal{X}_t$ varies from time t and can have infinite arms. Also, non-stationary environment is considered where true parameter $\theta_t^\star$ changes within total variation B_T. The expected dynamic regret is

decomposed into surrogate regret and bias arising from total variation.

$$\begin{aligned} E[R(T)] &= \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \theta_t^\star \rangle] \\ &= \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \theta_t^\star - \bar{\theta}_t \rangle] \\ &\leq \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + 2 \sum_{t=1}^{T} \|\theta_t^\star - \bar{\theta}_t\|_2 \end{aligned}$$

6.4.1 Surrogate instantaneous regret

To bound the surrogate instantaneous regret $E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle]$, we newly define three events E^{wls}, E_t^{conc}, and E_t^{anti}:

$$\begin{aligned} E^{wls} &= \{\forall (x,t) \in \bar{\mathcal{X}}_T; |\langle x, \hat{\theta}_t^{wls} - \bar{\theta}_t \rangle| \leq c_1 \|x\|_{V_t^{-1}}\}, \\ E_t^{conc} &= \{\forall x \in \mathcal{X}_t; |\tilde{f}_t(x) - \langle x, \hat{\theta}_t^{wls} \rangle| \leq c_2 \|x\|_{V_t^{-1}}\}, \\ E_t^{anti} &= \{\tilde{f}_t(x_t^\star) - \langle x_t^\star, \hat{\theta}_t^{wls} \rangle > c_1 \|x_t^\star\|_{V_t^{-1}}\}, \end{aligned}$$

where $\bar{\mathcal{X}}_T = \{(x,t) : x \in \mathcal{X}_t, t \in [T]\}$. The choice of $\tilde{f}_t(x)$ is made by algorithmic design, which decides choices on both c_1 and c_2 simultaneously. In round t, we consider the general algorithm which maximizes perturbed expected reward $\tilde{f}_t(x)$ over action space $\mathcal{X}_t$. The following theorem is a extension of Theorem 1 (Kveton et al., 2019) to the time-evolving environment.

Theorem 6.4.1. *Assume we have $\lambda > 0$ and $c_1, c_2 \geq 1$ satisfying $P(E^{wls}) \geq 1 - p_1$, $P(E_t^{conc}) \geq 1 - p_2$, and $P(E_t^{anti}) \geq p_3$, and $c_3 = 2d \log(\frac{1}{\gamma}) + 2\frac{d}{T} \log(1 + \frac{1}{d\lambda(1-\gamma)})$. Let A be an algorithm that chooses arm $X_t = \arg\max_{\mathcal{X}_t} \tilde{f}_t(x)$ at time t. Then the expected surrogate instantaneous regret of A, $E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle]$ is bounded by*

$$p_2 + (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big) E_t\big[\|X_t\|_{V_t^{-1}}\big].$$

Proof. Firstly, we newly define $\Delta_x = \langle x_t^\star - x, \bar{\theta}_t \rangle$ in round t. Given history $\mathcal{H}_{t-1}$, we assume that event E^{wls} holds and let $\bar{S}_t = \{x \in \mathcal{X}_t : (c_1 + c_2)\|x\|_{V_t^{-1}} \geq \Delta_x \text{ and } \Delta_x \geq 0\}$ be the set of arms that are under-sampled and worse than $x_t^\star$ given $\bar{\theta}_t$ in round t. Among them, let $U_t = \arg\min_{x \in \bar{S}_t} \|x\|_{V_t^{-1}}$ be the least uncertain under-sampled arm in round t. By definition of the optimal arm, $x_t^\star \in \bar{S}_t$. The set of sufficiently sampled arms is defined as $S_t = \{x \in \mathcal{X}_t : (c_1 + c_2)\|x\|_{V_t^{-1}} \leq \Delta_x \text{ and } \Delta_x \geq 0\}$ and let $c = c_1 + c_2$. Note that any

actions $x \in \mathcal{X}_t$ with $\Delta_x < 0$ can be neglected since the regret induced by these actions are always negative so that it is upper bounded by zero. Given history $\mathcal{H}_{t-1}$, U_t is deterministic term while X_t is random because of innate randomness in $\tilde{f}_t$. Thus surrogate instantaneous regret can be bounded as,

$$\begin{aligned}
\Delta_{X_t} &= \Delta_{U_t} + \langle U_t, \bar{\theta}_t \rangle - \langle X_t, \bar{\theta}_t \rangle \\
&\leq \Delta_{U_t} + \tilde{f}_t(U_t) - \tilde{f}_t(X_t) + c\|X_t\|_{V_t^{-1}} + c\|U_t\|_{V_t^{-1}} \\
&\leq c\|X_t\|_{V_t^{-1}} + 2c\|U_t\|_{V_t^{-1}}.
\end{aligned}$$

Thus, the expected surrogate instantaneous regret can be bounded as,

$$\begin{aligned}
E_t[\Delta_{X_t}] &= E_t[\Delta_{X_t} I\{E_t^{conc}\}] + E_t[\Delta_{X_t} I\{\bar{E}_t^{conc}\}] \\
&\leq cE_t[\|X_t\|_{V_t^{-1}}] + 2c\|U_t\|_{V_t^{-1}} + P_t(\bar{E}_t^{conc}) \\
&\leq cE_t[\|X_t\|_{V_t^{-1}}] + 2c\|U_t\|_{V_t^{-1}} + p_2 \\
&\leq cE_t[\|X_t\|_{V_t^{-1}}] + 2c\frac{E_t[\|X_t\|_{V_t^{-1}}]}{P_t(X_t \in \bar{S}_t)} + p_2 \\
&= c(1 + \frac{2}{P_t(X_t \in \bar{S}_t)})E_t[\|X_t\|_{V_t^{-1}}] + p_2 \\
&\leq c(1 + \frac{2}{p_3 - p_2})E_t[\|X_t\|_{V_t^{-1}}] + p_2
\end{aligned}$$

The third inequality holds because of definition of U_t that is the least uncertain in $\bar{S}_t$ and deterministic as follows,

$$\begin{aligned}
E_t[\|X_t\|_{V_t^{-1}}] &\geq E_t[\|X_t\|_{V_t^{-1}} | X_t \in \bar{S}_t] \cdot P_t(X_t \in \bar{S}_t) \\
&\geq \|U_t\|_{V_t^{-1}} \cdot P_t(X_t \in \bar{S}_t).
\end{aligned}$$

The second last inequality holds since on event E_t^{ls},

$$\begin{aligned}
P_t(X_t \in \bar{S}_t) &\geq P_t\big(\exists x \in \bar{S}_t : \tilde{f}_t(x) \geq \max_{y \in S_t} \tilde{f}_t(y)\big) \\
&\geq P_t\big(\tilde{f}_t(x_t^\star) \geq \max_{y \in S_t} \tilde{f}_t(y)\big)
\end{aligned}$$

$$
\begin{aligned}
&\geq P_t\big(\tilde{f}_t(x_t^\star) \geq \max_{y \in S_t} \tilde{f}_t(y), E_t^{conc}\big) \\
&\geq P_t\big(\tilde{f}_t(x_t^\star) \geq \langle x_t^\star, \bar{\theta}_t \rangle, E_t^{conc}\big) \\
&\geq P_t\big(\tilde{f}_t(x_t^\star) \geq \langle x_t^\star, \bar{\theta}_t \rangle\big) - P_t\big(\bar{E}_t^{conc}\big) \\
&\geq p_3 - p_2.
\end{aligned}
$$

The fourth inequality holds since for any $y \in S_t$, $\tilde{f}_t(y) \leq \langle y, \bar{\theta}_t \rangle + c\|y\|_{V_t^{-1}} \leq \langle y, \bar{\theta}_t \rangle + \Delta_y = \langle x_t^\star, \bar{\theta}_t \rangle$. □

In the following three lemmas, the probability of events E^{wls}, E_t^{conc}, and E_t^{anti} can be controlled with optimal choices of c_1 and c_2 for D-RandLinUCB and D-LinTS algorithms.

Lemma 6.4.1 (Proposition 3, Russac et al. (2019))**.** *For $\lambda > 0$, and*

$$
c_1 = \sqrt{2\log T + d\log(1 + \frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})} + \lambda^{1/2},
$$

then, the event E^{wls} holds with probability at least $1 - 1/T$.

Lemma 6.4.2 (Concentration)**.** *Given history $\mathcal{H}_{t-1}$,*
(a) D-RandLinUCB : $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_t \cdot \|x\|_{V_t^{-1}}$ where $Z_t \sim \mathcal{N}(0, a^2)$, and $c_2 = a\sqrt{2\log(T/2)}$. Then, $P(\bar{E}_t^{conc}) \leq 1/T$.
(b) D-LinTS : $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + x^T W_{t,\lambda}^{-1} \tilde{W}_{t,\lambda}^{1/2} Z^{(t)}$, where $Z^{(t)} \sim \mathcal{N}(0, a^2 I_d)$, and $c_2 = a\sqrt{2\log(KT/2)}$. Then, $P(\bar{E}_t^{conc}) \leq 1/T$.

Proof. (a) We have $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_t \|x\|_{V_t}^{-1}$ in D-RandLinUCB algorithm, and thus

$$
\begin{aligned}
&P(\bar{E}_t^{conc}) = 1 - P(E_t^{conc}) \\
&= 1 - P(\forall x \in \mathcal{X}_t; |\tilde{f}_t(x) - \langle x, \hat{\theta}_t^{wls} \rangle| \leq c_2 \|x\|_{V_t^{-1}}) \\
&= 1 - P(\forall x \in \mathcal{X}_t; |Z_t| \cdot \|x\|_{V_t^{-1}} \leq c_2 \|x\|_{V_t^{-1}}) \\
&= 1 - P(|Z_t| \leq c_2) \because \text{Lemma } C.1.1 \\
&\leq 1/T, \text{ where } c_2 = a\sqrt{2\log(T/2)}.
\end{aligned}
$$

(b) Given history $\mathcal{H}_{t-1}$, we have $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + x^T W_{t,\lambda}^{-1} \tilde{W}_{t,\lambda}^{1/2} Z^{(t)}$ is equivalent to $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_{t,x} \cdot \|x\|_{V_t}^{-1}$ where $Z_{t,x} \sim \mathcal{N}(0, a^2)$ by the linear invariant property of

Gaussian distributions. Thus,

$$
\begin{aligned}
&P(\bar{E}_t^{conc}) = 1 - P(E_t^{conc}) \\
&= 1 - P(\forall x \in \mathcal{X}_t; |\tilde{f}_t(x) - \langle x, \hat{\theta}_t^{wls} \rangle| \leq c_2 \|x\|_{V_t^{-1}}) \\
&= 1 - P(\forall x \in \mathcal{X}_t; |Z_{t,x}| \cdot \|x\|_{V_t^{-1}} \leq c_2 \|x\|_{V_t^{-1}}) \\
&= 1 - P(\forall x \in \mathcal{X}_t; |Z_{t,x}| \leq c_2) \because \text{Lemma } C.1.1 \\
&\leq 1/T, \text{ where } c_2 = a\sqrt{2\log(KT/2)}.
\end{aligned}
$$

□

Lemma 6.4.3 (Anti-concentration)**.** *Given* $\mathcal{H}_{t-1}$,
(a) D-RandLinUCB : $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_t \|x\|_{V_t^{-1}}$*, where* $Z_t \sim \mathcal{N}(0, a^2)$*. Then,* $P(E_t^{anti}) \geq e^{-1/4}/(8\sqrt{\pi})$ *when we have* $a^2 = 14c_1^2$.
(b) D-LinTS : $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + x^T W_{t,\lambda}^{-1} \tilde{W}_{t,\lambda}^{1/2} Z^{(t)}$ *where* $Z^{(t)} \sim \mathcal{N}(0, a^2 I_d)$*. If we assume* $a^2 = 14c_1^2$*, then* $P(E_t^{anti}) \geq e^{-1/4}/(8\sqrt{\pi})$.

Proof. (a) We denote perturbed expected reward as $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_t \|x\|_{V_t}^{-1}$ for D-RandLinUCB. Thus,

$$
\begin{aligned}
P(E_t^{anti}) &= P(\tilde{f}_t(x_t^\star) - \langle x_t^\star, \hat{\theta}_t^{wls} \rangle > c_1 \|x_t^\star\|_{V_t^{-1}}) \\
&= P(Z_t \geq c_1) \\
&\geq \exp\big(-7c_1^2/(2a^2)\big)/(8\sqrt{\pi}) \\
&= e^{-1/4}/(8\sqrt{\pi}) \quad \text{where } a^2 = 14c_1^2.
\end{aligned}
$$

(b) In the same way as the proof of Lemma 6.4.2 (b), $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + x^T W_{t,\lambda}^{-1} \tilde{W}_{t,\lambda}^{1/2} Z^{(t)}$ is equivalent to $\tilde{f}_t(x) = \langle x, \hat{\theta}_t^{wls} \rangle + Z_{t,x} \cdot \|x\|_{V_t}^{-1}$ where $Z_{t,x} \sim \mathcal{N}(0, a^2)$. Thus,

$$
\begin{aligned}
P(E_t^{anti}) &= P(\tilde{f}_t(x_t^\star) - \langle x_t^\star, \hat{\theta}_t^{wls} \rangle > c_1 \|x_t^\star\|_{V_t^{-1}}) \\
&= P(Z_{t,x_t^\star} \geq c_1) \\
&\geq \exp\big(-7c_1^2/(2a^2)\big)/(8\sqrt{\pi}) \\
&= e^{-1/4}/(8\sqrt{\pi}) \quad \text{where } a^2 = 14c_1^2.
\end{aligned}
$$

□

6.4.2 Dynamic regret

The dynamic regret bound of general randomized algorithm is stated below.

Theorem 6.4.2 (Dynamic regret)**.** *Assume we have* $c_1, c_2 \geq 1$ *satisfying* $P(E^{wls}) \geq 1 - p_1$*,* $P(E_t^{conc}) \geq 1 - p_2$*, and* $P(E_t^{anti}) \geq p_3$*, and* $c_3 = 2d\log(\frac{1}{\gamma}) + 2\frac{d}{T}\log(1 + \frac{1}{d\lambda(1-\gamma)})$*. Let* A

be an algorithm that chooses arm $X_t = \arg\max_{\mathcal{X}_t} \tilde{f}_t(x)$ at time t. The expected dynamic regret of A is bounded as for any integer $D > 0$,

$$E[R(T)] \leq (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big)\sqrt{c_3 T} + T(p_1 + p_2) + d + 2\sqrt{\frac{d}{\lambda}} D^{3/2} B_T + \frac{4}{\lambda}\frac{\gamma^D}{1-\gamma} T.$$

Proof. The dynamic regret bound is decomposed into two terms, (A) expected surrogate regret and (B) bias arising from time variation on true parameter,

$$E[R(T)] \leq \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + 2\sum_{t=1}^{T} \|\theta_t^\star - \bar{\theta}_t\|_2.$$

The expected surrogate regret term (A) is bounded by

$$\begin{aligned}
&\sum_{t=d+1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle I\{E^{wls}\}] + T \cdot P(\bar{E}^{wls}) + d \\
&\leq (c_1 + c_2) \sum_{t=1}^{T} \big(1 + \frac{2}{p_3 - p_2}\big) E_t\big[\|X_t\|_{V_t^{-1}}\big] + T(p_1 + p_2) + d \\
&\leq (c_1 + c_2) \sum_{t=1}^{T} \big(1 + \frac{2}{p_3 - p_2}\big) E_t\big[\min(1, \|X_t\|_{V_t^{-1}})\big] + T(p_1 + p_2) + d \\
&\leq (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big)\sqrt{c_3 T} + T(p_1 + p_2) + d
\end{aligned}$$

The first inequality holds due to Theorem 6.4.1. The second inequality works because both dynamic regret and surrogate regret are upper bounded by $2T$ and $c_1 + c_2 \geq 2$. Also, the last inequality holds by Lemma C.2.1 in Section C.2. For any integer $D > 0$, the bias term (B) is bounded as

$$\begin{aligned}
(B) &= 2\sum_{t=1}^{T} \|W_{t,\lambda}^{-1} \sum_{l=1}^{t-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 \\
&\leq 2\sum_{t=1}^{T} \|W_{t,\lambda}^{-1} \sum_{l=t-D}^{t-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 \\
&+ 2\sum_{t=1}^{T} \|W_{t,\lambda}^{-1} \sum_{l=1}^{t-D-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2
\end{aligned}$$

$$
\begin{aligned}
&\leq 2\sum_{t=1}^{T}\sum_{m=t-D}^{t-1}\|W_{t,\lambda}^{-1}\sum_{l=t-D}^{m}\gamma^{-l}X_l X_l^T(\theta_m^\star-\theta_{m+1}^\star)\|_2\\
&+\sum_{t=1}^{T}\frac{2}{\lambda}\|\sum_{l=1}^{t-D-1}\gamma^{t-l-1}X_l X_l^T(\theta_l^\star-\theta_t^\star)\|_2\\
&\leq 2\sqrt{\frac{dD}{\lambda}}\sum_{t=1}^{T}\sum_{m=t-D}^{t-1}\|\theta_m^\star-\theta_{m+1}^\star\|_2+\frac{4}{\lambda}\frac{\gamma^D}{1-\gamma}T\\
&\leq 2\sqrt{\frac{d}{\lambda}}D^{3/2}B_T+\frac{4}{\lambda}\frac{\gamma^D}{1-\gamma}T.
\end{aligned}
$$

The second inequality holds by interchanging the order of summations and $W_{t,\lambda}^{-2}\preccurlyeq(\frac{\gamma^{t-1}}{\lambda})^2 I_d$. The second last inequality works by Lemma C.2.2 □

With the optimal choice of c_1, c_2 and a derived from Lemma 6.4.1-6.4.3, the dynamic regret bounds of D-RandLinUCB and D-LinTS are stated below.

Corollary 6.4.1 (Dynamic regret of D-RandLinUCB)**.** *Suppose*

$$
\begin{aligned}
c_1&=\sqrt{2\log T+d\log(1+\frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})}+\lambda^{1/2},\\
c_2&=a\sqrt{2\log(T/2)},\text{ and } a^2=14c_1^2.
\end{aligned}
$$

Let $\mathcal{A}$ be D-RandLinUCB (Algorithm 3). If B_T is known, then with optimal choice of

$$
D=\frac{\log T}{1-\gamma},\ \gamma=1-d^{-\frac{1}{4}}B_T^{\frac{1}{2}}T^{-\frac{1}{2}},
$$

the expected dynamic regret of $\mathcal{A}$ is asymptotically bounded by $\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$ as $T\to\infty$.

If B_T is unknown, D-RandLinUCB together with Bandits-over-Bandits mechanism enjoys the expected dynamic regret of $\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$.

Corollary 6.4.2 (Dynamic regret of D-LinTS)**.** *Suppose*

$$
\begin{aligned}
c_1&=\sqrt{2\log T+d\log(1+\frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})}+\lambda^{1/2},\\
c_2&=a\sqrt{2\log(KT/2)},\text{ and } a^2=14c_1^2.
\end{aligned}
$$

Let $\mathcal{A}$ be D-LinTS (Algorithm 4). If B_T is known, then with optimal choice of

$$
D=\frac{\log T}{1-\gamma},\ \gamma=1-d^{-\frac{1}{4}}(\log K)^{-\frac{1}{4}}B_T^{\frac{1}{2}}T^{-\frac{1}{2}},
$$

the expected dynamic regret of $\mathcal{A}$ is asymptotically bounded by $\mathcal{O}(d^{\frac{7}{8}}(\log K)^{\frac{3}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$ as $T \to \infty$.

If B_T is unknown, D-LinTS together with Bandits-over-Bandits mechanism enjoys the expected dynamic regret of $\mathcal{O}(d^{\frac{7}{8}}(\log K)^{\frac{3}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$.

The detailed proof of Theorem 6.4.2 and Corollary 6.4.1 and 6.4.2 for the known B_T are deferred to Appendix C.2. The details for the case of unknown B_T are deferred to Appendix C.3.

Note that exponentially discounting weights can be replaced by sliding window strategy or restarted strategy to accommodate to evolving environment. We can construct sliding-window randomized LinUCB (SW-RandLinUCB) and sliding-window linear Thompson sampling (SW-LinTS), or restarting randomized LinUCB (Restart-RandLinUCB) and restarting linear Thompson sampling (Restart-LinTS) via two perturbation approaches, and they maintain the trade-off between oracle efficiency and theoretical guarantee. With unknown total variation B_T, we can also utilize Bandits-over-Bandits mechanism by applying the EXP3 algorithm over these algorithms with different window sizes (Cheung et al., 2019) or epoch sizes (Zhao et al., 2020; Zhao and Zhang, 2021), respectively.

6.4.3 Trade-off between oracle efficiency and theoretical guarantee

Corollary 6.4.1 shows that D-RandLinUCB does not match the lower bound for dynamic regret, $\Omega(d^{2/3}B_T^{1/3}T^{2/3})$, but it achieve the same dynamic regret bound as that of three non-randomized algorithms such as SW-LinUCB, D-LinUCB and Restart-LinUCB. However, D-RandLinUCB is computationally inefficient as D-LinUCB in large action space since the spectral norm of each action in terms of matrix V_t^{-1} should be computed in every round t. In contrast, D-LinTS algorithm relies on offline optimization oracle access via perturbation and thus can be efficiently implemented in infinite-arm setting, and even contextual bandit setting. As a cost of its oracle efficiency, D-LinTS achieves the dynamic regret bound $(\log K)^{3/8}$ worse than that of D-RandLinUCB in finite-arm setting. There exist two variations in D-LinTS; algorithmic variation generated by perturbing an estimate $\hat{\theta}_t^{wls}$ and environmental variation induced by time-varying environments. Two variations are hard to distinguish from the learner's perspective, and thus the effect of algorithmic variation is alleviated by being partially absorbed in environmental variation. This is why D-LinTS and D-LinUCB produce $d^{3/8}$ gap of dynamic regret bounds with infinite set of arms which is less than $d^{1/2}$ gap between regret bounds of LinUCB and LinTS in the stationary environment.

