MULTI-ARMED BANDITS IN LARGE-SCALE COMPLEX SYSTEMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Xiao Xu

May 2020

MULTI-ARMED BANDITS IN LARGE-SCALE COMPLEX SYSTEMS

Xiao Xu, Ph.D.

Cornell University 2020

This dissertation focuses on the multi-armed bandit problem (MAB) where the objective is a sequential arm selection policy that maximizes the total reward over time. In canonical formulations of MAB, the following assumptions are adopted: the size of the action space is much smaller than the length of the time horizon, computation resources such as memory are unlimited in the learning process, and the generative models of arm rewards are time-invariant. This dissertation aims to relax these assumptions, which are unrealistic in emerging applications involving large-scale complex systems, and develop corresponding techniques to address the resulting new issues.

The first part of the dissertation aims to address the issue of a massive number of actions. A stochastic bandit problem with side information on arm similarity and dissimilarity is studied. The main results include a unit interval graph (UIG) representation of the action space that succinctly models the side information and a two-step learning structure that fully exploits the topological structure of the UIG to achieve an optimal scaling of the learning cost with the size of the action space. Specifically, in the UIG representation, each node represents an arm and the presence (absence) of an edge between two nodes indicates similarity (dissimilarity) between their mean rewards. Based on whether the UIG is fully revealed by the side information, two settings with complete and partial side information are considered. For each setting, a two-step learning policy consisting of an offline reduction of the action space and online aggregation

of reward observations from similar arms is developed. The computation efficiency and the order optimality of the proposed strategies in terms of the size of the action space and the time length are established. Numerical experiments on both synthetic and real-world datasets are conducted to verify the performance of the proposed policies in practice.

In the second part of the dissertation, the issue of limited memory during the learning process is studied in the adversarial bandit setting. Specifically, a learning policy can only store the statistics of a subset of arms summarizing their reward history. A general hierarchical learning structure that trades off the regret order with memory complexity is developed based on multi-level partitions of the arm set into groups and the time horizon into epochs. The proposed learning policy requires only a sublinear order of memory space in terms of the number of arms. Its sublinear regret orders with respect to the time horizon are established for both weak regret and shifting regret in expectation and/or with high probability, when appropriate learning strategies are adopted as subroutines at all levels. By properly choosing the number of levels in the adopted hierarchy, the policy adapts to different sizes of the available memory space. A memory-dependent regret bound is established to characterize the tradeoff between memory complexity and the regret performance of the policy. Numerical examples are provided to verify the performance of the policy.

The third part of the dissertation focuses on the issue of time-varying rewards within the contextual bandit framework, which finds applications in various online recommendation systems. The main results include two reward models characterizing the fact that the preferences of users toward different items change asynchronously and distinctly, and a learning algorithm that adapts to the dynamic environment. In particular, the two models assume

disjoint and hybrid rewards. In the disjoint setting, the mean reward of playing an arm is determined by an arm-specific preference vector, which is piecewise-stationary with asynchronous change times across arms. In the hybrid setting, the mean reward of an arm also depends on a joint coefficient vector shared by all arms representing the time-invariant component of user interests, in addition to the arm-specific one that is time-varying. Two algorithms based on change detection and restarts are developed in the two settings respectively, of which the performance is verified through simulations on both synthetic and real-world data. Theoretical regret analysis of the algorithm with certain modifications is provided under the disjoint reward model, which shows that a near-optimal regret order in the time length is achieved.

BIOGRAPHICAL SKETCH

Xiao Xu received his B.S. degree in Electronic Engineering from Tsinghua University, Beijing, China, and the M.S. degree in Electrical and Computer Engineering from Cornell University, Ithaca, NY, USA, in 2011 and 2018, respectively. He will receive the Ph.D. degree from Cornell University in 2020. His research interests include statistical online learning, sequential decision theory, algorithmic game theory, and distributed learning with applications in various socio-economic networks and communication systems.