6.5 Numerical experiments

In simulation studies[1], we evaluate the empirical performance of D-RandLinUCB and D-LinTS. We use a sample of 30 days of Criteo live traffic data (Diemert et al., 2017) by 10% downsampling without replacement. Each line corresponds to one impression that was displayed to a user with contextual variables as well as information of whether it was clicked or not. We kept *campaign* variable and categorical variables from *cat1* to *cat9* except for *cat7*. We experiment with several dimensions $d = 10, 20, 50$ and the number of arms $K = 10, 100$. Among all one-hot coded contextual variables, d feature variables were selected by Singular Value Decomposition for dimensionality reduction. We construct two linear models and the model switch occurs at time 4000. The parameter $\theta^\star$ in the initial model is obtained from linear regression model and we obtain true parameter $\theta^\star$ in the second model by switching the signs of 60% of the components of $\theta^\star$. In each round, K arms given to all algorithms are equally sampled from two separate pools of 10000 arms corresponding to clicked or not clicked impressions. The rewards are generated from linear model with additional Gaussian noise of variance $\sigma^2 = 0.15$.

We compare randomized algorithms D-RandLinUCB and D-LinTS to discounted linear UCB (D-LinUCB) as a benchmark. Also, we compare them to linear Thompson sampling (LinTS) and oracle restart LinTS (LinTS-OR). An oracle restart knows about the change-point and restarts the algorithm immediately after the change. In D-RandLinUCB, we use truncated normal distribution with zero mean and standard deviation $2/5$ over $[0, \infty)$ as $\mathcal{D}$ to ensure that its randomly chosen confidence bound belongs to that of D-LinUCB with high probability. Also, we use non-inflated version by setting $a = 1$ when implementing both LinTS and D-LinTS (Vaswani et al., 2020). The regularization parameter is $\lambda = 1$, the time horizon is $T = 10000$ and the cumulative dynamic regret of algorithms are averaged over 100 independent replications in Figure 6.1.

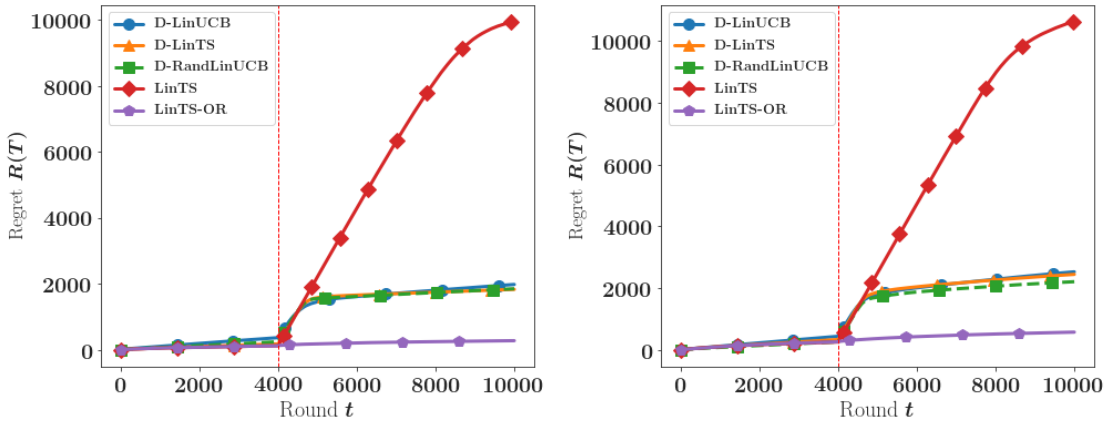
We observe the following patterns in Figure 6.1. First, two randomized algorithms, D-RandLinUCB and D-LinTS outperform the non-randomized one, D-LinUCB when action space is quite large ($K = 100$) in figure 6.1b, 6.1d, and 6.1f. In the setting where the number of arms is small ($K = 10$), however, non-randomized algorithm (D-LinUCB) performs better than two randomized algorithms once relatively high-dimension feature is considered (figure 6.1c and 6.1e), while three nonstationary algorithms show almost similar performance when feature is low-dimensional (figure 6.1a).

Second, D-RandLinUCB always works better than D-LinTS in all scenarios. Though D-LinTS can enjoy oracle efficiency in computational aspect, it has slightly worse regret

[1] https://github.com/baekjin-kim/NonstationaryLB

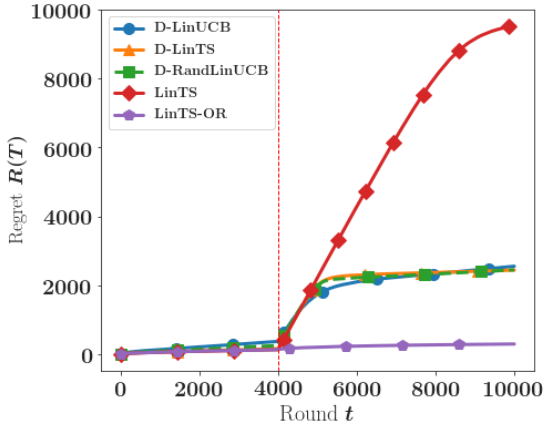
bound than D-RandLunUCB. The difference in theoretical guarantees can be empirically evaluated in this result. The poor performance of D-LinUCB in large action space is due to its very large confidence bound so that the issue regarding conservatism can be partially tackled by randomizing a confidence level in D-RandLinUCB.

Lastly, the interesting observation in figure 6.1f, non-randomized algorithm D-LinUCB shows better performance in recovering a reliable estimator after experiencing a change point than other two competitors in the initial phase. It takes longer time for randomized algorithms to recover their performance. This is because the agent cannot distinguish which factor causes this nonstationarity it is experiencing: either randomness inherited from algorithm nature or environmental change. However, randomized algorithms eventually beat the non-randomized competitor in the final phase.

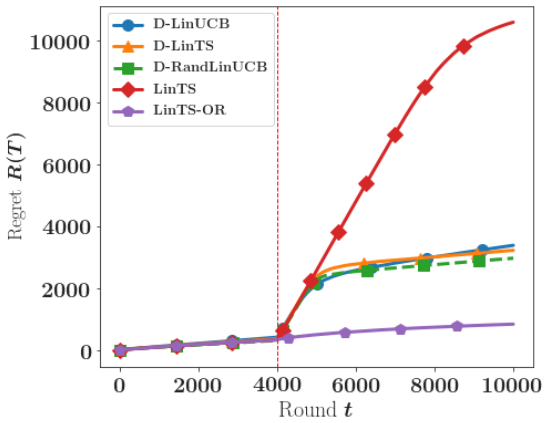


(a) $d = 10, K = 10$

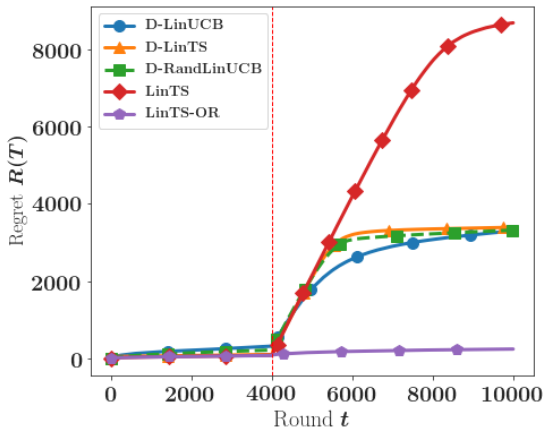
(b) $d = 10, K = 100$



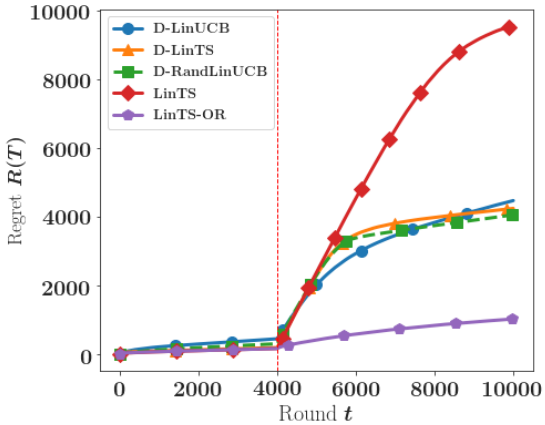
(c) $d = 20, K = 10$



(d) $d = 20, K = 100$



(e) $d = 50, K = 10$



(f) $d = 50, K = 100$

Figure 6.1: Plots of cumulative dynamic regret for algorithms under $d = 10, 20, 50$ and $K = 10, 100$.

CHAPTER 7

On the Equivalence between Online and Private Learnability beyond Binary Classification via Stability

Alon et al. (2019) and Bun et al. (2020) recently showed that online learnability and private PAC learnability are equivalent in binary classification. Alon et al. (2019) showed that private PAC learnability implies finite Littlestone dimension (Ldim) in two steps; (i) every approximately DP learner for a class with Ldim d requires $\Omega(\log^* d)$ thresholds (see Section 2.6 for the definition of $\log^*$), and (ii) the class of thresholds over $\mathbb{N}$ cannot be learned in a private manner. Bun et al. (2020) proved the converse statement via a notion of algorithmic stability, called *global stability*. They showed (i) every class with finite Ldim can be learned by a globally-stable learning algorithm and (ii) they use global stability to derive a DP algorithm.

We investigate whether this equivalence extends to multi-class classification and regression. Our main technical contributions are as follows.

- In Section 7.1, we develop a novel variant of the Littlestone dimension that depends on a tolerance parameter τ, denoted by Ldim_τ. While online learnable regression problems do not naturally reduce to learnable MC problems by discretization, this relaxed complexity measure bridges online MC learnability and regression learnability in that it allows us to consider a regression problem as a relatively simpler MC problem (see Proposition 7.1.1).

- In Section 7.2, we show that private PAC learnability implies online learnability in both MC and regression settings. We appropriately generalize the concept of threshold functions beyond the binary classification setting and lower bound the number of these functions using the complexity measures (see Theorem 7.2.1). Then the ar-

gument of Alon et al. (2019) that an infinite class of thresholds cannot be privately learned can be extended to both settings of interest.

- In Section 7.3, we show that while online learnability continues to imply private learnability in MC (see Theorem 7.3.1), current proof techniques based on *global stability* and *stable histogram* encounter significant obstacles in the regression problem. While this direction for regression setting still remains open, we provide non-trivial sufficient conditions for an online learnable class to also be privately learnable (see Theorem 7.3.3).

7.1 A link between multi-class and regression problems

As a tool to analyze regression problems, we discretize the continuous space $\mathcal{Y}$ into intervals and consider the problem as a multi-class problem. Specifically, given a function $f \in [-1,1]^{\mathcal{X}}$ and a scalar γ, we split the interval $[-1,1]$ into $\lceil \frac{2}{\gamma} \rceil$ intervals of length γ and define $[f]_\gamma(x)$ to be the index of interval that $f(x)$ belongs to. We can also define $[\mathcal{F}]_\gamma = \{[f]_\gamma \mid f \in \mathcal{F}\}$. In this way, if the multi-class problem associated with $[\mathcal{F}]_\gamma$ is learnable, we can infer that the original regression problem is learnable up to accuracy $O(\gamma)$. Quite interestingly, however, the fact that $\mathcal{F}$ is (regression) learnable does not imply that $[\mathcal{F}]_\gamma$ is (multi-class) learnable. For example, it is well known that a class $\mathcal{F}$ of bounded Lipschitz functions on [0,1] is learnable, but $[\mathcal{F}]_1$ includes all binary functions on $[0,1]$, which is not online learnable.

In order to tackle this issue, we propose a generalized zero-one loss in multi-class problems. In particular, we define a *zero-one loss with tolerance* τ,

$$\ell_\tau^{0-1}(\hat{y}; y) = \mathbb{I}(|y - \hat{y}| > \tau).$$

Note that the classical zero-one loss is simply ℓ_0^{0-1}. This generalized loss allows the learner to predict labels that are not equal to the true label but close to it. This property is well-suited in our setting since as far as $|y - \hat{y}|$ is small, the absolute loss in the regression problem remains small.

We also extend the Littlestone dimension with tolerance τ. Fix a tolerance level τ. When we construct a mistake tree T, we add another constraint that each node's descending edges are labeled by two labels $k, k' \in [K]$ such that $\ell_\tau^{0-1}(k; k') = 1$. Let $\mathrm{Ldim}_\tau(\mathcal{H})$ be the maximal height of such binary shattered trees. (Again, $\mathrm{Ldim}_0(\mathcal{H})$ becomes the standard $\mathrm{Ldim}(\mathcal{H})$.)

We record several useful observations. The proofs can be found in Appendix D.1.

Algorithm 5 Standard optimal algorithm with tolerance τ (SOA_τ)

Initialize: $V_0 = \mathcal{H}$
for $t = 1, \cdots, T$ **do**
  Receive x_t
  For $k \in [K]$, let $V_t^{(k)} = \{h \in V_{t-1} \mid h(x_t) = k\}$
  Predict $\hat{y}_t = \arg\max_k \mathrm{Ldim}_\tau(V_t^{(k)})$
  Receive true label y_t and update $V_t = V_t^{(y_t)}$
end for

Lemma 7.1.1. *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a class of multi-class hypotheses.*

1. $\mathrm{Ldim}_\tau(\mathcal{H})$ *is decreasing in* τ.
2. SOA_τ *(Algorithm 5) makes at most* $\mathrm{Ldim}_\tau(\mathcal{H})$ *mistakes with respect to* ℓ_τ^{0-1}.
3. *For any deterministic learning algorithm, an adversary can force* $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ *mistakes with respect to* ℓ_τ^{0-1}.

Equipped with the relaxed loss, the following proposition connects regression learnability to multi-class learnability with discretization. We emphasize that even though the regression learnability does not imply multi-class learnability with the standard zero-one loss, learnability under ℓ_τ^{0-1} can be derived. In addition to that, it can be shown that finite $\mathrm{Ldim}_\tau([\mathcal{F}]_\gamma)$ implies finite $\mathrm{fat}_\gamma(\mathcal{F})$.

Proposition 7.1.1. *Let $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be a regression hypothesis class and suppose* $\mathrm{fat}_\gamma(\mathcal{F}) = d$. *Then we have for any positive integer* n,

$$\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/2(n+1)}) \geq d \geq \mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n}).$$

Proof. Since $\mathrm{fat}_\gamma(\mathcal{F}) = d$, in the online learning setting an adversary can force any deterministic learner to suffer at least $\gamma/2$ absolute loss for d rounds. If we think of this problem as a multi-class classification problem using the hypothesis class $[\mathcal{F}]_{\gamma/2(n+1)}$, using the same strategy, the adversary can force any deterministic learner to make mistakes with respect to ℓ_n^{0-1} for d rounds. Note that the adversary reveals less information to the learner in the discretized multi-class problem. Then Lemma 7.1.1 implies $\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/2(n+1)}) \geq d$.

On the other hand, suppose $\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n}) > d$ and let T be the binary shattered tree with tolerance n. For each node, we can set the witness point to be the middle point between the two labels of descending edges, and the resulting tree is γ-shattered by $\mathcal{F}$. This contradicts the fact that $\mathrm{fat}_\gamma(\mathcal{F}) = d$, and hence we obtain $d \geq \mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n})$. □

There exist a few works that used regression models in multi-class classification (Rakesh and Suganthan, 2017; Yang et al., 2005). To the best of our knowledge, however, our work is the first one that studies regression learnability by transforming the problem into a discretized classification problem along with a novel bridge, *Littlestone dimension with tolerance*.

7.2 Private learnability implies online learnability

In this section, we show that if a class of functions is privately learnable, then it is online learnable. To do so, we prove a lower bound of the sample complexity of privately learning algorithms using either $\mathrm{Ldim}(\mathcal{H})$ for the multi-class hypotheses or $\mathrm{fat}_\gamma(\mathcal{F})$ for the regression hypotheses. Alon et al. (2019) proved this in the binary classification setting first by showing that any large Ldim class contains sufficiently many threshold functions and then providing a lower bound of the sample complexity to privately learn threshold functions. We adopt their arguments, but one of the first non-trivial tasks is to define analogues of threshold functions in multi-class or regression problems. Note that, a priori, it is not clear what the right analogy is. Let us first introduce threshold functions in the binary case. We say a binary hypothesis class $\mathcal{H}$ has n thresholds if there exist $\{x_i\}_{1:n} \subset \mathcal{X}$ and $\{h_i\}_{1:n} \subset \mathcal{H}$ such that $h_i(x_j) = 1$ if $i \leq j$ and $h_i(x_j) = 0$ if $i > j$. We extend this as below.

Definition 7.2.1 (Threshold functions in multi-class problems)**.** Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a hypothesis class. We say $\mathcal{H}$ contains n thresholds with a gap τ if there exist $k, k' \in [K]$, $\{x_i\}_{1:n} \subset \mathcal{X}$, and $\{h_i\}_{1:n} \subset \mathcal{H}$ such that $|k - k'| > \tau$ and $h_i(x_j) = k$ if $i \leq j$ and $h_i(x_j) = k'$ if $i > j$.

Definition 7.2.2 (Threshold functions in regression problems)**.** Let $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be a hypothesis class. We say $\mathcal{F}$ contains n thresholds with a margin γ if there exist $\{x_i\}_{1:n} \subset \mathcal{X}$, $\{f_i\}_{1:n} \subset \mathcal{F}$, and $u, u' \in [-1, 1]$ such that $|u - u'| \geq \gamma$ and $|f_i(x_j) - u| \leq \frac{\gamma}{20}$ if $i \leq j$ and $|f_i(x_j) - u'| \leq \frac{\gamma}{20}$ if $i > j$.

In Definition 7.2.2, we allow the functions to oscillate with a margin $\frac{\gamma}{20}$ which is arbitrary. Any small margin compared to $|u - u'|$ would work, but this number is chosen to facilitate later arguments.

Next we show that complex hypothesis classes contain a sufficiently large set of threshold functions. The following theorem extends the results by Alon et al. (2019, Theorem 3). A complete proof can be found in Appendix D.2.

Algorithm 6 COLORANDCHOOSE

1: **Input:** multi-class hypothesis class $\mathcal{H} \subset [K]^{\mathcal{X}}$, shattered binary tree T, tolerance τ
2: Choose an arbitrary hypothesis $h_0 \in \mathcal{H}$
3: Color each vertex x of T by $h_0(x) \in [K]$
4: Find a color k such that the sub-tree $T' \subset T$ of color k has the largest height
5: Let x_0 be the root node of T'
6: Let x_1 be a child of x_0 such that the edge (x_0, x_1) is labeled as k' with $|k - k'| > \frac{\tau}{2}$
7: Let T'' be a sub-tree of T' rooted at x_1
8: Let $\mathcal{H}' = \{h \in \mathcal{H} \mid h(x_0) = k'\}$
9: **Output:** $k, k', h_0, x_0, \mathcal{H}', T''$

Theorem 7.2.1 (Existence of a large set of thresholds)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively.*

1. *If $\mathrm{Ldim}_{2\tau}(\mathcal{H}) \geq d$, then $\mathcal{H}$ contains $\lfloor \frac{\log_K d}{K^2} \rfloor$ thresholds with a gap τ.*
2. *If $\mathrm{fat}_\gamma(\mathcal{F}) \geq d$, then $\mathcal{F}$ contains $\lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ thresholds with a margin $\frac{\gamma}{5}$.*

Proof sketch. We begin with the multi-class setting. Suppose $d = K^{K^2 t}$. It suffices to show $\mathcal{H}$ contains t thresholds. Let T be a shattered binary tree of height d and tolerance 2τ. Letting $\mathcal{H}_0 = \mathcal{H}$ and $T_0 = T$, we iteratively apply COLORANDCHOOSE (Algorithm 6). Namely, we write

$$k_n, k'_n, h_n, x_n, \mathcal{H}_n, T_n = \text{COLORANDCHOOSE}(\mathcal{H}_{n-1}, T_{n-1}, 2\tau). \tag{7.1}$$

Observe that for all n, we can infer $h_n(x_n) = h_n(x) = k_n$ for all internal vertices x of T_n ($\because$ line 4 of Algorithm 6) and $h(x_n) = k'_n$ for all $h \in \mathcal{H}_n$ ($\because$ line 8 of Algorithm 6).

Additionally, it can be shown that the height of T_n is no less than $\frac{1}{K}$ times the height of T_{n-1} (see Lemma D.2.1 in Appendix D.3). This means that the iterative step (7.1) can be repeated $K^2 t$ times since $d = K^{K^2 t}$. Then there exist k, k' and indices $\{n_i\}_{i=1}^t$ such that $k_{n_i} = k$ and $k'_{n_i} = k'$ for all i.

It is not hard to check that the functions $\{h_{n_i}\}_{1:t}$ and the arguments $\{x_{n_i}\}_{1:t}$ form thresholds with labels k, k'. Since $|k - k'| > \tau$ ($\because$ line 6 of Algorithm 6), this completes the proof.

The result in the regression setting can also be shown in a similar manner using Proposition 7.1.1. □

Alon et al. (2019, Theorem 1) proved a lower bound of the sample complexity in order to privately learn threshold functions. Then the multi-class result (with $\tau = 0$) of Theorem 7.2.1 immediately implies that if $\mathcal{H}$ is privately learnable, then it is online learnable. For the regression case, we need to slightly modify the argument to deal with the margin condition

in Definition 7.2.2. The next theorem summarizes the result, and the proof appears in Appendix D.2.

Theorem 7.2.2 (Lower bound of the sample complexity to privately learn thresholds)**.** *Let $\mathcal{F} = \{f_i\}_{1:n} \subset [-1,1]^{\mathcal{X}}$ be a set of threshold functions with a margin γ on a domain $\{x_i\}_{1:n} \subset \mathcal{X}$ along with bounds $u, u' \in [-1,1]$. Suppose $\mathcal{A}$ is a $(\frac{\gamma}{200}, \frac{\gamma}{200})$-accurate learning algorithm for $\mathcal{F}$ with sample complexity m. If $\mathcal{A}$ is (ϵ, δ)-DP with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$, then it can be shown that $m \geq \Omega(\log^* n)$.*

Combining Theorem 7.2.1 and 7.2.2, we present our main result.

Corollary 7.2.1 (Private learnability implies online learnability)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively. Let $\mathrm{Ldim}(\mathcal{H}) = \mathrm{fat}_\gamma(\mathcal{F}) = d$. Suppose there is a learning algorithm $\mathcal{A}$ that is $(\frac{1}{16}, \frac{1}{16})$-accurate for $\mathcal{H}$ ($(\frac{\gamma}{200}, \frac{\gamma}{200})$-accurate for $\mathcal{F}$) with sample complexity m. If $\mathcal{A}$ is (ϵ, δ)-DP with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$, then $m \geq \Omega(\log^* d)$.*

7.3 Online learnability implies private learnability

In this section, we show that online-learnable multi-class hypothesis classes can be learned in a DP manner. For regression hypothesis classes, we provide sufficient conditions for private learnability.

7.3.1 Multi-class classification

Bun et al. (2020) proved that every binary hypothesis class with a finite Ldim is privately learnable by introducing a new notion of algorithmic stability called *global stability* as an intermediate property between online learnability and differentially-private learnability. Their arguments can be naturally extended to MC hypothesis classes, which is summarized in the next theorem.

Theorem 7.3.1 (Online MC learning implies private MC learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$. Let $\epsilon, \delta \in (0,1)$ be privacy parameters and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For $n = O_d\big(\frac{\log(1/\beta\delta)}{\alpha\epsilon}\big)$, there exists an (ϵ, δ)-DP learning algorithm such that for every realizable distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies $\mathrm{loss}_{\mathcal{D}}(f) \leq \alpha$ with probability at least $1 - \beta$.*

While we consider the realizable setting in Theorem 7.3.1, a similar result also holds in the agnostic setting. The extension to the agnostic setting is discussed in Appendix D.3.3 due to limited space.

As a key to the proof of Theorem 7.3.1, we introduce global stability (GS) as follows.

Definition 7.3.1 (Global stability (Bun et al., 2020))**.** Let $n \in \mathbb{N}$ be a sample size and $\eta > 0$ be a global stability parameter. An algorithm $\mathcal{A}$ is (n, η)-GS with respect to $\mathcal{D}$ if there exists a hypothesis h such that $\mathbb{P}_{S\sim\mathcal{D}^n}\big(\mathcal{A}(S) = h\big) \geq \eta$.

Theorem 7.3.1 can be proved in two steps. We first show that every MC hypothesis class with a finite Ldim is learnable by a GS algorithm $\mathcal{A}$ (Theorem 7.3.2). Then we prove that any GS algorithm can be extended to a DP learning algorithm with a finite sample complexity.

Theorem 7.3.2 (Online MC learning implies GS learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$. Let $\alpha > 0$, and $m = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{\alpha}]$. Then there exists a randomized algorithm $G : (\mathcal{X} \times [K])^m \to [K]^{\mathcal{X}}$ such that for a realizable distribution $\mathcal{D}$ and an input sample $S \sim \mathcal{D}^m$, there exists a h such that*

$$\mathbb{P}\big(G(S) = h\big) \geq \frac{K-1}{(d+1)K^{d+1}} \quad \textit{and} \quad loss_{\mathcal{D}}(h) \leq \alpha.$$

Next, we give a brief overview on how to construct a GS learner G and a DP learner M in order to prove Theorem 7.3.1. The complete proofs are deferred to Appendix D.3.

7.3.1.1 Online multi-class learning implies globally-stable learning

Let $\mathcal{H}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$ and $\mathcal{D}$ be a realizable distribution over examples $\big(x, c(x)\big)$ where $c \in \mathcal{H}$ is an unknown target hypothesis. Recall that $\mathcal{H}$ is learnable by SOA_0 (Algorithm 5) with at most d mistakes on any realizable sequence. Prior to building a GS learner G, we construct a distribution $\mathcal{D}_k$ by appending k *tournament examples* between random samples from $\mathcal{D}$, which force SOA_0 to make at least k mistakes when run on S drawn from $\mathcal{D}_k$. Using the fact that SOA_0 identifies the true labeling function after making d mistakes, we can show that there exists $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) \geq K^{-d}.$$

A GS learner G is built by firstly drawing $k \in \{0, 1, \cdots, d\}$ uniformly at random and then running the SOA_0 on $S \circ T$ where $S \sim \mathcal{D}_k, T \sim \mathcal{D}^n$. The learner G outputs a good

hypothesis that enjoys small population loss with probability at least $\frac{K-d}{d+1}$. We defer the detailed construction of $\mathcal{D}_k$ and proofs to Appendix D.3.

7.3.1.2 Globally-stable learning implies private multi-class learning

Let G be a (η, m)-GS algorithm with respect to a target distribution $\mathcal{D}$. We run G on k independent samples of size m to non-privately produce a long list $H := (h_i)_{1:k}$. The *Stable Histogram* algorithm is a primary tool that allows us to publish a short list of frequent hypotheses in a DP manner. The fact that G is GS ensures that some good hypotheses appear frequently in H. Then Lemma 7.3.1 implies that these good hypotheses remain in the short list with high probability. Once we obtain a short list, a generic DP learning algorithm (Kasiviswanathan et al., 2011) is applied to privately select an accurate hypothesis.

Lemma 7.3.1 (Stable Histogram (Dwork et al., 2006; Korolova et al., 2009))**.** *Let X be any data domain. For $n \geq O(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon})$, there exists an (ϵ, δ)-DP algorithm* HIST *which with probability at least $1-\beta$, on input $S = (x_i)_{1:n}$ outputs a list $L \subset X$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that (i) every x with $\mathrm{Freq}_S(x) \geq \eta$ appears in L, and (ii) for every $x \in L$, the estimate a_x satisfies $|a_x - \mathrm{Freq}_S(x)| \leq \eta$ where $\mathrm{Freq}_S(x) := \big|\{i \in [n] \mid x_i = x\}\big|/n$.*

7.3.2 Regression

In classification, *Global Stability* was an essential intermediate property between online and private learnability. A natural approach to obtaining a DP algorithm from an online-learnable real-valued function class $\mathcal{F}$ is to transform the problem into a multi-class problem with $[\mathcal{F}]_\gamma$ for some γ and then construct a GS learner using the previous techniques. If $[\mathcal{F}]_\gamma$ is privately-learnable, then we can infer that the original regression problem is also private-learnable up to an accuracy $O(\gamma)$.

Unfortunately, however, finite $\mathrm{fat}_\gamma(\mathcal{F})$ only implies finite $\mathrm{Ldim}_1([\mathcal{F}]_\gamma)$, and $\mathrm{Ldim}([\mathcal{F}]_\gamma)$ can still be infinite (see Proposition 7.1.1). This forces us to run SOA_1 instead of SOA_0, and as a consequence, after making $\mathrm{Ldim}_1([\mathcal{F}]_\gamma)$ mistakes, the algorithm can identify the true function up to some tolerance. Therefore we only get the relaxed version of GS property as follows; there exist $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_1(S \circ T) \approx_1 f\big) \geq (\gamma/2)^d$$

where $f \approx_1 g$ means $\sup_{x\in\mathcal{X}} \big|f(x) - g(x)\big| \leq 1$. If we proceed with this relaxed condition, it is no longer guaranteed the long list H contains a good hypothesis with suffi-

ciently high frequency. This hinders us from using Lemma 7.3.1, and a private learner cannot be produced in this manner. The limitation of proving the equivalence in regression stems from existing proof techniques. With another method, it is still possible to show that online-learnable real-valued function classes can be learned by a DP algorithm. Instead, we provide sufficient conditions for private learnability in regression problems.

Theorem 7.3.3 (Sufficient conditions for private regression learnability)**.** *Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a real-valued function class such that $\mathrm{fat}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$. If one of the following conditions holds, then $\mathcal{F}$ is privately learnable.*

1. *Either $\mathcal{F}$ or $\mathcal{X}$ is finite.*
2. *The range of $\mathcal{F}$ over $\mathcal{X}$ is finite (i.e., $\left|\{f(x) \mid f \in \mathcal{F}, x \in \mathcal{X}\}\right| < \infty$).*
3. *$\mathcal{F}$ has a finite cover with respect to the sup-norm at every scale.*
4. *$\mathcal{F}$ has a finite sequential Pollard Pseudo-dimension.*

We present the proof of Condition 4, and proofs of other conditions are deferred to Appendix D.3.4.

Proof of Condition 4. Assume for contradiction that there exists γ such that $\mathrm{Ldim}([\mathcal{F}]_\gamma) = \infty$. Then we can obtain a shattered tree T of an arbitrary depth. Choose an arbitrary node x. Note that its descending edges are labeled by $k, k' \in [\lceil 2/\gamma \rceil]$. We can always find a witness to shattering s between the intervals corresponding to k and k'. With these witness values, the tree T must be zero-shattered by $\mathcal{F}$. Since the depth of T can be arbitrarily large, this contradicts to $\mathrm{Pdim}(\mathcal{F})$ being finite. From this, we can claim that $\mathrm{Ldim}([\mathcal{F}]_\gamma) \leq \mathrm{Pdim}(\mathcal{F})$ for any γ. Then using the ideas in Section 7.3.1, we can conclude that $[\mathcal{F}]_\gamma$ is private-learnable for any γ. Therefore the original class $\mathcal{F}$ is also private-learnable. □

We emphasize that Conditions 3 and 4 do not imply each other. For example, a class of point functions $\mathcal{F}^{\text{point}} := \{\mathbb{I}(\cdot = x) \mid x \in \mathcal{X}\}$ does not have a finite sup-norm cover because any two distinct functions have the sup-norm difference one, but $\mathrm{Pdim}(\mathcal{F}^{\text{point}}) = 1$. A class $\mathcal{F}^{\text{Lip}}$ of bounded Lipschitz functions on $[0, 1]$ has an infinite sequential Pollard pseudo-dimension, but $\mathcal{F}^{\text{Lip}}$ has a finite cover with respect to the sup-norm due to compactness of $[0, 1]$ along with the Lipschitz property.

CHAPTER 8

Conclusion

Online learning is a well-developed branch of machine learning that studies how the learner dynamically updates models as new data instances arrive in a sequential fashion. This thesis investigates the notion of *stability in online learning* in the following two directions. First, we examine the *random perturbation* techniques as a source of stability along with regularization techniques in *bandit problems*. Second, we consider *differential privacy* and study stability as the concept of connecting online learning and differential privacy.

In Chapter 3, we studied the statistical optimality of *perturbation technique* in stochastic multi-armed bandit problems. We provide a unified regret analysis for both sub-Weibull and bounded perturbations when rewards are sub-Gaussian. Our bounds are instance optimal for sub-Weibull perturbations with parameter 2 that also have a matching lower tail bound, and all bounded support perturbations where there is sufficient probability mass at the extremes of the support. We believe that this chapter paves the way for similar extension for more complex settings, e.g., stochastic linear bandits, stochastic partial monitoring, and Markov decision processes.

In Chapter 4, we showed that the open problem regarding minimax optimal perturbations for adversarial multi-armed bandit problems cannot be solved in two ways that might seem very natural from discrete choice model and extreme value theory, respectively. While our results are negative, they do point the way to a possible affirmative solution of the problem. They led us to a conjecture that the optimal perturbation, if it exists, will be of Fréchet-type.

In Chapter 5, we examined stationary stochastic linear bandits and explicate the role of two perturbation approaches in overcoming conservatism that UCB-type algorithms chronically suffer from in practice. In one approach, we replace optimism with a simple randomization when using confidence sets. In the other, we add random perturbations to the current estimate before maximizing the expected reward. These two approaches result in randomized LinUCB and Gaussian linear Thompson sampling for stationary linear bandits. We highlight the statistical optimality versus oracle efficiency trade-off between them.

In Chapter 6, we considered non-stationary stochastic linear bandits, developed two randomized exploration strategies, and investigated the trade-off between theoretical guarantee and computational efficiency embedded in two design principles of randomized algorithms constructed (1) by replacing optimism with a simple randomization when deciding a confidence level in UCB-type algorithms, or (2) by adding the random perturbations to estimates.

In Chapter 7, it has been recently shown that online learning and differential-private learning are equivalent in binary classification via the notion of *stability*, and we investigated whether this equivalence extends to multi-class classification and regression. We proved that private learnability implies online learnability in the MC and regression settings. We also showed the converse in the MC setting and provided sufficient conditions for an online learnable class to also be privately learnable in regression problems.

8.1 Future work

There are a few general directions of bandit and differential privacy research which may be particularly interesting in the future.

8.1.1 Best of both worlds in nonstationary stochastic linear bandit : parameter-free and optimal in total variation and number of distribution changes

In nonstationary bandit settings, we measure the nonstationarity of the environment by the *total number of distribution changes* S or by the *total variation* V. It is well-known that Auer et al. (2002) and Besbes et al. (2014) investigate the classical multi-armed bandit problem and develop adaptive algorithms with prior knowledge about the amount of nonstationarity S and V, respectively. Auer et al. (2019) proposes the first parameter-free algorithm with dynamic regret that is optimal in relevant parameters T, K, S and V. In the contextual bandit problem, Chen et al. (2019) suggests a fully adaptive, minimax-optimal and oracle-efficient algorithm assuming access to an optimization oracle when S and V are unknown for the learner. This algorithm relies on ILOVETOCONBANDITS algorithm of Agarwal et al. (2014) and replay phases for detection purpose.

Nonstationarity in linear bandit has been only studied with prior information of total variation V via sliding window (Cheung et al., 2019), exponential discounting weights (Russac et al., 2019; Kim and Tewari, 2020), and restarting regularly (Zhao et al., 2020). It is a very interesting open problem to develop a *parameter-free* algorithm with optimal

dynamic regret in relevant parameters T, K, S and V. Furthermore, it remains open to examine the adaptive algorithm in non-stationary infinite-horizon MDP setting without knowledge of three quantities: number of distribution switches, total variance and diameter.

8.1.2 Sublinear algorithms in nonstationary kernelized linear bandit : weighting, sliding window, and regularly restarting

We formalize the task of optimizing an unknown, noisy reward function f that is expensive to evaluate as a bandit problem. If the unknown reward function is linear, it represents the standard linear bandit problem. The reward function is either sampled from a Gaussian process in Bayesian optimization or a fixed function in an RKHS with a bounded norm in frequentist setting. Srinivas et al. (2010) developed GP-UCB algorithm with sub-linear regret in T. The result and analysis were later improved in Valko et al. (2013) and Chowdhury and Gopalan (2017). Bogunovic et al. (2016) considered the sequential Bayesian optimization problem with bandit feedback when reward function is allowed to vary with time. This work is quite limited in that reward function is generated from a Gaussian process that evolves according a simple Markov model. In nonstationary (time-varying) linear bandit problems, Cheung et al. (2019); Russac et al. (2019), and Zhao et al. (2020) developed no-regret algorithms based on approaches of sliding window, exponential discounting weights, and restarting regularly, respectively. It would be an interesting open problem to investigate a fully adaptive algorithm with optimal dynamic regret in the setting where the time-varying environment is captured by the RKHS norm by $V = \sum_{t=1}^{T-1} \|f_t - f_{t-1}\|_{\mathcal{H}}$. They might be based on three approaches used in nonstationary linear bandits. Also, it remains open to develop an adaptive algorithm in the piecewise stationary kernelized linear bandits.

8.1.3 Perturbation based algorithm under corrupted or delayed bandit feedback

It would be interesting to study perturbation methods in bandit problems under more realistic assumptions, *delayed* or *corrupted* feedback, which is motivated from my internship project at Twitter. In many ad systems like Twitter's, positive responses of users are only observed after a possibly long and random delay after ad is served, making it challenging to build a representative dataset of ad engagements in real time. Also, ad systems are sometimes exposed to click fraud via malware that effectively sabotage an ad campaign run by a competitor, and cause a typical online advertising platform to reject those ads from con-

sideration. Though perturbation approaches have been empirically observed to be robust to corrupted or delayed environment, theoretical guarantees are not available yet since their innate randomness makes their mathematical analysis difficult.

Lykouris et al. (2018) first studied the corrupted setting for multi-armed bandits. This were improved by Gupta et al. (2019) and Zimmert and Seldin (2019) and later extended to linear bandit (Li et al., 2019; Bogunovic et al., 2021), Gaussian bandit (Bogunovic et al., 2020), and reinforcement learning (Lykouris et al., 2019). The algorithms for complex bandits extended from the classical multi-armed bandits are designed based on successive arm elimination and achieve regret bounds multiplied by a corruption level C, though it is ideal to the regret of a robust stochastic algorithm degrade with an additive term $\mathcal{O}(C)$. It remains widely open to develop optimal algorithms, especially *randomized* ones, and analyze their regret bounds for linear bandit, Gaussian Process bandit, and reinforcement learning under corrupted feedback.

8.1.4 Open problems in differential privacy

We have a few suggestions for future work on differential privacy. First, we need to understand whether online learnability implies private learnability in the regression setting. Second, like Bun et al. (2020), we create an improper DP learner for an online learnable class. It would be interesting to see if we can construct *proper* DP learners. Third, Gonen et al. (2019) provide an efficient black-box reduction from *pure* DP learning to online learning. It is natural to explore whether such efficient reductions are possible for *approximate* DP algorithms for MC and regression problems. Finally, there are huge gaps between the lower and upper bounds for sample complexities in both classification and regression settings. It would be desirable to show tighter bounds and reduce these gaps.

APPENDIX A

Detailed Proofs for Stochastic Multi-armed Bandits

In this section, the proofs omitted in Chapter 3 are presented.

A.1 Proof of Theorem 3.2.1

Proof. For each arm $i \neq 1$, we will choose two thresholds $x_i = \mu_i + \frac{\Delta_i}{3}, y_i = \mu_1 - \frac{\Delta_i}{3}$ such that $\mu_i < x_i < y_i < \mu_1$ and define two types of events, $E_i^{\mu}(t) = \{\hat{\mu}_i(t) \leq x_i\}$, and $E_i^{\theta}(t) = \{\theta_i(t) \leq y_i\}$. Intuitively, $E_i^{\mu}(t)$ and $E_i^{\theta}(t)$ are the events that the estimate $\hat{\mu}_i$ and the sample value $\theta_i(t)$ are not too far above the mean μ_i, respectively. $\mathrm{E}[T_i(T)] = \sum_{t=1}^{T} \mathrm{P}(A_t = i)$ is decomposed into the following three parts according to events $E_i^{\mu}(t)$ and $E_i^{\theta}(t)$,

$$\mathrm{E}[T_i(T)] = \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, (E_i^{\mu}(t))^c)}_{(a)} + \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^{\mu}(t), (E_i^{\theta}(t))^c)}_{(b)} + \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^{\mu}(t), E_i^{\theta}(t))}_{(c)}$$

Let τ_k denote the time at which k-th trial of arm i happens. Set $\tau_0 = 0$.

$$\begin{aligned}(a) &\leq 1 + \sum_{k=1}^{T-1} \mathrm{P}((E_i^{\mu}(\tau_k + 1))^c) \\ &\leq 1 + \sum_{k=1}^{T-1} \exp\big(-\frac{k(x_i - \mu_i)^2}{2}\big) \leq 1 + \frac{18}{\Delta_i^2}. \end{aligned} \tag{A.1}$$

The probability in part (b) is upper bounded by 1 if $T_i(t)$ is less than $L_i(T) = \frac{\sigma^2[2\log(T\Delta_i^2)]^{2/p}}{(y_i - x_i)^2}$, and by $C_a/(T\Delta_i^2)$ otherwise. The latter can be proved as below,

$$\begin{aligned}
&\mathrm{P}(A_t = i, (E_i^\theta(t))^c | E_i^\mu(t)) \\
&\leq \mathrm{P}(\theta_i(t) > y_i | \hat{\mu}_i(t) \leq x_i) \\
&\leq \mathrm{P}\Big(\frac{Z_{it}}{\sqrt{T_i(t)}} > y_i - x_i | \hat{\mu}_i(t) \leq x_i\Big) \\
&\leq C_a \cdot \exp\big(-\frac{T_i(t)^{p/2}(y_i - x_i)^p}{2\sigma^p}\big) \leq \frac{C_a}{T\Delta_i^2} \;\text{ if }\; T_i(t) \geq L_i(T).
\end{aligned}$$

The third inequality holds by sub-Weibull (p) assumption on perturbation Z_{it}. Let τ be the largest step until $T_i(t) \leq L_i(T)$, then part (b) is bounded as,

$$(b) \leq L_i(T) + \sum_{t=\tau+1}^{T} C_a/(T\Delta_i^2) \leq L_i(T) + C_a/\Delta_i^2.$$

Regarding part (c), define $p_{i,t}$ as the probability $p_{i,t} = \mathrm{P}(\theta_1(t) > y_i | \mathcal{H}_{t-1})$ where $\mathcal{H}_{t-1}$ is defined as the history of plays until time $t-1$. Let δ_j denote the time at which j-th trial of arm 1 happens.