To my family, friends, and my beloved fiancée.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest and most sincere gratitude to my Ph.D. advisor, Professor Qing Zhao, for her guidance, support, encouragement, and help during my Ph.D. study. She is an outstanding scholar with great enthusiasm and integrity, a brilliant researcher with deep insights and broad horizons, and an experienced mentor with incredible patience. I have acquired so many academic skills from her on how to find potential research directions, formulate and solve problems, write technical and tutorial articles, and give academic presentations. All the achievements in my graduate studies could not happen without her countless help. It is my honor to be her student.

I am also fortunate to have Professor Jayadev Acharya and Professor Yudong Chen as my committee members. I have learned a lot in discussions with and lectures from them. Their informative suggestions really help me deepen and sharpen my academic thinking. I would like to extend my gratitude to them.

During my Ph.D. study, I did two impressive internships at the U.S. Army Research Lab (ARL) in Adelphi, Maryland, and Alibaba Group in Hangzhou, China. I would like to thank my mentor Dr. Ananthram Swami at ARL for his insightful suggestions in the research project as well as his help in my settlement in a new environment. Chapter 2 in this dissertation is based on the joint work with him. At Alibaba Group, I learned from my colleagues and mentors: Fang Dong, Yanghua Li, Shaojian He, and Xin Li on how to formulate problems from the view of industrial and commercial practices and apply theoretical results to real-world applications. Chapter 4 in this dissertation is based on our collaborated paper.

I have been extremely fortunate to work with a wonderful group of colleagues and labmates at Cornell: Sattar Vakili, Chao Wang, Boshuang Huang,

Ziteng Sun, Huanyu Zhang, Yuhan Liu, and Sudeep Salgia who have made my time as a graduate student fun and memorable. In particular, I would like to thank Sattar Vakili for providing me with initial guidance when I started working on multi-armed bandit problems. Special thanks to Chao Wang for his help in nearly every aspect of research at my early stages as a Ph.D. student. I spent wonderful times with Boshuang Huang, Ziteng Sun and Huanyu Zhang in Ithaca. We can discuss anytime at various places: in offices, at restaurants, and on WeChat. I would also like to thank my friends Bo Hu and Yang Liu with whom, I can share my happiness and sadness in my good and bad times.

Last but most importantly, I am so grateful to my parents, who have always been my strongest backing with unconditional love in my entire life, and my beloved fiancée Yuejun Niu, whose unreserved trust, dedication, and support across Eurasia and the Atlantic Ocean give me courage to overcome all the challenges in my study and life throughout all these years, especially in my darkest hours. There are no words to convey how much I appreciate and love them. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation focuses on the problem of online learning and sequential decision-making under unknown models. The objective in this class of problems is to learn, in real time, the most rewarding actions among a number of options. Example applications include various socio-economic applications (e.g., ad display in search engines, product/news recommendation systems, targeted marketing, political campaigns, and drug therapy in clinic trials), and networking issues in communication systems (e.g., dynamic channel access) and urban transportation (e.g., route selection).

The problem is formulated and studied under the classic framework of *multi-armed bandits* (MAB) in this dissertation. We point out several emerging issues and new challenges in applications with large-scale complex systems that call for new models and new learning strategies, and develop corresponding solutions with performance guarantees in both theory and practice.

1.1 Multi-Armed Bandits

The MAB problem was first posed in [60] for the application of clinical trials. In a bandit model, potential actions with unknown rewards are abstracted as arms of a slot machine. At every time in a horizon of length T, a player selects one arm to play and receives a reward generated from an unknown reward model. The objective of the player is to choose sequentially which arm to play based on past reward observations, with the hope of improved performance over time. The essence of the problem is in the tradeoff between exploration—

to gather information from less explored arms—and exploitation—to maximize the instantaneous reward by favoring arms with better reward history.

A commonly adopted performance measure of an arm selection policy is *regret*, defined as the (expected) cumulative reward loss over the entire time horizon against a properly defined benchmark policy with hindsight vision and/or certain clairvoyant knowledge about the problem. A policy is said to achieve no-regret learning if the induced regret has a sublinear growth rate in T. In other words, the policy offers, asymptotically as $T \to \infty$, the same average reward as the specific benchmark adopted in the corresponding regret measure.