Lemma A.1.1 (Lemma 1 (Agrawal and Goyal, 2013a)). *For $i \neq 1$,*

$$\begin{aligned}
(c) &= \sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^\mu(t), E_i^\theta(t)) \\
&\leq \sum_{t=1}^{T} \mathrm{E}\Big[\frac{1 - p_{i,t}}{p_{i,t}} I(A_t = 1, E_i^\mu(t), E_i^\theta(t))\Big] \leq \sum_{j=0}^{T-1} \mathrm{E}\Big[\frac{1 - p_{i,\delta_j+1}}{p_{i,\delta_j+1}}\Big].
\end{aligned}$$

Proof. See Section A.2. □

The average rewards from the first arm, $\hat{\mu}_1(\delta_j + 1)$, has a density function denoted by $\phi_{\hat{\mu}_{1,j}}$.

$$\begin{aligned}
\mathrm{E}\Big[\frac{1 - p_{i,\delta_j+1}}{p_{i,\delta_j+1}}\Big] &= \mathrm{E}\Big[\frac{1}{\mathrm{P}(\theta_1(\delta_j + 1) \geq y_i | \mathcal{H}_{\delta_j+1})} - 1\Big] \\
&= \int_{\mathbb{R}} \Big[\frac{1}{\mathrm{P}\big(x + \frac{Z}{\sqrt{j}} > \mu_1 - \frac{\Delta_i}{3}\big)} - 1\Big] \phi_{\hat{\mu}_{1,j}}(x) dx
\end{aligned}$$

The above integration is divided into three intervals, $(-\infty, \mu_1 - \frac{\Delta_i}{3}], (\mu_1 - \frac{\Delta_i}{3}, \mu_1 - \frac{\Delta_i}{6}]$,

and $(\mu_1 - \frac{\Delta_i}{3}, \infty)$. We denote them as $(1), (2)$ and (3), respectively.

$$
\begin{aligned}
&\int_{-\infty}^{\mu_1-\frac{\Delta_i}{3}} \Big[\frac{1}{\mathrm{P}\big(Z > -\sqrt{j}(x-\mu_1+\frac{\Delta_i}{3})\big)} - C_b\Big]\phi_{\hat{\mu}_{1,j}}(x)dx \\
&= \int_0^{\infty} \Big[\frac{1}{\mathrm{P}(Z>u)} - C_b\Big]\frac{1}{\sqrt{j}}\phi_{\hat{\mu}_{1,j}}\big(-\frac{u}{\sqrt{j}} + \mu_1 - \frac{\Delta_i}{3}\big)du \;\; \because u = -\sqrt{j}\big(x-\mu_1+\frac{\Delta_i}{3}\big) \\
&\le \int_0^{\infty} \Big[C_b \cdot \exp\big(\frac{u^q}{2\sigma^q}\big) - C_b\Big]\frac{1}{\sqrt{j}}\phi_{\hat{\mu}_{1,j}}\big(-\frac{u}{\sqrt{j}} + \mu_1 - \frac{\Delta_i}{3}\big)du \\
&= \int_0^{\infty} \Big[\int_0^u G'(v)dv\Big]\frac{1}{\sqrt{j}}\phi_{\hat{\mu}_{1,j}}\big(-\frac{u}{\sqrt{j}} + \mu_1 - \frac{\Delta_i}{3}\big)du \;\; \because G(u) = C_b \cdot \exp\big(\frac{u^q}{2\sigma^q}\big) \\
&\le \int_0^{\infty} \exp\big(-\frac{(v+\frac{\sqrt{j}\Delta_i}{3})^2}{2}\big)\cdot G'(v)dv \;\; \because \text{Fubini's Theorem } \& \text{ sub-Gaussian} \\
&= \int_0^{\infty} \exp\big(-\frac{(v+\frac{\sqrt{j}\Delta_i}{3})^2}{2}\big)\cdot C_b\frac{qv^{q-1}}{2\sigma^q}\exp\big(\frac{v^q}{2\sigma^q}\big)dv \\
&\le C_b M_{q,\sigma}\exp\big(-\frac{j\Delta_i^2}{18}\big) \quad \because \exists\, 0 < M_{q,\sigma} < \infty \text{ if } q<2 \text{ or } (q=2, \sigma \ge 1)
\end{aligned}
$$

$$
\begin{aligned}
(1) &= \int_{-\infty}^{\mu_1-\frac{\Delta_i}{3}} \Big[\big[\frac{1}{\mathrm{P}\big(Z > -\sqrt{j}(x-\mu_1+\frac{\Delta_i}{3})\big)} - C_b\big]\phi_{\hat{\mu}_{1,j}}(x) + (C_b-1)\phi_{\hat{\mu}_{1,j}}(x)\Big]dx \\
&\le C_b M_{q,\sigma}\exp\big(-\frac{j\Delta_i^2}{18}\big) + (C_b-1)\exp\big(-\frac{j\Delta_i^2}{18}\big) \\
(2) &= \int_{\mu_1-\frac{\Delta_i}{3}}^{\mu_1-\frac{\Delta_i}{6}} 2\mathrm{P}\big(Z < -\sqrt{j}(x-\mu_1+\frac{\Delta_i}{3})\big)\phi_{\hat{\mu}_{1,j}}(x)dx \\
&\le 2\mathrm{P}(Z<0)\cdot \mathrm{P}\big(\mu_1 - \frac{\Delta_i}{3} \le \hat{\mu}_{1,j} \le \mu_1 - \frac{\Delta_i}{6}\big) \\
&\le 2\mathrm{P}\big(\hat{\mu}_{1,j} \le \mu_1 - \frac{\Delta_i}{6}\big) \le 2\exp\big(-\frac{j\Delta_i^2}{72}\big) \\
(3) &= \int_{\mu_1-\frac{\Delta_i}{6}}^{\infty} 2\mathrm{P}\big(Z < -\sqrt{j}(x-\mu_1+\frac{\Delta_i}{3})\big)\phi_{\hat{\mu}_{1,j}}(x)dx \\
&\le 2\mathrm{P}\big(Z < -\frac{\sqrt{j}\Delta_i}{6}\big)\int_{\mu_1-\frac{\Delta_i}{6}}^{\infty}\phi_{\hat{\mu}_{1,j}}(x)dx \le 2\mathrm{P}\big(Z < -\frac{\sqrt{j}\Delta_i}{6}\big) \le 2C_a\exp\big(-\frac{j^{p/2}\Delta_i^p}{2\cdot(6\sigma)^p}\big)
\end{aligned}
$$

$$
(c) = \sum_{j=0}^{T-1}(1)+(2)+(3) < \frac{18C_b(M_{q,\sigma}+1)+126}{\Delta_i^2} + \frac{4C_a(6\sigma)^p}{\Delta_i^p} \tag{A.2}
$$

Combining parts (a), (b), and (c),

$$\mathrm{E}[T_i(T)] \leq 1 + \frac{144 + C_a + 18C_b(M_{q,\sigma}+1)}{\Delta_i^2} + \frac{4C_a(6\sigma)^p}{\Delta_i^p} + \frac{\sigma^2[2\log(T\Delta_i^2)]^{2/p}}{(y_i - x_i)^2}$$

We obtain the following instance-dependent regret that there exists $C'' = C(\sigma, p, q)$ independent of K, T, and Δ_i such that

$$\mathrm{R}(T) \leq C'' \sum_{\Delta_i > 0} \Big(\Delta_i + \frac{1}{\Delta_i} + \frac{1}{\Delta_i^{p-1}} + \frac{\log(T\Delta_i^2)^{2/p}}{\Delta_i} \Big). \tag{A.3}$$

The optimal choice of $\Delta = \sqrt{K/T}(\log K)^{1/p}$ gives the instance independent regret bound $\mathrm{R}(T) \leq \mathcal{O}(\sqrt{KT}(\log K)^{1/p})$. □

A.2 Proof of Lemma A.1.1

Proof. First of all, we will show the following inequality holds for all realizations H_{t-1} of $\mathcal{H}_{t-1}$,

$$\mathrm{P}(A_t = i, E_i^\theta(t), E_i^\mu(t) | H_{t-1}) \leq \frac{1 - p_{i,t}}{p_{i,t}} \cdot \mathrm{P}(A_t = 1, E_i^\theta(t), E_i^\mu(t) | H_{t-1}). \tag{A.4}$$

To prove the above inequality, it suffices to show the following inequality in (A.5). This is because whether $E_i^\mu(t)$ is true or not depends on realizations H_{t-1} of history $\mathcal{H}_{t-1}$ and we would consider realizations H_{t-1} where $E_i^\mu(t)$ is true. If it is not true in some H_{t-1}, then inequality in (A.4) trivially holds.

$$\mathrm{P}(A_t = i | E_i^\theta(t), H_{t-1}) \leq \frac{1 - p_{i,t}}{p_{i,t}} \cdot \mathrm{P}(A_t = 1 | E_i^\theta(t), H_{t-1}) \tag{A.5}$$

Considering realizations H_{t-1} satisfying $E_i^\theta(t) = \{\theta_i(t) \leq y_i\}$, all $\theta_j(t)$ should be smaller than y_i including optimal arm 1 to choose a sub-optimal arm i.

$$\begin{aligned}
&\mathrm{P}(A_t = i | E_i^\theta(t), H_{t-1}) \\
&\leq \mathrm{P}(\theta_j(t) \leq y_i, \forall j \in [K] | E_i^\theta(t), H_{t-1}) \\
&= \mathrm{P}(\theta_1(t) \leq y_i | H_{t-1}) \cdot \mathrm{P}(\theta_j(t) \leq y_i, \forall j \in [K] \setminus \{1, i\} | E_i^\theta(t), H_{t-1}) \\
&= (1 - p_{i,t}) \cdot \mathrm{P}(\theta_j(t) \leq y_i, \forall j \in [K] \setminus \{1, i\} | E_i^\theta(t), H_{t-1})
\end{aligned} \tag{A.6}$$

The first equality above holds since θ_1 is independent of other $\theta_j, \forall j \neq 1$ and events $E_i^\theta(t)$ given $\mathcal{H}_{t-1}$. In the same way it is obtained as below,

$$\begin{aligned}
&\mathrm{P}(A_t = 1 | E_i^\theta(t), H_{t-1}) \\
&\geq \mathrm{P}(\theta_1(t) > y_i \geq \theta_j(t), \forall j \in [K] \setminus \{1\} | E_i^\theta(t), H_{t-1}) \\
&= \mathrm{P}(\theta_1(t) \geq y_i | H_{t-1}) \cdot \mathrm{P}(\theta_j(t) \leq y_i, \forall j \in [K] \setminus \{1, i\} | E_i^\theta(t), H_{t-1}) \\
&= p_{i,t} \cdot \mathrm{P}(\theta_j(t) \leq y_i, \forall j \in [K] \setminus \{1, i\} | E_i^\theta(t), H_{t-1}) \qquad \text{(A.7)}
\end{aligned}$$

Combining two inequalities (A.6) and (A.7), inequality (A.5) is obtained. The rest of proof is as followed.

$$\begin{aligned}
\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^\mu(t), E_i^\theta(t)) &\leq \sum_{t=1}^{T} \mathrm{E}[\mathrm{P}(A_t = i, E_i^\mu(t), E_i^\theta(t) | \mathcal{H}_{t-1})] \\
&\leq \sum_{t=1}^{T} \mathrm{E}\Big[\frac{1 - p_{i,t}}{p_{i,t}} \cdot \mathrm{P}(A_t = 1, E_i^\mu(t), E_i^\theta(t) | \mathcal{H}_{t-1})\Big] \\
&\leq \sum_{t=1}^{T} \mathrm{E}\Big[\mathrm{E}\Big[\frac{1 - p_{i,t}}{p_{i,t}} \cdot \mathrm{I}(A_t = 1, E_i^\mu(t), E_i^\theta(t)) | \mathcal{H}_{t-1}\Big]\Big] \\
&\leq \sum_{t=1}^{T} \mathrm{E}\Big[\frac{1 - p_{i,t}}{p_{i,t}} \cdot \mathrm{I}(A_t = 1, E_i^\mu(t), E_i^\theta(t))\Big] \\
&\leq \sum_{j=0}^{T-1} \mathrm{E}\Big[\frac{1 - p_{i,\delta_j+1}}{p_{i,\delta_j+1}} \sum_{t=\delta_j+1}^{\delta_{j+1}} \mathrm{I}(A_t = 1, E_i^\mu(t), E_i^\theta(t))\Big] \\
&\leq \sum_{j=0}^{T-1} \mathrm{E}\Big[\frac{1 - p_{i,\delta_j+1}}{p_{i,\delta_j+1}}\Big]
\end{aligned}$$

□

A.3 Proof of Theorem 3.2.2

Proof. For each arm $i \neq 1$, we will choose two thresholds $x_i = \mu_i + \frac{\Delta_i}{3}$, $y_i = \mu_1 - \frac{\Delta_i}{3}$ such that $\mu_i < x_i < y_i < \mu_1$ and define three types of events, $E_i^\mu(t) = \{\hat{\mu}_i(t) \leq x_i\}$, $E_i^\theta(t) = \{\theta_i(t) \leq y_i\}$, and $E_{1,i}^\mu(t) = \{\mu_1 - \frac{\Delta_i}{6} - \sqrt{\frac{2 \log T}{T_1(t)}} \leq \hat{\mu}_1(t)\}$. The last event is to control the behavior of $\hat{\mu}_1(t)$ not too far below the mean μ_1. $\mathrm{E}[T_i(T)] = \sum_{t=1}^{T} \mathrm{P}(A_t = i)$

is decomposed into the following four parts according to events $E_i^\mu(t)$, $E_i^\theta(t)$, and $E_{1,i}^\mu(t)$,

$$\begin{aligned}\mathrm{E}[T_i(T)] = &\underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, (E_i^\mu(t))^c)}_{(a)} + \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^\mu(t), (E_i^\theta(t))^c)}_{(b)} \\ &+ \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^\mu(t), E_i^\theta(t), (E_{1,i}^\mu(t))^c)}_{(c)} \\ &+ \underbrace{\sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^\mu(t), E_i^\theta(t), E_{1,i}^\mu(t))}_{(d)}.\end{aligned}$$

Let τ_k denote the time at which k-th trial of arm i happens. Set $\tau_0 = 0$.

$$\begin{aligned}(a) &\le \mathrm{E}[\sum_{k=0}^{T-1} \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathrm{I}(A_t = i)\mathrm{I}((E_i^\mu(t))^c)] \le \mathrm{E}[\sum_{k=0}^{T-1} \mathrm{I}((E_i^\mu(\tau_k+1))^c) \sum_{t=\tau_k+1}^{\tau_{k+1}} \mathrm{I}(A_t = i)] \\ &\le 1 + \sum_{k=1}^{T-1} \mathrm{P}((E_i^\mu(\tau_k+1))^c) \le 1 + \sum_{k=1}^{T-1} \exp(-\frac{k(x_i-\mu_i)^2}{2}) \le 1 + \frac{18}{\Delta_i^2}.\end{aligned}$$

The second last inequality above holds by Hoeffding bound of sample mean of k sub-Gaussian rewards, $\hat{\mu}_i(t)$ in Lemma 3.0.1. The probability in part (b) is upper bounded by 1 if $T_i(t)$ is less than $L_i(T) = \frac{9(2+\epsilon)\log T}{\Delta_i^2}$ and is equal to 0, otherwise. The latter can be proved as below,

$$\begin{aligned}&\mathrm{P}(A_t = i, (E_i^\theta(t))^c | E_i^\mu(t)) \\ &\le \mathrm{P}(\theta_i(t) > y_i | \hat{\mu}_i(t) \le x_i) \\ &\le \mathrm{P}\Big(Z_{it} > \sqrt{\frac{T_i(t)(y_i - x_i)^2}{(2+\epsilon)\log T}} | \hat{\mu}_i(t) \le x_i\Big) = 0 \quad \text{if} \quad T_i(t) \ge L_i(T).\end{aligned}$$

The last equality holds by bounded support of perturbation Z_{it}. Let τ be the largest step until $T_i(t) \leq L_i(T)$, then part (b) is bounded by $L_i(T)$. Regarding part (c),

$$\begin{aligned}
(c) &= \sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^{\mu}(t), E_i^{\theta}(t), (E_{1,i}^{\mu}(t))^c) \\
&\leq \sum_{t=1}^{T} \mathrm{P}((E_{1,i}^{\mu}(t))^c) = \sum_{t=1}^{T}\sum_{s=1}^{T} \mathrm{P}\Big(\mu_1 - \frac{\Delta_i}{6} - \sqrt{\frac{2\log T}{s}} \geq \hat{\mu}_{1,s}\Big) \\
&= \sum_{t=1}^{T}\sum_{s=1}^{T} \mathrm{P}\Big(\mu_1 - \frac{\Delta_i}{6} \geq \hat{\mu}_{1,s} + \sqrt{\frac{2\log T}{s}}\Big) \\
&= \sum_{t=1}^{T} \frac{1}{T} \sum_{s=1}^{T} \exp\big(-\frac{s\Delta_i^2}{72}\big) \leq \frac{72}{\Delta_i^2}
\end{aligned}$$

Define $p_{i,t}$ as the probability $p_{i,t} = \mathrm{P}(\theta_1(t) > y_i | \mathcal{H}_{t-1})$ where $\mathcal{H}_{t-1}$ is defined as the history of plays until time $t-1$. Let δ_j denote the time at which j-th trial of arm 1 happens. In the history where the event $E_{1,i}^{\mu}(t)$ holds, then $\mathrm{P}(\theta_1(t) > y_i | \mathcal{H}_{t-1})$ is strictly greater than zero because of wide enough support of scaled perturbation by adding an extra logarithmic term in T. For $i \neq 1$,

$$\begin{aligned}
(d) &= \sum_{t=1}^{T} \mathrm{P}(A_t = i, E_i^{\mu}(t), E_i^{\theta}(t), E_{1,i}^{\mu}(t)) \leq \sum_{j=0}^{T-1} \mathrm{E}\Big[\frac{1 - p_{i,\delta_j+1}}{p_{i,\delta_j+1}} \mathrm{I}(E_{1,i}^{\mu}(\delta_j + 1))\Big] \\
&= \sum_{j=0}^{T-1} \mathrm{E}\Big[\frac{1 - \mathrm{P}\big(\hat{\mu}_{1,j} + \sqrt{\frac{(2+\epsilon)\log T}{j}} Z \geq \mu_1 - \frac{\Delta_i}{3}\big)}{\mathrm{P}\big(\hat{\mu}_{1,j} + \sqrt{\frac{(2+\epsilon)\log T}{j}} Z \geq \mu_1 - \frac{\Delta_i}{3}\big)} \mathrm{I}\Big(\hat{\mu}_{1,j} \geq \mu_1 - \frac{\Delta_i}{6} - \sqrt{\frac{2\log T}{j}}\Big)\Big] \\
&= \sum_{j=0}^{T-1} \frac{\mathrm{P}\big(Z \leq \sqrt{\frac{2}{2+\epsilon}} - \frac{\sqrt{j}\Delta_i}{6\sqrt{(2+\epsilon)\log T}}\big)}{\mathrm{P}\big(Z \geq \sqrt{\frac{2}{2+\epsilon}} - \frac{\sqrt{j}\Delta_i}{6\sqrt{(2+\epsilon)\log T}}\big)} \quad \because \hat{\mu}_{1,j} = \mu_1 - \frac{\Delta_i}{6} - \sqrt{\frac{2\log T}{j}} \\
&= \sum_{j=0}^{M_i(T)} \frac{\mathrm{P}\big(Z \leq \sqrt{\frac{2}{2+\epsilon}} - \frac{\sqrt{j}\Delta_i}{6\sqrt{(2+\epsilon)\log T}}\big)}{\mathrm{P}\big(Z \geq \sqrt{\frac{2}{2+\epsilon}} - \frac{\sqrt{j}\Delta_i}{6\sqrt{(2+\epsilon)\log T}}\big)} \\
&\leq M_i(T) \cdot \frac{\mathrm{P}\big(Z \leq \sqrt{\frac{2}{2+\epsilon}}\big)}{\mathrm{P}\big(Z \geq \sqrt{\frac{2}{2+\epsilon}}\big)} = M_i(T) \cdot C_{Z,\epsilon} \quad \because \text{maximized when } j = 0
\end{aligned}$$

The first inequality holds by Lemma A.1.1, and the last equality works since the term inside expectation becomes zero if $j \geq M_i(T) = \big(36(\sqrt{2} + \sqrt{(2+\epsilon)})^2 \log T\big)/\Delta_i^2$. This is because the lower bound of perturbed average rewards from the arm 1 becomes larger

than y_i for $j \geq M_i(T)$. Combining parts (a), (b), (c) and (d),

$$\mathrm{E}[T_i(T)] \leq 1 + \frac{90}{\Delta_i^2} + \frac{9(2+\epsilon)\log T}{\Delta_i^2} + C_{Z,\epsilon} \cdot \frac{36(\sqrt{2}+\sqrt{(2+\epsilon)})^2 \log T}{\Delta_i^2}$$

Thus, the instance-dependent regret bound is obtained as below, there exist a universal constant $C''' > 0$ independent of T, K and Δ_i,

$$\mathrm{R}(T) = C''' \sum_{\Delta_i > 0} \Big(\Delta_i + \frac{\log(T)}{\Delta_i} \Big).$$

The optimal choice of $\Delta = \sqrt{K \log T / T}$, the instance-independent regret bound is derived as it follows,

$$\mathrm{R}(T) \leq \mathcal{O}(\sqrt{KT \log T})$$

□

A.4 Proof of Theorem 3.2.3

Proof. The proof is a simple extension of the work of Agrawal and Goyal (2013a). Let $\mu_1 = \Delta = \sqrt{K/T}(\log K)^{1/q}, \mu_2 = \mu_3 = \cdots = \mu_K = 0$ and each reward is generated from a point distribution. Then, sample means of rewards are $\hat{\mu}_1(t) = \Delta$ and $\hat{\mu}_i(t) = 0$ if $i \neq 1$. The normalized $\theta_i(t)$ sampled from the FTPL algorithm (Algorithm 1-(2)) is distributed as $\sqrt{T_i(t)} \cdot (\theta_i(t) - \hat{\mu}_i(t)) \sim Z$.

Define the event $E_{t-1} = \{\sum_{i \neq 1} T_i(t) \leq c\sqrt{KT}(\log K)^{1/q}/\Delta\}$ for a fixed constant c. If E_{t-1} is not true, then the regret until time t is at least $c\sqrt{KT}(\log K)^{1/q}$. For any $t \leq T$, $\mathrm{P}(E_{t-1}) \leq 1/2$. Otherwise, the expected regret until time t, $\mathrm{E}[R(t)] \geq \mathrm{E}[R(t)|E_{t-1}^c] \cdot 1/2 = \Omega(\sqrt{KT}(\log K)^{1/q})$. If E_{t-1} is true, the probability of playing a suboptimal is at least a constant, so that regret is $\Omega(T\Delta) = \Omega(\sqrt{KT}(\log K)^{1/q})$.

$$\begin{aligned}
\mathrm{P}(\exists i \neq 1, \theta_i(t) > \mu_1 | \mathcal{H}_{t-1}) &= \mathrm{P}(\exists i \neq 1, \theta_i(t)\sqrt{T_i(t)} > \Delta\sqrt{T_i(t)} | \mathcal{H}_{t-1}) \\
&= \mathrm{P}(\exists i \neq 1, Z > \Delta\sqrt{T_i(t)} | \mathcal{H}_{t-1}) \\
&\geq 1 - \prod_{i \neq 1} \Big(1 - \exp\big(-(\sqrt{T_i(t)}\Delta/\sigma)^q/2 \big)/C_b \Big)
\end{aligned}$$

Given realization H_{t-1} of history $\mathcal{H}_{t-1}$ such that E_{t-1} is true, we have $\sum_{i \neq 1} T_i(t) \leq$

$\frac{c\sqrt{KT}(\log K)^{1/q}}{\Delta}$ and it is minimized when $T_i(t) = \frac{c\sqrt{KT}(\log K)^{1/q}}{(K-1)\Delta}$ for all $i \neq 1$. Then,

$$\begin{aligned}\mathrm{P}(\exists i \neq 1, \theta_i(t) > \mu_1 | H_{t-1}) &\geq 1 - \prod_{i \neq 1} \Big(1 - \exp\big(-\frac{\big(\sqrt{T_i(t)}\Delta\big)^q}{2\sigma^q}\big)/C_b\Big) \\ &= 1 - \Big(1 - \frac{\sigma(q,K)}{K}\Big)^{K-1}\end{aligned}$$

where $\sigma(q,K) = \exp\big(\frac{c^{q/2}}{2\nu^q}(\frac{K}{K-1})^{q/2}\big)/C_b$. Accordingly,

$$\mathrm{P}(\exists i \neq 1, A_t = i) \geq \frac{1}{2}\Big(1 - \big(1 - \frac{\sigma(q,K)}{K}\big)^{K-1}\Big) \cdot \frac{1}{2} \to p^\star \in (0,1).$$

Therefore, the regret in time T is at least $Tp^\star\Delta = \Omega(\sqrt{KT}(\log K)^{1/q})$. □

APPENDIX B

Detailed Proofs for Adversarial Multi-armed Bandits

In this section, the proofs omitted in Chapter 4 are presented.

B.1 Proof of Theorem 4.2.2

Proof. Fix $\eta = 1$ without loss of generality in FTRL algorithm via Tsallis entropy. For any $\alpha \in (0,1)$, Tsallis entropy yields the following choice probability, $C_i(\mathbf{G}) = \left(\frac{1-\alpha}{\alpha}\right)^{\frac{1}{\alpha-1}}(\lambda(\mathbf{G}) - G_i)^{\frac{1}{\alpha-1}}$, where $\sum_{i=1}^{K} C_i(\mathbf{G}) = 1,\ \lambda(\mathbf{G}) \geq \max_{i\in[K]}\ G_i$. Then for $1 \leq i \neq j \leq K$, the first derivative is negative as shown below,

$$\frac{\partial C_i(\mathbf{G})}{\partial G_j} = \left(\frac{1-\alpha}{\alpha}\right)^{\frac{1}{\alpha-1}} \frac{1}{\alpha-1} \frac{(\lambda(\mathbf{G}) - G_i)^{\frac{1}{\alpha-1}-1}(\lambda(\mathbf{G}) - G_j)^{\frac{1}{\alpha-1}-1}}{\sum_{l=1}^{K}(\lambda(\mathbf{G}) - G_l)^{\frac{1}{\alpha-1}-1}} < 0.$$

and it implies that $\mathcal{D}\mathbf{C}(\mathbf{G})$ is symmetric. For, $1 \leq i \neq j \neq k \leq K$, the second partial derivative, $\frac{\partial^2 C_i(\mathbf{G})}{\partial G_j \partial G_k}$ is derived as

$$C_i(\mathbf{G}) \cdot \Bigg(\Big(\sum_{l=1}^{K} (\lambda(\mathbf{G}) - G_l)^{\frac{2-\alpha}{\alpha-1}}\Big)\Big(\frac{1}{\lambda(\mathbf{G}) - G_i} + \frac{1}{\lambda(\mathbf{G}) - G_j} + \frac{1}{\lambda(\mathbf{G}) - G_k}\Big) - \sum_{l=1}^{K} (\lambda(\mathbf{G}) - G_l)^{\frac{3-2\alpha}{\alpha-1}} \Bigg).$$

If we set $1/(\lambda(\mathbf{G}) - G_i) = X_i$, the term except for the term $C_i(\mathbf{G})$ is simplified to $\sum_{i=1}^{K} X_i^{\frac{2-\alpha}{1-\alpha}} \cdot (X_1 + X_2 + X_3) - \sum_{i=1}^{K} X_i^{\frac{3-2\alpha}{1-\alpha}}$ where $\sum_{i=1}^{K} X_i^{\frac{1}{1-\alpha}} = \left(\frac{\alpha}{1-\alpha}\right)^{\frac{1}{\alpha-1}}$. If we set $K = 4$, $C(\mathbf{G}) = (\epsilon, \epsilon, \epsilon, 1-3\epsilon)$, and $X_i = C_i^{1-\alpha}\frac{1-\alpha}{\alpha}$, then it is equal to

$$\left(\frac{1-\alpha}{\alpha}\right)^{\frac{3-2\alpha}{1-\alpha}} \left(6\epsilon^{3-2\alpha} + 3\epsilon^{1-\alpha}(1-3\epsilon)^{2-\alpha} - (1-3\epsilon)^{3-2\alpha}\right). \tag{B.1}$$

So, there always exists $\epsilon > 0$ small enough to make the value of (B.1) negative where $\alpha \in (0, 1)$, which means condition (4) in Theorem 4.2.1 is violated. □

B.2 Asymptotic expected block maxima and supremum of hazard rate

B.2.1 Extreme value theory

Theorem B.2.1 (Proposition 0.3 (Resnick, 2013)). *Suppose that there exist* $\{a_K > 0\}$ *and* $\{b_K\}$ *such that*

$$\mathrm{P}((M_K - b_K)/a_K \leq z) = F^K(a_K \cdot z + b_K) \to G(z) \quad \text{as } K \to \infty \tag{B.2}$$

where G *is a non-degenerate distribution function, then* G *belongs to one of families; Gumbel, Fréchet and Weibull. Then,* F *is in the domain of attraction of* G*, written as* $F \in D(G)$*.*
1. Gumbel type (Γ*) with* $G(x) = \exp(-\exp(-x))$ *for* $x \in \mathbb{R}$*.*
2. Fréchet type (Φ_α*) with* $G(x) = 0$ *for* $x > 0$ *and* $G(x) = \exp(-x^{-\alpha})$ *for* $x \geq 0$*.*
3. Weibull type (Ψ_α*) with* $G(x) = \exp(-(-x)^\alpha)$ *for* $x \leq 0$ *and* $G(x) = 1$ *for* $x > 0$*.*

Let $\gamma_K = F^{\leftarrow}(1 - 1/K) = \inf\{x : F(x) \geq 1 - 1/K\}$.

Theorem B.2.2 (Proposition 1.1 (Resnick, 2013)). *Type 1 - Gumbel (*Λ*)*
1. If $F \in D(\Gamma)$*, there exists some strictly positive function* $g(t)$ *s.t.* $\lim_{t\to\omega_F} \frac{1-F(t+x\cdot g(t))}{1-F(t)} = \exp(-x)$ *for all* $x \in \mathbb{R}$ *with exponential tail decay. Its corresponding normalizing sequences are* $a_K = g(\gamma_K)$ *and* $b_K = \gamma_K$*, where* $g = (1 - F)/F'$*.*
2. If $\lim_{x\to\infty} \frac{F''(x)(1-F(x))}{\{F'(x)\}^2} = -1$*, then* $F \in D(\Lambda)$*.*
3. If $\int_{-\infty}^0 |x|F(dx) < \infty$*, then* $\lim_{K\to\infty} \mathrm{E}\big((M_K - b_K)/a_K\big) = -\Gamma^{(1)}(1)$*. Accordingly,* $\mathrm{E}M_K$ *behaves as* $-\Gamma^{(1)}(1) \cdot g(\gamma_K) + \gamma_K$*.*

Theorem B.2.3 (Proposition 1.11 (Resnick, 2013)). *Type 2 - Fréchet (*Φ_α*)*
1. If $F \in D(\Phi_\alpha)$*, its upper end point is infinite,* $\omega_F = \infty$*, and it has tail behavior that decays polynomially* $\lim_{t\to\infty} \frac{1-F(tx)}{1-F(t)} = x^{-\alpha}$*, for* $x > 0, \alpha > 0$*. Its corresponding normalizing sequences are* $a_K = \gamma_K$ *and* $b_K = 0$*.*
2. If $\lim_{x\to\infty} \frac{xF'(x)}{1-F(x)} = \alpha$ *for some* $\alpha > 0$*, then* $F \in D(\Phi_\alpha)$*.*
3. If $\alpha > 1$ *and* $\int_{-\infty}^0 |x|F(dx) < \infty$*, then* $\lim_{K\to\infty} \mathrm{E}\big(M_K/a_K\big) = \Gamma\big(1 - 1/\alpha\big)$*. Accordingly,* $\mathrm{E}M_K$ *behaves as* $\Gamma\big(1 - \frac{1}{\alpha}\big) \cdot \gamma_K$*.*

B.2.2 Gumbel distribution

Gumbel has the following distribution function, the first derivative and the second derivative, $F(x) = \exp(-e^{-x})$, $F'(x) = e^{-x}F(x)$, and $F''(x) = (e^{-x} - 1)F'(x)$. Then we have

$$\lim_{x\to\infty} \frac{F''(x)(1 - F(x))}{\{F'(x)\}^2} = -1,$$

thus this is Gumbel-type distribution by Theorem B.2.2, $F \in D(\Lambda)$.

If $g(x) = e^x(e^{e^{-x}} - 1)$, then normalizing constants are obtained as

$$b_K = -\log(-\log(1 - 1/K)) \sim \log K,$$

$$a_K = g(b_K) = (1 - F(b_K))/(\exp(-b_K)F(b_K)) = 1 + 1/K + o(\frac{1}{K}).$$

Accordingly, $\mathrm{E}M_K = -\Gamma^{(1)}(1) \cdot (1 + \frac{1}{K}) + \log K + o(1/K)$.

Its hazard rate is derived as $h(x) = \frac{F'(x)}{1-F(x)} = \frac{e^{-x}}{\exp(e^{-x})-1}$, and since it increases monotonically and converges to 1 as x goes to infinity, it has an asymptotically tight bound 1.

B.2.3 Gamma distribution

For $x > 0$, the first derivative and the second derivative of distribution function are given as $F'(x) = (x^{\alpha-1}e^{-x})/\Gamma(\alpha)$ and $F''(x) = -F'(x)(1 + (\alpha - 1)/t) \sim -F'(x)$. It satisfies $\frac{F''(1-F(x))}{(F'(x))^2} \sim -\frac{1-F(x)}{F'(x)} \to -1$ so it is Gumbel-type by Theorem B.2.2, $F \in D(\Lambda)$. It has $g(x) \to 1$ and thus $a_K = 1$. Since $F'(b_K) \sim 1 - F(b_K) = 1/K$, then we have

$$(\alpha - 1)\log b_K - b_K - \log\Gamma(\alpha) = -\log K.$$

Thus, we have $b_K = \log K + (\alpha - 1)\log\log K - \log\Gamma(\alpha)$. Accordingly,

$$\mathrm{E}M_K = -\Gamma^{(1)}(1) + \log K + (\alpha - 1)\log\log K - \log\Gamma(\alpha).$$

Its hazard function is expressed by $h(x) = (x^{\alpha-1}\exp(-x))/[\int_x^\infty t^{\alpha-1}\exp(-t)dt]$. It increases monotonically and converges to 1, and thus has an asymptotically tight bound 1.

B.2.4 Weibull distribution

The Weibull distribution function and its first derivative are obtained as as $F(x) = 1 - \exp(-(x + 1)^\alpha + 1)$ and $F'(x) = \alpha(x + 1)^{\alpha-1}(1 - F(x))$. Its second derivative is $(\frac{\alpha-1}{x+1} - \alpha(x + 1)^{\alpha-1}) \cdot F'(x)$. The second condition in Theorem B.2.2 is satisfied, and thus $F \in$

$D(\Lambda)$ and $g(x) = x^{-\alpha+1}/\alpha$. Corresponding normalizing constants are derived as $b_K = (1+\log K)^{1/\alpha} - 1 \sim (\log K)^{1/\alpha}$ and $a_K = g(b_K) = (\log K)^{1/\alpha-1}/\alpha$. So,

$$\mathrm{E}M_K = -\Gamma^{(1)}(1) \cdot (\log K)^{1/\alpha-1}/\alpha + (\log K)^{1/\alpha}.$$

Its hazard rate function is $h(x) = \alpha(x+1)^{\alpha-1}$ for $x \geq 0$. If $\alpha > 1$, it increases monotonically and becomes unbounded. If the case for $\alpha \leq 1$ is only considered, then the hazard rate is tightly bounded by α.

B.2.5 Fréchet distribution

The first derivative of Fréchet distribution function is $F'(x) = \exp(-x^{-\alpha})\alpha x^{-\alpha-1}$ for $x > 0$ and the second condition in Theorem B.2.3 is satisfied as

$$\lim_{x\to\infty} \frac{xF'(x)}{1-F(x)} = \lim_{x\to\infty} \frac{\alpha x^{-\alpha}}{\exp(x^{-\alpha})-1} \to \alpha.$$

Thus, it is Fréchet-type distribution (Φ_α) so that $b_K = 0$ and $a_K = [-\log(1-1/K)]^{-1/\alpha} = [1/K + o(1/K)]^{-1/\alpha} \sim K^{1/\alpha}$. So, $\mathrm{E}M_K = \Gamma(1-1/\alpha) \cdot K^{1/\alpha}$.

The hazard rate is $h(x) = \alpha x^{-\alpha-1} \frac{1}{\exp(x^{-\alpha})-1}$. It is already known that supremum of hazard is upper bound by 2α in Appendix D.2.1 in Abernethy et al. (2015). Regarding the lower bound of a hazard rate, $\sup_{x>0} h(x) \geq h(1) = \alpha/(e-1)$.

B.2.6 Pareto distribution

The modified Pareto distribution function is $F(x) = 1 - \frac{1}{(1+x)^\alpha}$ for $x \geq 0$. The second condition in Theorem B.2.3 is met as $\lim_{x\to\infty} \frac{xF'(x)}{1-F(x)} = \lim_{x\to\infty} \frac{\alpha x}{1+x} \to \alpha > 1$. Thus, it is Fréchet-type distribution (Φ_α), and has normalizing constants, $b_K = 0$ and $a_K = K^{1/\alpha} - 1$. Accordingly, $\mathrm{E}M_K \approx \Gamma(1-1/\alpha) \cdot (K^{1/\alpha} - 1)$.

Its hazard rate is $h(x) = \frac{\alpha}{1+x}$ for $x \geq 0$ so that it is tightly bounded by α.

B.3 Two-armed bandit setting

In this section we consider two-armed bandit setting and study both Shannon entropy and Tsallis entropy in FTRL algorithm.

B.3.1 Shannon entropy

There is a mapping between $\mathcal{R}$ and $F_{\mathcal{D}^\star}$,

$$\mathcal{R}(w) - \mathcal{R}(0) = -\int_0^w F_{\mathcal{D}^\star}^{-1}(1-z)dz. \tag{B.3}$$

Let $\mathcal{R}(w)$ be one-dimensional Shannon entropic regularizer, $\mathcal{R}(w) = -w\log w - (1-w)\log(1-w)$ for $w \in (0,1)$ and its first derivative is $\mathcal{R}'(w) = \log\frac{1-w}{w} = F_{\mathcal{D}^\star}^{-1}(1-w)$. Then $F_{\mathcal{D}^\star}(z) = \frac{\exp(z)}{1+\exp(z)}$, which can be interpreted as the difference of two Gumbel distribution as follows,

$$\begin{aligned}
&\mathrm{P}(\arg\max_{w\in\Delta_1}\langle w, (G_1+Z_1, G_2, +Z_2)\rangle = 1) \\
&= \mathrm{P}(G_1+Z_1 > G_2+Z_2)) \\
&= \mathrm{P}(Y > G_2 - G_1) \text{ where } Y = Z_1 - Z_2 \sim \mathcal{D}^\star \\
&= 1 - F_{\mathcal{D}^\star}(G_2 - G_1) \\
&= \frac{\exp(G_1)}{\exp(G_1)+\exp(G_2)}.
\end{aligned}$$

If $Z_1, Z_2 \sim Gumbel(\alpha, \beta)$ and are independent, then $Z_1 - Z_2 \sim Logistic(0, \beta)$. Therefore, the perturbation, $F_{\mathcal{D}^\star}$ is not distribution function for Gumbel, but Logistic distribution which is the difference of two i.i.d Gumbel distributions. Interestingly, the logistic distribution turned out to be also Gumbel types extreme value distribution as Gumbel distribution. It is naturally conjectured that the difference between two i.i.d Gumbel types distribution with exponential tail decay must be Gumbel types as well. The same holds for Fréchet-type distribution with polynomial tail decay.

B.3.2 Tsallis entropy

Theorem 4.2.2 states that there does not exist a perturbation that gives the choice probability function same as that from FTRL via Tsallis entropy when $K \geq 4$. In two-armed setting, however, there exists a perturbation equivalent to Tsallis entropy and this perturbation naturally yields an optimal perturbation based algorithm.

Let us consider Tsallis entropy regularizer in one dimensional decision set expressed by $R(w) = \frac{1}{1-\alpha}(-1 + w^\alpha + (1-w)^\alpha)$ for $w \in (0,1)$ and its first derivative is $R'(w) = \frac{\alpha}{1-\alpha}(w^{\alpha-1} - (1-w)^{\alpha-1}) = F_{\mathcal{D}^\star}^{-1}(1-w)$. If we set $u = 1-w$, then the implicit form of distribution function and density function are given as $F_{\mathcal{D}^\star}(\frac{\alpha}{1-\alpha}((1-u)^{\alpha-1} - u^{\alpha-1})) = u$ and $f_{\mathcal{D}^\star}(\frac{\alpha}{1-\alpha}((1-u)^{\alpha-1} - u^{\alpha-1})) = \frac{1}{\alpha((1-u)^{\alpha-2}+u^{\alpha-2})}$. As u converges to 1, then $z =$

$\frac{\alpha}{1-\alpha}((1-u)^{\alpha-1} - u^{\alpha-1})$ goes to positive infinity. This distribution satisfies the second condition in Theorem B.2.3 so that it turns out to be Fréchet-type.

$$\lim_{z\to\infty} \frac{z f_{\mathcal{D}^\star}(z)}{1 - F_{\mathcal{D}^\star}(z)} = \lim_{u\to 1} \frac{\frac{\alpha}{1-\alpha}((1-u)^{\alpha-1} - u^{\alpha-1})}{(1-u) \times \alpha((1-u)^{\alpha-2} + u^{\alpha-2})} = \frac{1}{1-\alpha}.$$

If the conjecture above holds, the optimal perturbation that corresponds to Tsallis entropy regularizer must be also Fréchet-type distribution in two armed bandit setting. This result strongly support our conjecture that the perturbation in an optimal FTPL algorithm must be Fréchet-type.

APPENDIX C

Detailed Proofs for Non-stationary Stochastic Linear Bandit

In this section, the proofs omitted in Chapter 6 are presented.

C.1 Useful Lemma

Lemma C.1.1 (Concentration and anti-concentration of Gaussian distribution (Abramowitz and Stegun, 1964))**.** *Let Z be the Gaussian random variable with mean μ and variance σ^2. For any $z > 0$,*

$$\frac{1}{4\sqrt{\pi}} \exp(-\frac{7z^2}{2}) \leq P(|Z - \mu| > z\sigma) \leq \frac{1}{2} \exp(-\frac{z^2}{2}).$$

C.2 Proof of Theorem 6.4.2

Proof of Theorem 6.4.2. The dynamic regret bound is decomposed into two terms, (A) expected surrogate regret and (B) bias arising from variation on true parameter.

$$\begin{aligned} E[R(T)] &= \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \theta_t^\star \rangle] = \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \theta_t^\star - \bar{\theta}_t \rangle] \\ &\leq \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + 2 \sum_{t=1}^{T} \|\theta_t^\star - \bar{\theta}_t\|_2 = (A) + (B) \end{aligned}$$

The expected surrogate regret term (A) is bounded as,

$$
\begin{aligned}
(A) &= \sum_{t=1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] \leq \sum_{t=d+1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle] + d \\
&\leq \sum_{t=d+1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle I\{E^{wls}\}] + T \cdot P(\bar{E}^{wls}) + d \\
&\leq \sum_{t=d+1}^{T} E[\langle x_t^\star - X_t, \bar{\theta}_t \rangle I\{E^{wls}\}] + T p_1 + d \\
&\leq (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big) E_t\big[\sum_{t=d+1}^{T} \min(1, \|X_t\|_{V_t^{-1}}) \big] + T(p_1 + p_2) + d \\
&\because \text{Theorem } 6.4.1 \\
&\leq (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big) \sqrt{c_3 T} + T(p_1 + p_2) + d \quad \because \text{C-S ineq. \& Lemma } C.2.1
\end{aligned}
$$

Lemma C.2.1 (Corollary 4, Russac et al. (2019))**.** *For any* $\lambda > 0$*,*

$$
\sum_{t=d+1}^{T} \min(1, \|X_t\|^2_{V_t^{-1}}) \leq c_3 T
$$

where $c_3 = 2d \log(1/\gamma) + 2\frac{d}{T} \log(1 + \frac{1}{d\lambda(1-\gamma)})$.