1.1.1 Reward Models and Regret Measures

Depending on the generative model of arm rewards, bandit problems can be categorized into the stochastic and the adversarial settings. In the former, rewards from successive plays of an arm obey a given, albeit unknown, stochastic model. In the latter, rewards are assigned by an adversary.

Earlier studies on MAB focused on the stochastic setting. The canonical model assumes that rewards from each arm are drawn i.i.d. from a fixed distribution. In this case, the benchmark policy in the regret definition is to play the arm with the greatest mean reward throughout the time horizon, and the regret is measured in expectation taken over the randomness of both reward realizations and the arm selection policy. There exist two settings in evaluating the regret performance of a learning policy: problem-specific and problem-independent. In the former, the regret is specific to the set of reward distributions associated with the given problem instance. In the latter, the regret is

measured against the worst-case assignment of reward distributions.

In the problem-specific setting, the seminal work by Lai and Robins in 1985 showed that the minimum regret growth rate is $\Omega(\log T)$ [46]. A number of learning policies have since been developed that offer the optimal regret order in T (see [9, 34, 61] and references therein). In the problem-independent setting, an $\Omega(\sqrt{T})$ lower bound on regret was shown in [10] and an optimal learning policy was later proposed in [7].

A number of variations of stochastic bandits have been studied in recent years for diverse application domains. One notable example in the application of personalized recommendation is the contextual bandit formulation where the reward distributions are affected by certain context information revealed at each time. The context information can be feature vectors associated with either the current user or the available items to be recommended.

Under the commonly adopted assumption of linear rewards, the mean reward of playing an arm at each time step is assumed to be the inner product of the currently revealed context vector and an unknown coefficient vector representing the preference of users towards items. In this model, the benchmark policy in the regret definition is to play the best arm specific to the current context information at every time step. The regret performance is usually evaluated under the problem-independent setting where the lower bound was shown to be $\Omega(\sqrt{T})$ [24]. A number of near-optimal policies that achieve a regret order of $O(\sqrt{T}\text{polylog}(T))$ have been developed (see [24, 1] and references therein).

The adversarial bandit problem, first studied in [10], was motivated by the problem of learning in repeated unknown games. In the game setting, a player's

reward of playing an action (arm) is jointly determined by the payoff function of the game and the actions taken by all opponents, which can be aggregated as an adversary from the view of the player [20]. Connections between the regret performance of every player and certain system-level objectives (e.g., convergence to equilibria of the game) have been revealed [68, 54, 31]. A comprehensive survey on distributed no-regret learning in multi-agent systems can be found in our tutorial paper [67].

Various benchmark policies have been considered, leading to different regret notions. Corresponding to the external regret in the game setting, weak regret was proposed in [11], which is defined against the best fixed arm with the greatest cumulative reward in hindsight. The weak regret is evaluated against the worst-case assignment of the reward sequence by the adversary (not necessarily follows a stochastic model). It was proven in [11] that the lower bound on weak regret is in the order of $\Omega(\sqrt{T})$, which was shown to be achieved by a class of randomized policies proposed in [7]. It should be noted that randomization is necessary to achieve no-regret learning against an adversary: it was shown in [13] that for every deterministic policy, there always exists a reward sequence that inflicts a linear regret order in T.

A stronger regret notion is the shifting regret, which is defined against a sequence of actions with a hardness constraint on the number of action changes. Achieving no-regret learning under this stronger regret notion and its variations plays a key role in achieving certain optimality in terms of the system-level performance in games with dynamically changing compositions [54, 31].

1.1.2 Emerging Issues and Challenges

In the past few decades, the MAB problem with various reward models has been extensively studied in their canonical forms. However, existing results are usually established upon idealistic assumptions in terms of the small size of the action space compared with the time length, the availability of unlimited resources such as memory during the learning process, and the stationarity of the underlying reward models. Many emerging complex systems, however, involve a massive number of actions with a limited memory space, and are inherently dynamic in the reward models.