The bias term (B) is bounded in terms of total variation, B_T. We first bound the individual bias term at time t. For any integer $D > 0$,

$$
\begin{aligned}
\|\theta_t^\star - \bar{\theta}_t\|_2 &= \|W_{t,\lambda}^{-1} \sum_{l=1}^{t-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 \\
&\leq \|W_{t,\lambda}^{-1} \sum_{l=t-D}^{t-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 + \|W_{t,\lambda}^{-1} \sum_{l=1}^{t-D-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 \\
&\leq \|W_{t,\lambda}^{-1} \sum_{l=t-D}^{t-1} \gamma^{-l} X_l X_l^T \sum_{m=l}^{t-1} (\theta_m^\star - \theta_{m+1}^\star)\|_2 + \| \sum_{l=1}^{t-D-1} \gamma^{-l} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_{W_{t,\lambda}^{-2}} \\
&\leq \|W_{t,\lambda}^{-1} \sum_{m=t-D}^{t-1} \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T (\theta_m^\star - \theta_{m+1}^\star)\|_2 + \frac{1}{\lambda} \| \sum_{l=1}^{t-D-1} \gamma^{t-l-1} X_l X_l^T (\theta_l^\star - \theta_t^\star)\|_2 \\
&\leq \sum_{m=t-D}^{t-1} \|W_{t,\lambda}^{-1} \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T (\theta_m^\star - \theta_{m+1}^\star)\|_2 + \frac{2}{\lambda} \frac{\gamma^D}{1-\gamma}
\end{aligned}
$$

$$\leq \sum_{m=t-D}^{t-1} \lambda_{\max}\Big(W_{t,\lambda}^{-1} \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T\Big) \|\theta_m^\star - \theta_{m+1}^\star\|_2 + \frac{2}{\lambda}\frac{\gamma^D}{1-\gamma}$$
$$\leq \sqrt{\frac{dD}{\lambda}} \sum_{m=t-D}^{t-1} \|\theta_m^\star - \theta_{m+1}^\star\|_2 + \frac{2}{\lambda}\frac{\gamma^D}{1-\gamma}$$

The third inequality holds due to $W_{t,\lambda}^{-2} \preccurlyeq (\frac{\gamma^{t-1}}{\lambda})^2 I_d$. The last inequality works by Lemma C.2.2. By combining individual bias terms over T rounds, we can derive the upper bound of bias term (B) as,

$$\begin{aligned}(B) &= 2\sum_{t=1}^{T} \|\theta_t^\star - \bar{\theta}_t\|_2 \\ &\leq 2\sum_{t=1}^{T} \sqrt{\frac{dD}{\lambda}} \sum_{m=t-D}^{t-1} \|\theta_m^\star - \theta_{m+1}^\star\|_2 + \frac{4}{\lambda}\frac{\gamma^D}{1-\gamma} T \\ &\leq 2\sqrt{\frac{d}{\lambda}} D^{3/2} B_T + \frac{4}{\lambda}\frac{\gamma^D}{1-\gamma} T\end{aligned}$$

Lemma C.2.2. *For $t - D \leq m \leq t - 1$,*

$$\lambda_{\max}\Big(W_{t,\lambda}^{-1} \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T\Big) \leq \sqrt{\frac{dD}{\lambda}}.$$

Proof. Denote by $\mathbb{B}(1) = \{x | \|x\|_2 = 1\}$ the unit ball.

$$\begin{aligned}\lambda_{\max}\Big(W_{t,\lambda}^{-1} \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T\Big) &= \sup_{z\in\mathbb{B}(1)} \Big| z^T W_{t,\lambda}^{-1} \big(\sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T \big) z \Big| \\ &= \Big| z_\star^T W_{t,\lambda}^{-1} \big(\sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T \big) z_\star \Big| \quad z_\star : \text{optimizer} \\ &\leq \|z_\star\|_{W_{t,\lambda}^{-1}} \Big\| \sum_{l=t-D}^{m} \gamma^{-l} X_l X_l^T z_\star \Big\|_{W_{t,\lambda}^{-1}} \\ &\leq \|z_\star\|_{W_{t,\lambda}^{-1}} \Big\| \sum_{l=t-D}^{m} \gamma^{-l} X_l \|X_l\|_2 \|z_\star\|_2 \Big\|_{W_{t,\lambda}^{-1}} \\ &\leq \frac{\gamma^{(t-1)/2}}{\sqrt{\lambda}} \Big\| \sum_{l=t-D}^{m} \gamma^{-l} X_l \Big\|_{W_{t,\lambda}^{-1}} \leq \frac{\gamma^{(t-1)/2}}{\sqrt{\lambda}} \sum_{l=t-D}^{m} \Big\| \gamma^{-l} X_l \Big\|_{W_{t,\lambda}^{-1}} \\ &\leq \sqrt{\frac{D}{\lambda}} \sqrt{\gamma^{t-1} \sum_{l=t-D}^{m} \|\gamma^{-l} X_l\|_{W_{t,\lambda}^{-1}}^2} \leq \sqrt{\frac{dD}{\lambda}}\end{aligned}$$

In this proof, we utilized the fact that for any x, we have $\|x\|_{W_{t,\lambda}^{-1}} \leq \|x\|_2/\sqrt{\lambda\gamma^{-(t-1)}} = \frac{\|x\|_2\gamma^{(t-1)/2}}{\sqrt{\lambda}}$. The last step makes use of the following result: for any $m \in \{t-D, \cdots, t-1\}$,

$$
\begin{aligned}
&\gamma^{t-1}\sum_{l=t-D}^{m}\|\gamma^{-l}X_l\|^2_{W_{t,\lambda}^{-1}} \\
&= \sum_{l=t-D}^{m}\text{tr}(\gamma^{-l}X_l^T W_{t,\lambda}^{-1}X_l) \\
&= \text{tr}\Big(W_{t,\lambda}^{-1}\sum_{l=t-D}^{m}\gamma^{-l}X_l X_l^T\Big) \\
&\leq \text{tr}\Big(W_{t,\lambda}^{-1}\sum_{l=t-D}^{m}\gamma^{-l}X_l X_l^T\Big) + \sum_{s=m+1}^{t-1}\gamma^{-l}X_l^T W_{t,\lambda}^{-1}X_l + \lambda\gamma^{-t}\sum_{i=1}^{d}e_i^T W_{t,\lambda}^{-1}e_i \\
&= \text{tr}\Big(W_{t,\lambda}^{-1}\sum_{l=t-D}^{m}\gamma^{-l}X_l X_l^T\Big) + \text{tr}\Big(W_{t,\lambda}^{-1}\sum_{l=m+1}^{t-1}\gamma^{-l}X_l X_l^T\Big) + \text{tr}\Big(W_{t,\lambda}^{-1}\lambda\gamma^{-t}\sum_{i=1}^{d}e_i e_i^T\Big) \\
&= \text{tr}(I_d) = d
\end{aligned}
$$

□

Therefore, the expected dynamic regret is bounded as,

$$
\begin{aligned}
E[R(T)] &\leq (A) + (B) \\
&\leq (c_1+c_2)\big(1+\frac{2}{p_3-p_2}\big)\sqrt{c_3 T} + T(p_1+p_2) + d + 2\sqrt{\frac{d}{\lambda}}D^{3/2}B_T + \frac{4}{\lambda}\frac{\gamma^D}{1-\gamma}T
\end{aligned}
$$

In Corollary 6.4.1, the choices of a, c_1, c_2, and c_3 are

$$
\begin{aligned}
&a^2 = 14c_1^2,\ c_1 = \sqrt{2\log T + d\log(1+\frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})} + \lambda^{1/2} \\
&c_2 = a\sqrt{2\log(T/2)}, \text{and } c_3 = 2d\log(1/\gamma) + 2\frac{d}{T}\log(1+\frac{1}{d\lambda(1-\gamma)}).
\end{aligned}
$$

With optimal choice of

$$
D = \frac{\log T}{1-\gamma},\ \gamma = 1 - d^{-\frac{1}{4}}B_T^{\frac{1}{2}}T^{-\frac{1}{2}},
$$

the dynamic regret of the D-RandLinUCB algorithm is asymptotically upper bounded by $\mathcal{O}(d^{\frac{7}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$ as $T \to \infty$ as $T \to \infty$.

In Corollary 6.4.2, the choices of a, c_1, c_2, and c_3 are

$$a^2 = 14c_1^2,\ c_1 = \sqrt{2\log T + d\log(1 + \frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})} + \lambda^{1/2}$$
$$c_2 = a\sqrt{2\log(KT/2)}, \text{and } c_3 = 2d\log(1/\gamma) + 2\frac{d}{T}\log(1 + \frac{1}{d\lambda(1-\gamma)}).$$

With optimal choice of

$$D = \frac{\log T}{1-\gamma},\ \gamma = 1 - d^{-\frac{1}{4}}(\log K)^{-\frac{1}{4}}B_T^{\frac{1}{2}}T^{-\frac{1}{2}},$$

the dynamic regret of the D-LinTS algorithm is asymptotically upper bounded by $\mathcal{O}(d^{\frac{7}{8}}(\log K)^{\frac{3}{8}}B_T^{\frac{1}{4}}T^{\frac{3}{4}})$ as $T \to \infty$. □

C.3 Adapting to unknown non-stationarity

The optimal discounting factor $\gamma^\star$ requires prior information of non-stationarity measure B_T, which is unavailable in general. We make up for the lack of this information via running the EXP3 algorithm as a meta algorithm to adaptively choose the optimal discounting factor. This method of adapting to unknown non-stationarity is called as Bandits-over-Bandits (BOB) (Cheung et al., 2019).

The BOB mechanism divides the entire time horizon into $\lceil T/H \rceil$ blocks of equal length H rounds, and specifies a set $J \subset [H]$ from which each critical window size D_i is drawn from. For each block i, the BOB mechanism selects a critical window size D_i and starts a new copy of D-RandLinUCB algorithm. On top of this, the BOB mechanism separately maintains the EXP3 algorithm to carefully control the selection of critical window size for each block, and the total reward of each block is used as bandit feedback for the EXP3 algorithm.

We set $H = d^{\frac{1}{4}}T^{\frac{1}{2}}$ and we consider D-RandLinUCB algorithm together with BOB mechanism. The details for D-LinTS algorithm will be skipped since its dynamic regret bound can be obtained in a very similar fashion. With choices of parameters,

$$a^2 = 14c_1^2,\ c_1 = \sqrt{2\log T + d\log(1 + \frac{1-\gamma^{2(T-1)}}{\lambda d(1-\gamma^2)})} + \lambda^{1/2},$$
$$c_2 = a\sqrt{2\log(T/2)}, \text{and } c_3 = 2d\log(1/\gamma) + 2\frac{d}{T}\log(1 + \frac{1}{d\lambda(1-\gamma)})$$

the expected dynamic regret bound is bounded as

$$E[R(T)] \leq (c_1 + c_2)\big(1 + \frac{2}{p_3 - p_2}\big)\sqrt{c_3 T} + T(p_1 + p_2) + d + 2\sqrt{\frac{d}{\lambda}} D^{3/2} B_T + \frac{4}{\lambda}\frac{\gamma^D}{1-\gamma} T \tag{C.1}$$

$$\leq \tilde{\mathcal{O}}\big(\sqrt{d} D^{3/2} B_T + \frac{dT}{\sqrt{D}}\big), \tag{C.2}$$

where D is called *critical window size* in the sense that the observations outside this critical window size would not affect the order of regret bound (only by constant instead). This quantity is closely related to discounting factor γ in the following equation, $D = (\log T)/(1-\gamma)$. That is, to find the optimal discounting factor is equivalent to finding the optimal critical window size.

$$\begin{aligned}
\mathbb{E}[Regret_T(BOB)] =& \mathbb{E}[\sum_{t=1}^{T} \langle x_t^\star, \theta_t \rangle - \sum_{t=1}^{T} \langle X_t, \theta_t \rangle] \\
=& \underbrace{\mathbb{E}\big[\sum_{t=1}^{T} \langle x_t^\star, \theta_t \rangle - \sum_{i=1}^{[T/H]} \sum_{t=(i-1)H+1}^{iH \wedge T} \langle X_t(D_\dagger), \theta_t \rangle\big]}_{(a)} \\
&+ \underbrace{\mathbb{E}\big[\sum_{i=1}^{[T/H]} \sum_{t=(i-1)H+1}^{iH \wedge T} \langle X_t(D_\dagger) - X_t(D_i), \theta_t \rangle\big]}_{(b)}
\end{aligned}$$

where $D_\dagger$ is the best critical window size to approximate the optimal critical window size $D^\star$ in the pool $J = \{H^0, [H^{\frac{1}{\Delta}}], [H^{\frac{2}{\Delta}}], \cdots, H\}$ for some positive integer Δ, and we can set $H = [d^{\frac{1}{4}} T^{\frac{1}{2}}]$ and $\Delta = [\log H]$. Recall that $D^\star = d^{\frac{1}{4}} B_T^{-\frac{1}{2}} T^{\frac{1}{2}} \log T$. It suffices to bound terms (a) and (b).

The term (a) is bounded using Equation C.2,

$$\begin{aligned}
(a) &= \mathbb{E}\big[\sum_{i=1}^{[T/H]} \sum_{t=(i-1)H+1}^{iH \wedge T} \langle x_t^\star - X_t(D_\dagger), \theta_t \rangle\big] \\
&= \sum_{i=1}^{[T/H]} \tilde{\mathcal{O}}\big(\sqrt{d} D_\dagger^{3/2} B_T(i) + \frac{dH}{\sqrt{D_\dagger}}\big) \\
&= \tilde{\mathcal{O}}\big(\sqrt{d} D_\dagger^{3/2} B_T + \frac{dT}{\sqrt{D_\dagger}}\big)
\end{aligned}$$

where $B_T(i) = \sum_{t=(i-1)H+1}^{iH \wedge T-1} \|\theta_t - \theta_{t+1}\|$ is the total variation in block i.

Next, we bound the term (b) as below. The number of rounds in a block is $[T/H]$ and the number of possible options of D_i is $|J| = \Delta + 1$.

$$(b) \leq \tilde{\mathcal{O}}(\sqrt{H|J|T})$$

where this inequality follows by the same argument as in the sliding window based approach (Cheung et al., 2019).

Combining term (a) and (b), the regret of the BOB mechanism is

$$\mathbb{E}[Regret_T(BOB)] = \tilde{\mathcal{O}}\big(\sqrt{d}{D_\dagger}^{3/2} B_T + \frac{dT}{\sqrt{D_\dagger}} + \sqrt{H|J|T}\big).$$

Case 1 : $D^\star \leq H$**.** This condition implies that there exists ϵ_1 and ϵ_2 such that $B_T = \tilde{\Omega}(d^{\epsilon_1} T^{\epsilon_2})$ where at least one of ϵ_1 and ϵ_2 is positive, and thus $D_\dagger$ can automatically approximate to the nearly optimal critical window size $D^\star$. Then, the dynamic regret of the BOB mechanism becomes

$$\begin{aligned}
\mathbb{E}[Regret_T(BOB)] &= \tilde{\mathcal{O}}\big(\sqrt{d}{D_\dagger}^{3/2} B_T + \frac{dT}{\sqrt{D_\dagger}} + \sqrt{H|J|T}\big) \\
&= \tilde{\mathcal{O}}\big(\sqrt{d}D^{\star 3/2} H^{\frac{1}{\Delta}} B_T + \frac{dT}{\sqrt{D^\star H^{-\frac{1}{\Delta}}}} + d^{\frac{1}{8}} T^{\frac{3}{4}} \Delta^{\frac{1}{2}}\big) \\
&= \tilde{\mathcal{O}}\big(d^{\frac{7}{8}} B_T^{\frac{1}{4}} T^{\frac{3}{4}} + d^{\frac{1}{8}} T^{\frac{3}{4}} \Delta^{\frac{1}{2}}\big) \\
&= \tilde{\mathcal{O}}\big(d^{\frac{7}{8}} B_T^{\frac{1}{4}} T^{\frac{3}{4}}\big).
\end{aligned}$$

Case 2 : $D^\star > H$**.** This condition implies that $B_T = \tilde{\mathcal{O}}(1)$. Under this situation, $D_\dagger$ equals to H, which is the critical window size closest to $D^\star$, then the dynamic regret of the BOB mechanism becomes

$$\begin{aligned}
\mathbb{E}[Regret_T(BOB)] &= \tilde{\mathcal{O}}\big(\sqrt{d}{D_\dagger}^{3/2} B_T + \frac{dT}{\sqrt{D_\dagger}} + \sqrt{H|J|T}\big) \\
&= \tilde{\mathcal{O}}\big(\sqrt{d}H^{3/2} B_T + \frac{dT}{\sqrt{H}} + \sqrt{H|J|T}\big) \\
&= \tilde{\mathcal{O}}\big(d^{\frac{7}{8}} B_T T^{\frac{3}{4}} + d^{\frac{7}{8}} T^{\frac{3}{4}} + d^{\frac{1}{8}} T^{\frac{3}{4}} \Delta^{\frac{1}{2}}\big) \\
&= \tilde{\mathcal{O}}\big(d^{\frac{7}{8}} B_T^{\frac{1}{4}} T^{\frac{3}{4}}\big).
\end{aligned}$$

The last inequality holds by the condition $B_T = \tilde{\mathcal{O}}(1)$.

APPENDIX D

Detailed proofs for Differential Private Learnability

In this chapter, the proofs omitted in Chapter 7 are presented.

D.1 Section 7.1 details

We prove Lemma 7.1.1.

Lemma 7.1.1 (restated)**.** Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a class of multi-class hypotheses.

1. $\mathrm{Ldim}_\tau(\mathcal{H})$ is decreasing in τ.
2. SOA_τ (Algorithm 5) makes at most $\mathrm{Ldim}_\tau(\mathcal{H})$ mistakes with respect to ℓ_τ^{0-1}.
3. For any deterministic learning algorithm, an adversary can force $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ mistakes with respect to ℓ_τ^{0-1}.

Proof. Part 1 follows by observing that if T is a binary shattered tree with tolerance τ, then so is it with tolerance $\tau' < \tau$.

For part 2, assume SOA_τ makes a mistake at round t. We claim that $\mathrm{Ldim}_\tau(V_{t+1}) < \mathrm{Ldim}_\tau(V_t)$. If Ldim_τ does not decrease, we can infer that

$$\mathrm{Ldim}_\tau(V_t^{(\hat{y}_t)}) = \mathrm{Ldim}_\tau(V_t^{(y_t)}) = \mathrm{Ldim}_\tau(V_t) =: d.$$

Then we can find binary trees T_1 and T_2 of height d that are shattered by $V_t^{(\hat{y}_t)}$ and $V_t^{(y_t)}$, respectively. By concatenating T_1 and T_2 with a root node x_t and its edges labeled by $\hat{y}_t$ and y_t, we can obtain a binary tree T of height $d+1$ that is shattered by V_t. This contradicts to $\mathrm{Ldim}_\tau(V_t) = d$ and proves our assertion.

To prove part 3, let T be a binary shattered tree of height $\mathrm{Ldim}_{2\tau}(\mathcal{H})$. For a given node x, suppose the adversary shows x to the learner. Since the descending edges have labels apart from each other by more than 2τ, the adversary can choose a label that incurs a mistake with respect to ℓ_τ^{0-1}. Thus by following down the tree T from the root node, the adversary can force $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ mistakes. □

D.2 Section 7.2 details

In this section, the proofs omitted in Section 7.2 are presented.

D.2.1 Proof of Theorem 7.2.1

We first define *sub-trees*. Let T be a binary tree. Any node of T becomes its sub-tree of height 1. For $h > 1$, choose a node x and let T_1 and T_2 be the trees that are rooted at its two children. A sub-tree of height h is obtained by aggregating a sub-tree of height $h-1$ of T_1 and a sub-tree of height $h-1$ of T_2 at the root node x. Note that if the original tree T is shattered by some hypothesis class, then so is any sub-tree of it.

Next we prove a helper lemma.

Lemma D.2.1. *Suppose there are n colors $C = \{c_i\}_{1:n}$ and n positive integers $\{d_i\}_{1:n}$. Let T be a binary tree of height $-(n-1) + \sum_{i=1}^n d_i$ whose vertices are colored by C. Then there exists a color c_i such that T has a sub-tree of height d_i in which all internal vertices are colored by c_i.*

Proof. We will prove by induction on $\sum_{i=1}^n d_i$. If $d_i = 1$ for all i, then the height of T becomes 1, and the statement holds trivially. Now suppose the lemma holds for any d_i's whose summation is less than N and let T have the height $N - n + 1$. Without loss of generality, we may assume that the root node x_0 is colored by c_1. We consider two sub-trees T_1, T_2 of height $N - n$ whose root nodes are children of x_0. Let $e_1 = d_1 - 1$ and $e_i = d_i$ for $i > 1$. Since $\sum_{i=1}^n e_i = N - 1$, by the inductive assumption each T_j has a sub-tree of height e_{i_j} in which all internal vertices are colored by c_{i_j}. If $i_j \neq 1$ for some j, then we are done because $e_{i_j} = d_{i_j}$. If $i_j = 1$ for all $j = 1, 2$, then merging these two trees with the node x_0 forms a sub-tree of height $e_1 + 1 = d_1$ of color c_1. This completes the inductive argument. □

Now we are ready to prove Theorem 7.2.1.

Theorem 7.2.1 (restated)**.** Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively.

1. If $\mathrm{Ldim}_{2\tau}(\mathcal{H}) \geq d$, then $\mathcal{H}$ contains $\lfloor \frac{\log_K d}{K^2} \rfloor$ thresholds with a gap τ.

2. If $\mathrm{fat}_\gamma(\mathcal{F}) \geq d$, then $\mathcal{F}$ contains $\lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ thresholds with a margin $\frac{\gamma}{5}$.

Proof. We begin with the multi-class setting. Suppose $d = K^{K^2 t}$. It suffices to show $\mathcal{H}$ contains t thresholds. Let T be a shattered binary tree of height d and tolerance 2τ. Letting $\mathcal{H}_0 = \mathcal{H}$ and $T_0 = T$, we iteratively apply COLORANDCHOOSE (Algorithm 6). Namely, we write

$$k_n, k'_n, h_n, x_n, \mathcal{H}_n, T_n = \textsc{ColorAndChoose}(\mathcal{H}_{n-1}, T_{n-1}, 2\tau). \tag{D.1}$$

Observe that for all n, we can infer $h_n(x_n) = h_n(x) = k_n$ for all internal vertices x of T_n ($\because$ line 4 of Algorithm 6) and $h(x_n) = k'_n$ for all $h \in \mathcal{H}_n$ ($\because$ line 8 of Algorithm 6).

Additionally, Lemma D.2.1 ensures that the height of T_n is no less than $\frac{1}{K}$ times the height of T_{n-1}. This means that the iterative step (D.1) can be repeated $K^2 t$ times since $d = K^{K^2 t}$. Then there exist k, k' and indices $\{n_i\}_{i=1}^t$ such that $k_{n_i} = k$ and $k'_{n_i} = k'$ for all i.

It is not hard to check that the functions $\{h_{n_i}\}_{1:t}$ and the arguments $\{x_{n_i}\}_{1:t}$ form thresholds with labels k, k'. Since $|k - k'| > \tau$ ($\because$ line 6 of Algorithm 6), this completes the proof.

Now we move on to the regression setting. Proposition 7.1.1 implies that

$$\mathrm{Ldim}_{20}([\mathcal{F}]_{\gamma/50}) \geq \mathrm{Ldim}_{24}([\mathcal{F}]_{\gamma/50}) \geq d.$$

Then using the previous result in the multi-class setting, we can deduce that $[\mathcal{F}]_{\gamma/50}$ contains $n := \lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ thresholds with a gap 10. This means that there exist $k, k' \in [\frac{100}{\gamma}]$, $\{x_i\}_{1:n} \subset \mathcal{X}$, and $\{[f_i]_{\gamma/50}\}_{1:n} \subset \mathcal{H}$ such that $|k - k'| \geq 10$ and

$$[f_i]_{\gamma/50}(x_j) = \begin{cases} k & \text{if } i \leq j \\ k' & \text{if } i > j \end{cases}.$$

Let u, u' be the middles points of the intervals that correspond to the labels k, k'. Then it is easy to check that $|u - u'| \geq \gamma/5$ and

$$f_i(x_j) \in \begin{cases} [u - \frac{\gamma}{100}, u + \frac{\gamma}{100}) & \text{if } i \leq j \\ [u' - \frac{\gamma}{100}, u' + \frac{\gamma}{100}) & \text{if } i > j \end{cases}.$$

This proves the theorem. □

D.2.2 Proof of Theorem 7.2.2

Theorem 7.2.2 (restated)**.** Let $\mathcal{F} = \{f_i\}_{1:n} \subset [-1,1]^{\mathcal{X}}$ be a set of threshold functions with a margin γ on a domain $\{x_i\}_{1:n} \subset \mathcal{X}$ along with bounds $u, u' \in [-1,1]$. Suppose $\mathcal{A}$ is a $(\frac{\gamma}{200}, \frac{\gamma}{200})$-accurate learning algorithm for $\mathcal{F}$ with sample complexity m. If $\mathcal{A}$ is (ϵ, δ)-DP with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$, then it can be shown that $m \geq \Omega(\log^* n)$.

Proof. The proof consists of two main lemmas. Lemma D.2.2 proves that there is a large homogeneous set (see Definition D.2.1). Then Lemma D.2.4 yields the lower bound of the sample complexity when there exists a large homogeneous set. In particular, from these two lemmas, we can deduce that

$$\frac{\log^{(m)} n}{2^{O(m \log m)}} \leq 2^{O(m^2 \log^{(2)} m)}.$$

This means that there exists a constant c such that

$$\log^{(m)} n \leq e^{cm^2 \log m}.$$

Observing that $\log^* \big(\log^{(m)} n \big) \geq \big(\log^* n \big) - m$ and $\log^* \big(2^{O(m^2 \log^{(2)} m)} \big) = O(\log^* m)$, we can check the desired inequality $m \geq \Omega(\log^* n)$. □

D.2.2.1 Existence of a large homogenous set

Suppose $\mathcal{A}$ is a learning algorithm over a finite domain D. The hypothesis class consists of threshold functions over D with bounds u, u'. According to Definition 7.2.2, u and u' can be in an arbitrary order as long as $|u - u'| > \gamma$. But for simpler presentation, without loss of generality, we will assume $u > u'$. Also, let $\bar{u} = \frac{u+u'}{2}$. We define the following quantity:

$$\mathcal{A}_S(x) = \mathbb{P}_{f \sim \mathcal{A}(S)} \big(f(x) \geq \bar{u} \big).$$

The definition of homogenous sets (Definition D.2.1) and Lemma D.2.2 are adopted from Alon et al. (2019). Assume that $\mathcal{X}$ is linearly ordered. Given a training set $S = \big((x_i, y_i) \big)_{1:m}$, we say S is *increasing* if $x_1 \leq \cdots \leq x_m$. Additionally, we say S is *balanced* if $y_i = u'$ for all $i \leq \frac{m}{2}$ and $y_i = u$ for all $i > \frac{m}{2}$. Given $x \in \mathcal{X}$, we define $\mathrm{ord}_S(x) = \big| \{ i \mid x_i \leq x \} \big|$. Lastly, we use $S_{\mathcal{X}}$ to denote $(x_i)_{1:m}$.

Definition D.2.1 (m-homogeneous set)**.** A set $D' \subset D$ is m-homogeneous with respect to a learning algorithm $\mathcal{A}$ if there are numbers $p_i \in [0,1]$ for $0 \leq i \leq m$ such that for every

increasing balanced sample $S \in (D' \times \{u, u'\})^m$ and for every $x \in D' \setminus S_{\mathcal{X}}$

$$|\mathcal{A}_S(x) - p_i| \leq \frac{1}{100m},$$

where $i = \mathrm{ord}_S(x)$.

The following theorem is a well-known result in Ramsey theory. It was originally introduced by Erdos and Rado (1952) and rephrased by Alon et al. (2019).

Theorem D.2.1 (Alon et al. (2019, Theorem 11))**.** *Let $s > t \geq 2$ and q be integers, and let $N \geq \mathrm{twr}_t(3sq \log q)$. Then for every coloring of the subsets of size t of a universe of size N using q colors, there is a homogeneous subset [1] of size s.*

The next lemma states that we can find a large homogeneous set.

Lemma D.2.2 (Existence of a large homogeneous set)**.** *Let $\mathcal{A}$ be a learning algorithm over a domain D with $|D| = n$. Then there exists a set $D' \subset D$ which is m-homogeneous with respect to $\mathcal{A}$ such that*

$$|D'| \geq \frac{\log^{(m)} n}{2^{O(m \log m)}}.$$

Proof. We first define a coloring on the $(m+1)$-subsets of D. Let $B = \{x_1 < x_2 < \cdots < x_{m+1}\}$ be an $(m+1)$-subset. For each $i \in [m+1]$, let $B^{(i)} = B \setminus \{x_i\}$. Then by labeling the first half of $B^{(i)}$ by u' and the second half by u, we get a balanced increasing training set $S^{(i)}$. Then we compute p_i that is of the form $\frac{t}{100m}$ and closest to $\mathcal{A}_{S^{(i)}}(x_i)$ (in case of ties, choose the smaller one). Then we color B by the tuple $(p_i)_{1:m+1}$.

This scheme includes $(100m+1)^{m+1}$ colors, and Theorem D.2.1 provides that there exists a set D' of size larger than

$$\frac{\log^{(m)} n}{3(100m+1)^{m+1}(m+1)\log(100m+1)} = \frac{\log^{(m)} n}{2^{O(m \log m)}}$$

such that all $(m+1)$-subsets of D' have the same color. It is easy to verify that this set is indeed m-homogeneous with respect to $\mathcal{A}$ according to Definition D.2.1. □

D.2.2.2 Large homogeneous set implies the lower bound

Recall that PAC learning is defined with respect to $\mathrm{loss}_{\mathcal{D}}$ (see Definition 2.3.1). When $\mathrm{loss}_{\mathcal{D}}$ is replaced by loss_S, we say an algorithm $\mathcal{A}$ *empirically learns* a training set S. Bun et al. (2015, Lemma 5.9) prove that if a hypothesis class is PAC learnable, then there exists an empirical learner as well.

[1] A subset of the universe is homogeneous if all of its t-subsets have the same color.

Lemma D.2.3 (Empirical learner)**.** *Suppose $\mathcal{A}$ is an (ϵ, δ)-DP PAC learner for a hypothesis class $\mathcal{H}$ that is (α, β)-accurate and has sample complexity m. Then there is an (ϵ, δ)-DP and (α, β)-accurate empirical learner for $\mathcal{H}$ with sample complexity $9m$.*

The next is the main lemma.

Lemma D.2.4 (Large homogeneous sets imply lower bounds on sample complexity)**.** *Suppose a learning algorithm $\mathcal{A}$ is (ϵ, δ)-DP with sample complexity m. Let $X = [N]$ be m-homogeneous with respect to $\mathcal{A}$. If $\epsilon = 0.1$, $\delta \leq \frac{1}{1000m^2 \log m}$, and $\mathcal{A}$ empirically learns the threshold functions with a margin γ over X with $(\frac{\gamma}{200}, \frac{\gamma}{200})$-accuracy, then*

$$N \leq 2^{O(m^2 \log^{(2)} m)}.$$

Proof. The proof is done by combining Lemma D.2.5 and Lemma D.2.6, which come below. □

This is the first helper lemma to prove Lemma D.2.4. It adopts Alon et al. (2019, Lemma 12).

Lemma D.2.5. *Let $\mathcal{A}, X, m, N$ as in Lemma D.2.4 and assume $N > 2m$. Then there exists a family $\mathcal{P} = \{P_i\}_{1:N-m}$ of distributions over $\{-1, 1\}^{N-m}$ that satisfies the following two properties.*

1. *P_i and P_j are (ϵ, δ)-indistinguishable for all $i \neq j$.*
2. *There exists $r \in [0, 1]$ such that for all $i, j \in [N - m]$,*

$$\mathbb{P}_{v \sim P_i}(v_j = 1) \begin{cases} \leq r - \frac{1}{10m} & \text{if } j < i \\ \geq r + \frac{1}{10m} & \text{if } j > i \end{cases}.$$

Proof. Let $(p_i)_{0:m}$ be the probability list associated with m-homogeneous set $X = [N]$. We first prove that there exists i^* such that $p_{i^*} - p_{i^*-1} \geq \frac{1}{4m}$. Fix an increasing balanced training set $S := \big((x_i, y_i)\big)_{1:m} \in \big(X \times \{u, u'\}\big)^m$ such that $x_i - x_{i-1} \geq 2$ for all i, which is possible by the assumption $N > 2m$. By the definition of threshold functions with a margin γ, we can infer

$$\min_f \text{loss}_S(f) \leq \frac{\gamma}{20} = 0.05\gamma,$$

where the minimum is taken over the threshold functions with a margin γ.

Furthermore, since $\mathcal{A}$ is an $(\alpha = \frac{\gamma}{200}, \beta = \frac{\gamma}{200})$-accurate empirical learner, we can bound the expected loss of $\mathcal{A}(S)$ as

$$\mathbb{E}_{f\sim\mathcal{A}(S)}\text{loss}_S(f) \leq \alpha + \beta + \min_f \text{loss}_S(f) \leq 0.06\gamma. \tag{D.2}$$

Also, we can lower bound the expected empirical loss by using the quantity $\mathcal{A}_S(x_i)$ as follows (recall that we assumed $u > u'$)

$$\mathbb{E}_{f\sim\mathcal{A}(S)}\text{loss}_S(h) \geq \frac{1}{m}\cdot\frac{\gamma}{2}\left(\sum_{i=1}^{m/2}[\mathcal{A}_S(x_i)] + \sum_{i=m/2+1}^{m}[1-\mathcal{A}_S(x_i)]\right). \tag{D.3}$$

Combining (D.2) and (D.3), we can show that there exists $j \leq \frac{m}{2}$ such that $\mathcal{A}_S(x_j) \leq \frac{1}{4}$. Let $S' = (S \setminus \{(x_j, y_j)\}) \cup \{(x_j + 1, y_j)\}$. Since $\mathcal{A}$ is $(\epsilon = 0.1, \delta \leq \frac{1}{1000m^2\log m})$-DP, we have

$$p_{j-1} - \frac{1}{100m} \leq \mathcal{A}_{S'}(x_j) \leq \frac{1}{4}e^{\epsilon} + \delta \leq 0.3,$$

which implies that $p_{j-1} \leq 0.3 + \frac{1}{100m} \leq \frac{1}{3}$. Similarly, we can find $k > \frac{m}{2}$ such that $p_{k+1} \geq \frac{2}{3}$. Then we can find $i^* \in [j, k+1]$ such that $p_{i^*} - p_{i^*-1} \geq \frac{1}{4m}$, which proves our assertion.

Now we construct $\mathcal{P} = \{P_i\}_{1:N-m}$. Given i, let

$$B^{(i)} = \{1, \cdots, i^* - 1\} \cup \{i^* + i\} \cup \{i^* + N - m + 1, \cdots, N\} \subset X.$$

Observe that $B^{(i)}$ and $B^{(j)}$ only differ by one item at the position i^*. Then define $S^{(i)}$ to be the balanced increasing training set built upon $B^{(i)}$. Given a hypothesis f, we can compute a $N - m$ dimensional binary vector $v \in \{-1, 1\}^{N-m}$ such that

$$v_j = \mathbb{I}\left(f(i^* - 1 + j) \geq \bar{u}\right), \text{where } \bar{u} = \frac{u + u'}{2}.$$

This mapping induces a distribution over $\{-1, 1\}^{N-m}$ from $\mathcal{A}(S^{(i)})$, which we define to be P_i.

Due to DP property of $\mathcal{A}$, P_i and P_j are (ϵ, δ)-indistinguishable. Furthermore, our construction of i^* ensures the second property with $r = \frac{p_{i-1}+p_i}{2}$. This completes the proof. □

The second helper lemma is shown by Alon et al. (2019, Lemma 13).

Lemma D.2.6. *Suppose the family $\mathcal{P}$ as in Lemma D.2.5 exists. Then,*

$$N - m \leq 2^{1000m^2 \log^{(2)} m}.$$

D.3 Section 7.3 details

In this section, the proofs omitted in Section 7.3 are presented.

D.3.1 Proof of Theorem 7.3.2

Let $\mathcal{H}$ be a multi-class hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$ and $\mathcal{D}$ be a realizable distribution over examples $(x, c(x))$ where $c \in \mathcal{H}$ is an unknown target hypothesis. The globally-stable (GS) leaner G for $\mathcal{H}$ will make use of the Standard Optimal Algorithm (SOA_0, Algorithm 5).

SOA_0 can be simply extended to non-realizable sequences as follows.

Definition D.3.1 (Extending the SOA_0 to non-realizable sequences)**.** Consider a run of SOA_0 on examples $\big((x_i, y_i)\big)_{1:m}$, and let h_t denote the predictor used by the SOA_0 after observing the first t examples. Then after observing (x_{t+1}, y_{t+1}), proceed as below.

- If $\big((x_i, y_i)\big)_{1:t+1}$ is realizable by some $h \in \mathcal{H}$, then apply the usual update rule of the SOA_0 to obtain h_{t+1}.
- Else, set h_{t+1} as $h_{t+1}(x_{t+1}) = y_{t+1}$, and $h_{t+1}(x) = h_t(x)$ for every $x \neq x_{t+1}$. That is to say, h_{t+1} no longer belongs to $\mathcal{H}$.

This update rule keeps updating the predictor h_t to agree with the last example while observing the sequences which are not necessarily realized by a hypothesis in $\mathcal{H}$. Due to this extension, our resulting algorithm possibly becomes improper.

The finite Littlestone class is online learnable by SOA_0 (Algorithm 5) with at most d mistakes on any realizable sequence. Prior to building a GS learner G, we define a distribution $\mathcal{D}_k$ as in Algorithm 7.

Let k be such that $\mathcal{D}_k$ is well-defined and consider a sample S drawn from $\mathcal{D}_k$. The size of $\mathcal{D}_k$ is $k \cdot (n+1)$, and they consist of $k \cdot n$ instances randomly drawn from $\mathcal{D}$ and k examples generated in Item 3(iv) of Algorithm 7. We call these k examples *tournament examples*. Due to the construction of $\mathcal{D}_k$, SOA_0 always errs in tournament rounds, which means that SOA_0 makes at least k mistakes when run on $S \circ T$ where $S \sim \mathcal{D}_k, T \sim \mathcal{D}^n$.

Algorithm 7 Distribution $\mathcal{D}_k$

1: $\mathcal{D}_0$: output an empty set with probability 1
2: Let $k \geq 1$. If there exists an f satisfying $\mathbb{P}_{S\sim\mathcal{D}_{k-1},T\sim\mathcal{D}^n}\big(\text{SOA}_0(S\circ T)=f\big) \geq K^{-d}$, or if $\mathcal{D}_{k-1}$ is undefined, then $\mathcal{D}_k$ is undefined
3: Else, $\mathcal{D}_k$ is defined recursively as follows
4: (i) Randomly sample $S_0, S_1 \sim \mathcal{D}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$
5: (ii) Let $f_0 = \text{SOA}_0(S_0 \circ T_0)$ and $f_1 = \text{SOA}_0(S_1 \circ T_1)$
6: (iii) If $f_0 = f_1$, go back to step (i)
7: (iv) Else, pick $x \in \{x \mid f_0(x) \neq f_1(x)\}$ and sample $y \sim [K]$ uniformly at random
8: (v) If $f_0(x) \neq y$, output $S_0 \circ T_0 \circ (x, y)$ and $S_1 \circ T_1 \circ (x, y)$ otherwise

A natural way to obtain a GS learning algorithm G is to run the SOA_0 on this carefully chosen sample $S \circ T$. In fact, the output enjoys both global stability in multi-class learning and good generalization as follows.

Lemma D.3.1 (Global Stability)**.** *There exist $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that*

$$\mathbb{P}_{S\sim\mathcal{D}_k,T\sim\mathcal{D}^n}\big(\text{SOA}_0(S\circ T)=f\big) \geq K^{-d}.$$

Proof. Assume for contradiction that $\mathcal{D}_d$ is well-defined and for every f,

$$\mathbb{P}_{S\sim\mathcal{D}_k,T\sim\mathcal{D}^n}\big(\text{SOA}_0(S\circ T)=f\big) < K^{-d}.$$

In each tournament example (x_i, y_i), the label y_i is drawn uniformly at random from $[K]$. Accordingly, with probability K^{-d} over $S \sim \mathcal{D}_d$, all d tournament examples are consistent with the true labeling function c and thus $S\circ T$ becomes consistent with c. Since the number of total mistakes of SOA_0 should be no more than d, we can deduce that $\text{SOA}_0(S\circ T) = c$. This implies that

$$\mathbb{P}_{S\sim\mathcal{D}_k,T\sim\mathcal{D}^n}\big(\text{SOA}_0(S\circ T)=c\big) \geq K^{-d},$$

which is a contradiction, and hence completes the proof. □

Lemma D.3.2 (Generalization)**.** *Let k be such that $\mathcal{D}_k$ is well-defined. Then for every f such that*

$$\mathbb{P}_{S\sim\mathcal{D}_k,T\sim\mathcal{D}^n}\big(\text{SOA}_0(S\circ T)=f\big) \geq K^{-d}$$

satisfies $\text{loss}_{\mathcal{D}}(f) \leq \frac{d\log K}{n}$.

Proof. Let f be such hypothesis and let $\alpha = \text{loss}_{\mathcal{D}}(f)$. We will argue that $K^{-d} \leq (1-\alpha)^n$. Then the following result is derived, $\alpha \leq \frac{d\log K}{n}$ using the fact that $(1-\alpha)^n \leq e^{-n\alpha}$.

By the property of SOA_0, $\text{SOA}_0(S\circ T)$ is consistent with T. Thus, if $\text{SOA}_0(S\circ T) = f$, then it must be the case that f is consistent with T. By assumption, $\text{SOA}_0(S \circ T) = f$

holds with probability at least K^{-d} and f is consistent with T with probability $(1-\alpha)^n$ where n is the size of T. This gives the desired inequality. □

One challenge associated with the distribution $\mathcal{D}_k$ is computational limitation. It may require an unbounded number of samples from the target distribution $\mathcal{D}$, since during generation of tournament examples the number of samples drawn from $\mathcal{D}$ depends on how many times Item 3(i)-(iii) will be repeated. To handle this practical issue, we suggest a Monte-Carlo Variant of $\mathcal{D}_k$, $\tilde{\mathcal{D}}_k$, by setting an upper bound N of random samples drawn from $\mathcal{D}$ as an input parameter. Algorithm 8 summarizes how we construct the distribution $\tilde{\mathcal{D}}_k$.

Algorithm 8 Distribution $\tilde{\mathcal{D}}_k$

1: Let n be the auxiliary sample size and N be an upper bound on the number of samples from $\mathcal{D}$
2: $\tilde{\mathcal{D}}_0$: output an empty set with probability 1
3: Let $k \geq 1$. $\tilde{\mathcal{D}}_k$ is defined recursively by the following processes
4: ($\star$) Throughout the process, if more than N examples are drawn from $\mathcal{D}$, then output "Fail"
5: (i) Randomly sample $S_0, S_1 \sim \tilde{\mathcal{D}}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$
6: (ii) Let $f_0 = \mathrm{SOA}_0(S_0 \circ T_0)$ and $f_1 = \mathrm{SOA}_0(S_1 \circ T_1)$
7: (iii) If $f_0 = f_1$, go back to step (i)
8: (iv) Else, pick $x \in \{x \mid f_0(x) \neq f_1(x)\}$ and sample $y \sim [K]$ uniformly at random
9: (v) If $f_0(x) \neq y$, output $S_0 \circ T_0 \circ (x, y)$ and $S_1 \circ T_1 \circ (x, y)$ otherwise

The next step is to specify the upper bound N. The following lemma characterizes the expected sample complexity of sampling from $\mathcal{D}_k$.

Lemma D.3.3 (Expected sample complexity of sampling from $\mathcal{D}_k$)**.** *Let k be such that $\mathcal{D}_k$ is well-defined and M_k be the number of samples from $\mathcal{D}$ when generating $S \sim \mathcal{D}_k$. Then we have $\mathbb{E}M_k \leq 4^{k+1} \cdot n$.*

Proof. Initially, $\mathbb{E}M_0 = 0$ since $\mathcal{D}_0$ outputs an empty set with probability 1. It suffices to show that for all $0 < i < k$, $\mathbb{E}M_{i+1} \leq 4\mathbb{E}M_i + 4n$ to conclude the desired inequality by induction.

Let R be the number of times Item 3(i) was executed during generation of $S \sim \mathcal{D}_{i+1}$, and R is distributed geometrically with a success probability θ, where

$$\begin{aligned}
\theta &= 1 - \mathbb{P}_{S_0,S_1,T_0,T_1}\big(\mathrm{SOA}_0(S_0 \circ T_0) = \mathrm{SOA}_0(S_1 \circ T_1)\big) \\
&= 1 - \sum_f \big(\mathbb{P}_{S,T}\big(\mathrm{SOA}_0(S \circ T) = f\big)\big)^2 \\
&\geq 1 - K^{-d}.
\end{aligned}$$

The last inequality holds because $i < k$ and hence $\mathcal{D}_i$ is well-defined, which implies that $\mathbb{P}_{S,T}\big(\text{SOA}_0(S \circ T) = f\big) \leq K^{-d}$ for all f.

Let M_{i+1} be a random variable expressed as $M_{i+1} = \sum_{j=1}^{\infty} M_{i+1}^{(j)}$ where

$$M_{i+1}^{(j)} = \begin{cases} 0, & \text{if } R < j \\ \text{the number of examples from } \mathcal{D} \text{ in the } j\text{-th execution of Item 3(i)}, & \text{if } R \geq j \end{cases}.$$

Thus, we have

$$\begin{aligned} \mathbb{E}M_{i+1} &= \sum_{j=1}^{\infty} \mathbb{E}M_{i+1}^{(j)} = \sum_{j=1}^{\infty} (1-\theta)^{j-1} \cdot (2\mathbb{E}M_i + 2n) \\ &= \frac{1}{\theta} \cdot (2\mathbb{E}M_i + 2n) \leq 4\mathbb{E}M_i + 4n, \end{aligned}$$

where the last inequality holds since $\theta \geq 1 - K^{-d} \geq 1/2$ since $K \geq 2$ and $d \geq 1$. □

Equipped with Lemma D.3.1,D.3.2, and D.3.3, we are ready to prove Theorem 7.3.2.

Theorem 7.3.2 (restated)**.** Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\text{Ldim}(\mathcal{H}) = d$. Let $\alpha > 0$, and $m = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{\alpha}]$. Then there exists a randomized algorithm $G : (\mathcal{X} \times [K])^m \to [K]^{\mathcal{X}}$ such that for a realizable distribution $\mathcal{D}$ and an input sample $S \sim \mathcal{D}^m$, there exists a h such that

$$\mathbb{P}\big(G(S) = h\big) \geq \frac{K-1}{(d+1)K^{d+1}} \quad \text{and} \quad loss_{\mathcal{D}}(h) \leq \alpha.$$

Proof. The globally-stable algorithm G is defined in Algorithm 9.

Algorithm 9 Algorithm G

1: **Input :** target distribution $\tilde{\mathcal{D}}_k$, auxiliary sample size $n = [\frac{d \log K}{\alpha}]$, and the sample complexity upper bound $N = (4K)^{d+1} \cdot n$
2: Draw $k \in \{0, 1, \cdots, d\}$ uniformly at random
3: **Output :** $h = \text{SOA}_0(S \circ T)$, where $T \sim \mathcal{D}^n, S \sim \tilde{\mathcal{D}}_k$

The sample complexity of G is $|S| + |T| \leq N + n = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{n}]$. By Lemma D.3.1 and D.3.2, there exists $k^\star \leq d$ and $f^\star$ such that

$$\mathbb{P}_{S \sim \mathcal{D}_{k^\star}, T \sim \mathcal{D}^n}\big(\text{SOA}(S \circ T) = f^\star\big) \geq \frac{1}{K^d}, \quad \text{loss}_{\mathcal{D}}(f^\star) \leq \frac{d \log K}{n} \leq \alpha.$$

Let $M_{k^\star}$ denote the number of random examples from $\mathcal{D}$ during generation of $S \sim \mathcal{D}_{k^\star}$.

We obtain the following inequality from Lemma D.3.3 and Markov's inequality,

$$\mathbb{P}\big(M_{k^\star} > (4K)^{d+1} \cdot n\big) \le \mathbb{P}\big(M_{k^\star} > K^{d+1} \cdot 4^{k^\star+1} \cdot n\big) \le K^{-(d+1)}.$$

Accordingly,

$$\begin{aligned}
&\mathbb{P}_{S\sim\tilde{\mathcal{D}}_{k^\star}, T\sim\mathcal{D}^n}\big(\mathbf{SOA}_0(S\circ T) = f^\star\big) \\
&\qquad \ge \mathbb{P}_{S\sim\mathcal{D}_{k^\star}, T\sim\mathcal{D}^n}\big(\mathbf{SOA}_0(S\circ T) = f^\star \text{ and } M_{k^\star} \le (4K)^{d+1}\cdot n\big) \\
&\qquad \ge \mathbb{P}_{S\sim\mathcal{D}_{k^\star}, T\sim\mathcal{D}^n}\big(\mathbf{SOA}_0(S\circ T) = f^\star\big) - \mathbb{P}\big(M_{k^\star} > (4K)^{d+1}\cdot n\big) \\
&\qquad \ge K^{-d} - K^{-(d+1)} = (K-1)K^{-(d+1)}
\end{aligned}$$

Since $k = k^\star$ with probability $\frac{1}{d+1}$, G outputs $f^\star$ with probability at least $\frac{K-1}{(d+1)K^{d+1}}$. □

D.3.2 Globally-stable learning implies private multi-class learning

In this section, we utilize the GS algorithm from the previous section to derive a DP learning algorithm with a finite sample complexity. Theorem 7.3.1 establishes that online multi-class learnability implies private multi-class learnability, which can be proved by combining Theorem 7.3.2 and Theorem D.3.1.

Theorem D.3.1 (Globally-stable learning implies private multi-class learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a multi-class hypothesis class. Let $G : (\mathcal{X}\times[K])^m \to [K]^{\mathcal{X}}$ be a randomized algorithm such that for a realizable distribution $\mathcal{D}$ and $S\sim\mathcal{D}^m$, there exists a hypothesis h such that $\mathbb{P}\big(G(S) = h\big) \ge \eta$ and $\text{loss}_{\mathcal{D}}(h) \le \alpha/2$. Then for some $n = O(\frac{m\log(1/\eta\beta\delta)}{\eta\epsilon} + \frac{\log(1/\eta\beta)}{\alpha\epsilon})$, there exists an (ϵ,δ)-DP algorithm M which for n i.i.d. samples from $\mathcal{D}$, outputs a hypothesis $\hat{h}$ such that $\text{loss}_{\mathcal{D}}(\hat{h}) \le \alpha$ with probability at least $1-\beta$.*

To construct a private learner M, we first introduce standard tools in the DP community such as *Stable Histogram* and *Generic Private Learner*.

Lemma 7.3.1 (Stable Histogram, restated)**.** Let X be any data domain. For $n \ge O(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon})$, there exists an (ϵ,δ)-DP algorithm HIST which with probability at least $1-\beta$, on input $S = (x_1,\cdots,x_n)$ outputs a list $L \in X$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that

1. Every x with $\text{Freq}_S(x) \ge \eta$ appears in L, and
2. For every $x \in L$, the estimate a_x satisfies $|a_x - \text{Freq}_S(x)| \le \eta$,

where $\text{Freq}_S(x) = \big|\{i\in[n] \mid x_i = x\}\big|/n$.

Lemma D.3.4 (Generic Private Learner, (Bun et al., 2020)). *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a collection of multi-class hypotheses. For $n = O(\frac{\log|\mathcal{H}|+\log(1/\beta)}{\alpha\epsilon})$, there exists an $(\epsilon, 0)$-DP algorithm* GENERICLEARNER *: $(\mathcal{X} \times [K])^n \to \mathcal{H}$ satisfying the following; let $\mathcal{D}$ be a distribution over $\mathcal{X} \times [K]$ such that there exists an $h^\star \in \mathcal{H}$ with $loss_{\mathcal{D}}(h^\star) \leq \alpha$. Then on input $S \sim \mathcal{D}^n$,* GENERICLEARNER *outputs, with probability at least $1 - \beta$, a hypothesis $\hat{h} \in \mathcal{H}$ such that $loss_S(\hat{h}) \leq 2\alpha$.*

Now we are ready to prove Theorem D.3.1.

Proof of Theorem D.3.1. The learning algorithm M is built on top of the Stable Histogram and the Generic Private Learner as described in Algorithm 10. According to Lemma 7.3.1 and D.3.4, we choose parameters

$$k = O\big(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon}\big), \quad n' = O\big(\frac{\log(1/\eta\beta)}{\alpha\epsilon}\big).$$

Algorithm 10 Differentially-Private Learner M

1: Let $S_1, \cdots, S_k$ each consist of i.i.d. samples of size m from $\mathcal{D}$. Run G on each batch of samples producing $h_1 = G(S_1), \cdots, h_k = G(S_k)$
2: Run the Stable Histogram algorithm HIST on input $H = (h_1, \cdots, h_k)$ using privacy $(\epsilon/2, \delta)$ and accuracy $(\eta/8, \beta/3)$, publishing a list L of frequent hypotheses
3: Let S' consist of n' i.i.d. samples from $\mathcal{D}$. Run GENERICLEARNER(S') using L with privacy $\epsilon/2$ and accuracy $(\alpha/2, \beta/3)$ to output a hypothesis $\hat{h}$

We show that the algorithm M is (ϵ, δ)-DP. During the executions of $G(S_1), \cdots G(S_k)$, a change to one entry in a certain S_i changes at most one outcome $h_i \in H$. Thus, differential privacy for this step is observed by taking expectations over the coin tosses of all the executions of G. Then the differential privacy for overall algorithm holds by simple composition of differentially-private HIST and GENERICLEARNER.

Next, we prove that the algorithm M is accurate. By standard generalization arguments, we have with probability at least $1 - \beta/3$,

$$\big|\text{Freq}_H(h) - \mathbb{P}_{S\sim\mathcal{D}^m}\big(G(S) = h\big)\big| \leq \frac{\eta}{8}$$

for every $h \in [K]^{\mathcal{X}}$ as long as $k \geq O(\log(1/\beta)/\eta)$. Conditioned on this event, by accuracy of HIST, with probability $1 - \beta/2$, it produces a list L containing $h^\star$ together with a sequence of estimates that are accurate to within an additive error $\eta/8$. Then, $h^\star$ appears in L with an estimate $a_{h^\star} \geq \eta - \eta/8 - \eta/8 = 3\eta/4$.

Now remove from L every item h with $a_h \leq \frac{3\eta}{4}$. Since every estimate is accurate within $\eta/8$, h appears in L such that $\text{Freq}_H(h) \geq \frac{3\eta}{4} - \frac{\eta}{8} = \frac{5\eta}{8}$. Since sum of frequencies is less than 1, the number of list L should be less than $2/\eta$ (i.e. $|L| \leq 2/\eta$). This list contains $h^\star$ such that $\text{loss}_{\mathcal{D}}(h^\star) \leq \alpha$. Hence the GENERICLEARNER identifies $h^\star$ with $\text{loss}_{\mathcal{D}}(h^\star) \leq \alpha/2$ with probability at least $1 - \beta/3$. □

D.3.3 Extension to the agnostic setting

Theorem 7.3.1 showed that online MC learnability continues to imply private MC learnability in the realizable setting. A similar result also holds even when the realizability assumption is violated, which is called *agnostic setting*.

Corollary D.3.1 (Agnostic setting : Online MC learning implies private MC learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\text{Ldim}(\mathcal{H}) = d$. Let $\epsilon, \delta \in (0,1)$ be privacy parameters and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For $n = O_d\big(\frac{\log(1/\beta\delta)}{\alpha^2\epsilon}\big)$, there exists (ϵ, δ)-DP learning algorithm such that for every distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies*

$$\textit{loss}_{\mathcal{D}}(f) \leq \min_{h \in \mathcal{H}} \textit{loss}_{\mathcal{D}}(h) + \alpha$$

with probability at least $1 - \beta$.

Proof. Alon et al. (2020, Theorem 6) propose an algorithm, $\mathcal{A}_{PrivateAgnostic}$, which transforms a private learner in the realizable setting to a private learner that can operate in the agnostic setting. The main idea is based on the standard sub-sampling method, and as a result, the transformed agnostic learner has a larger sample complexity by a factor of $1/\epsilon$. Then Corollary D.3.1 is shown by applying $\mathcal{A}_{PrivateAgnostic}$ to the realizable learner used in Theorem 7.3.1. □

D.3.4 Proof of Theorem 7.3.3

We complete the proof of Theorem 7.3.3. The proof for Condition 4 is given in the main body.

Theorem 7.3.3 (restated)**.** Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a real-valued function class such that $\text{fat}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$. If one of the following conditions holds, then $\mathcal{F}$ is privately learnable.

1. Either $\mathcal{F}$ or $\mathcal{X}$ is finite.
2. The range of $\mathcal{F}$ over $\mathcal{X}$ is finite (i.e., $\big|\{f(x) \mid f \in \mathcal{F}, x \in \mathcal{X}\}\big| < \infty$).

3. $\mathcal{F}$ has a finite cover with respect to the sup-norm at every scale.

4. $\mathcal{F}$ has a finite sequential Pollard Pseudo-dimension.

Proof. 1. If $|\mathcal{F}| < \infty$, then for sample complexity $n = \mathcal{O}(\frac{\log|\mathcal{F}|+\log(1/\beta)}{\alpha\epsilon})$ we directly run the ϵ-DP Generic Private Learner to output with probability at least $1-\beta$, a hypothesis $\hat{f} \in \mathcal{F}$ such that $\text{loss}_S(\hat{f}) \leq \alpha$. Next, assume that $\mathcal{X}$ is finite. The finiteness of $\mathcal{X}$ does not imply finite $|\mathcal{F}|$ because $\mathcal{Y}$ is continuous, but we can discretize $\mathcal{F}$ at some scale γ, which gives us a finite MC hypothesis class $[\mathcal{F}]_\gamma$. It is private-learnable by ϵ-DP Generic Private Learner, and then the original class $\mathcal{F}$ is also privately-learnable within accuracy γ.

2. Observe that this regression problem is essentially a MC problem. Furthermore, $\text{Ldim}(\mathcal{F})$ by considering it as a MC problem is bounded above by $\text{fat}_\gamma(\mathcal{F})$, where γ is the minimal gap between consecutive values in the range of $\mathcal{F}$ over $\mathcal{X}$. This means that $\text{Ldim}(\mathcal{F})$ is finite, and hence by the argument of Section 7.3.1, $\mathcal{F}$ is privately learnable.

3. Given an accuracy α, $\mathcal{F}$ has n finite covers with a radius $r < \alpha$. We construct a set of representative function as $\mathcal{F}' = \{f_1, \cdots, f_n\} \subset \mathcal{F}$ by arbitrarily choosing a representative f_i from the i-th cover, and then run ϵ-DP Generic Private Learner on $\mathcal{F}'$ to output a hypothesis $\hat{f} \in \mathcal{F}$ with a small population loss. □

BIBLIOGRAPHY

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Abe, N., Biermann, A. W., and Long, P. M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4):263–293.

Abeille, M., Lazaric, A., et al. (2017). Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197.

Abernethy, J., Lee, C., Sinha, A., and Tewari, A. (2014). Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823.

Abernethy, J., Lee, C., and Tewari, A. (2015). Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems*, pages 2197–2205.

Abernethy, J. D., Jung, Y. H., Lee, C., McMillan, A., and Tewari, A. (2019). Online learning via the differential privacy lens. In *Advances in Neural Information Processing Systems*, pages 8892–8902.

Abramowitz, M. and Stegun, I. A. (1964). Handbook of mathematical functions. 1965.

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646. PMLR.

Agarwal, N. and Singh, K. (2017). The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 32–40. JMLR. org.

Agrawal, S. (2019). Recent advances in multiarmed bandits for sequential decision making. In *Operations Research & Management Science in the Age of Analytics*, pages 167–188. INFORMS.

Agrawal, S. and Goyal, N. (2013a). Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107.

Agrawal, S. and Goyal, N. (2013b). Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135.

Alon, N., Beimel, A., Moran, S., and Stemmer, U. (2020). Closure properties for private classification and online prediction. volume 125 of *Proceedings of Machine Learning Research*, pages 119–152. PMLR.

Alon, N., Livni, R., Malliaris, M., and Moran, S. (2019). Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860.

Audibert, J.-Y. and Bubeck, S. (2009). Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, pages 217–226.

Audibert, J.-Y. and Bubeck, S. (2010). Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(Oct):2785–2836.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Auer, P., Gajane, P., and Ortner, R. (2019). Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158. PMLR.

Bartlett, P. L., Long, P. M., and Williamson, R. C. (1996). Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452.

Beimel, A., Nissim, K., and Stemmer, U. (2013). Characterizing the sample complexity of private learners. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 97–110.

Ben-David, S., Pál, D., and Shalev-Shwartz, S. (2009). Agnostic online learning. In *Conference on Learning Theory*, volume 3, page 1.

Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pages 199–207.

Besbes, O. and Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420.

Bogunovic, I., Krause, A., and Scarlett, J. (2020). Corruption-tolerant gaussian process bandit optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1071–1081. PMLR.

Bogunovic, I., Losalka, A., Krause, A., and Scarlett, J. (2021). Stochastic linear bandits robust to adversarial attacks. In *The 24th International Conference on Artificial Intelligence and Statistics*. PMLR.

Bogunovic, I., Scarlett, J., and Cevher, V. (2016). Time-varying gaussian process bandit optimization. In *Artificial Intelligence and Statistics*, pages 314–323. PMLR.

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

Bun, M., Livni, R., and Moran, S. (2020). An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE.

Bun, M., Nissim, K., Stemmer, U., and Vadhan, S. (2015). Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 634–649. IEEE.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Chapelle, O., Manavoglu, E., and Rosales, R. (2014). Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):1–34.

Chen, Y., Lee, C.-W., Luo, H., and Wei, C.-Y. (2019). A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. In *Proceedings of the 32nd Annual Conference on Learning Theory*.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2019). Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1079–1087. PMLR.

Chowdhury, S. R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR.

Coles, S., Bawa, J., Trenner, L., and Dorazio, P. (2001). *An introduction to statistical modeling of extreme values*, volume 208. Springer.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*.

Daniely, A., Sabato, S., Ben-David, S., and Shalev-Shwartz, S. (2015). Multiclass learnability and the erm principle. *The Journal of Machine Learning Research*, 16(1):2377–2404.

Devroye, L., Lugosi, G., and Neu, G. (2013). Prediction by random-walk perturbation. In *Conference on Learning Theory*, pages 460–473.

Diemert, E., Meynet, J., Galland, P., and Lefortier, D. (2017). Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the ADKDD'17*, pages 1–6.

Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407.

Erdos, P. and Rado, R. (1952). Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London mathematical Society*, 3(1):417–439.

Feldman, V. and Xiao, D. (2014). Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Gonen, A., Hazan, E., and Moran, S. (2019). Private learning implies online learning: An efficient reduction. In *Advances in Neural Information Processing Systems*, pages 8699–8709.

Gupta, A., Koren, T., and Talwar, K. (2019). Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, pages 1562–1578. PMLR.

Hannan, J. (1957). Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139.

Hazan, E. (2019). Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*.

Hazan, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Hazan, T., Papandreou, G., and Tarlow, D., editors (2017). *Perturbations, Optimization and Statistics*. MIT Press.

Hoeffding, W. (1994). Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer.

Hofbauer, J. and Sandholm, W. H. (2002). On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294.

Jung, Y. H., Kim, B., and Tewari, A. (2020). On the equivalence between online and private learnability beyond binary classification. In *Advances in Neural Information Processing Systems*.

Kalai, A. and Vempala, S. (2005). Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kim, B. and Tewari, A. (2019). On the optimality of perturbations in stochastic and adversarial multi-armed bandit problems. In *Advances in Neural Information Processing Systems*, pages 2691–2700.

Kim, B. and Tewari, A. (2020). Randomized exploration for non-stationary stochastic linear bandits. In *Conference on Uncertainty in Artificial Intelligence*, pages 71–80. PMLR.

Korolova, A., Kenthapadi, K., Mishra, N., and Ntoulas, A. (2009). Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, pages 171–180.

Kujala, J. and Elomaa, T. (2005). On following the perturbed leader in the bandit setting. In *International Conference on Algorithmic Learning Theory*, pages 371–385. Springer.

Kuleshov, V. and Precup, D. (2014). Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*.

Kveton, B., Szepesvari, C., Ghavamzadeh, M., and Boutilier, C. (2019). Perturbed-history exploration in stochastic linear bandits. In *Proceedings of the 35th Annual Conference on Uncertainty in Artificial Intelligence*.

Kveton, B., Zaheer, M., Szepesvari, C., Li, L., Ghavamzadeh, M., and Boutilier, C. (2020). Randomized exploration in generalized linear bandits. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.

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

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Leadbetter, M. R., Lindgren, G., and Rootzén, H. (2012). *Extremes and related properties of random sequences and processes*. Springer Science & Business Media.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.

Li, N., Lyu, M., Su, D., and Yang, W. (2016). Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(4):1–138.

Li, Y., Lou, E. Y., and Shan, L. (2019). Stochastic linear optimization with adversarial corruption. *arXiv preprint arXiv:1909.02109*.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318.

Luo, H., Wei, C.-Y., Agarwal, A., and Langford, J. (2018). Efficient contextual bandits in non-stationary worlds. In *Proceedings of the 31st Annual Conference on Learning Theory*.

Lykouris, T., Mirrokni, V., and Paes Leme, R. (2018). Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. (2019). Corruption robust exploration in episodic reinforcement learning. *arXiv preprint arXiv:1911.08689*.

McFadden, D. (1981). Econometric models of probabilistic choice. *Structural analysis of discrete data with econometric applications*, 198272.

Rakesh, K. and Suganthan, P. N. (2017). An ensemble of kernel ridge regression for multi-class classification.

Rakhlin, A., Sridharan, K., and Tewari, A. (2015). Online learning via sequential complexities. *The Journal of Machine Learning Research*, 16(1):155–186.

Resnick, S. I. (2013). *Extreme values, regular variation and point processes*. Springer.

Russac, Y., Vernade, C., and Cappé, O. (2019). Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026.

Sarwate, A. D. and Chaudhuri, K. (2013). Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94.

Scott, S. L. (2010). A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, pages 844–853. PMLR.

Tewari, A. and Murphy, S. A. (2017). From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, pages 495–517. Springer.

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

Tossou, A. C. and Dimitrakakis, C. (2016). Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Tossou, A. C. Y. and Dimitrakakis, C. (2017). Achieving privacy in the adversarial multi-armed bandit. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. (2013). Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty in Artificial Intelligence*. PMLR.

Van Erven, T., Kotłowski, W., and Warmuth, M. K. (2014). Follow the leader with dropout perturbations. In *Conference on Learning Theory*, pages 949–974.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vaswani, S., Mehrabian, A., Durand, A., and Kveton, B. (2020). Old dog learns new tricks: Randomized ucb for bandit problems. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*.

Wong, K. C., Li, Z., and Tewari, A. (2019). Lasso guarantees for β-mixing heavy tailed time series. *Annals of Statistics*.

Woodroofe, M. (1979). A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806.

Yang, Z., Deng, N., and Tian, Y. (2005). A multi-class classification algorithm based on ordinal regression machine. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 810–815. IEEE.

Zhao, P. and Zhang, L. (2021). Non-stationary linear bandits revisited. *arXiv preprint arXiv:2103.05324*.

Zhao, P., Zhang, L., Jiang, Y., and Zhou, Z.-H. (2020). A simple approach for non-stationary linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 746–755. PMLR.

Zimmert, J. and Seldin, Y. (2019). An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 467–475. PMLR.