In addressing the aforementioned issues, the results of this dissertation are partitioned into three parts. In the first part, we focus on the issue of a massive number of actions in the stochastic bandit setting. We develop optimal learning strategies that scale well with the large action space. In the second part, we study the problem of learning with memory constraints in the adversarial bandit setting. We develop memory-efficient learning strategies that trades off the regret order with the memory complexity. In the third part, we consider time-varying reward models in the contextual bandit setting. We develop near-optimal learning strategies that adapts to the changing environment. We summarize the main results of the three parts in the following section.

1.2 Main Results

1.2.1 Bandits with Many Arms

The first part of this dissertation focuses on the stochastic bandit problem with a large action space. Classical solutions to stochastic bandits were developed under the assumption of independent arms, i.e., there is no structure in the set of reward distributions. As a result, a linear scaling of regret with the size of the action space is unavoidable due to exploring every arm sufficiently often to identify the optimal. For applications involving a massive number of arms, those solutions are no longer suitable.

The key to achieving a sublinear scaling with the number of arms is to exploit the inherent structures of the action space, i.e., various relations among the vast number of actions. In this part, we consider the statistical similarity and dissimilarity relations across arms, which is formulated through the difference between the expected rewards of arms. We first show that the similarity-dissimilarity structure of the action space can be represented by a *unit interval graph* (UIG) where the presence (absence) of an edge between two arms indicates that the difference of their mean rewards is within (beyond) a given threshold. Based on whether the UIG is fully revealed to the player, we consider two cases of complete and partial side information.

For both cases, we propose a general two-step learning structure—*LSDT (Learning from Similarity-Dissimilarity Topology)*—to achieve a full exploitation of the topological structure of the side information. The first step is an offline reduction of the action space to a candidate set, which consists of arms that

can assume the largest mean rewards under certain assignments of reward distributions without violating the side information. Arms outside the candidate set are sub-optimal and hence eliminated from online exploration. The second step carries out an online learning algorithm that further exploits the similarity structure through collective exploration using aggregated reward observations from similar arms.

The order optimality of the proposed learning strategies in terms of both the size of the action space and the time length was established in both cases with complete and partial side information. Specifically, we provide theoretical regret analysis of the learning strategies in the problem-specific setting along with matching lower bounds. Extensive numerical experiments on both synthesized and real-world datasets are conducted to verify the performance of the learning strategies in practice.

1.2.2 Bandits with Limited Memory

In the second part of this dissertation, we study the memory-constrained MAB problem under the adversarial setting. Existing policies for canonical adversarial bandits require a memory space linear in the number K of arm to store certain statistics of every arm, which is infeasible in applications involving a large action space but limited memory.

In the problem of memory-constrained adversarial bandits, a policy is only allowed M words of memory space (which has a sublinear growth rate with K) for storing input values and necessary variables. Therefore, a policy with memory size M can only store the statistics of at most M arms at any given time. As

a result, two new problems arise in addition to arm selection: one on deciding the statistics of which arms to store in the memory at every time step, the other on how to memorize the reward history of arms whose statistics are not stored.

To address the new issues induced by the memory constraint and trade off between memory complexity and regret performance, we propose a general hierarchical learning structure—*HLMC (Hierarchical Learning with Memory Constraints)*—based on multi-level partitions of the arm set into groups and the time horizon into epochs through a properly designed hierarchy. At every level of the hierarchy, every arm group (time epoch) is further partitioned into several next-level groups (epochs). The policy recursively zooms into an arm group selected for every epoch at the same level, and carries out a next-level selection strategy until the end of the epoch. At the last level, a group of arms are targeted and their arm statistics are stored in the memory for arm selection within the corresponding epoch. The reward information of the other arms outside the targeted group are jointly memorized in certain aggregated group statistics that are used for group selection at higher levels.

We show that HLMC requires a memory space with size sublinear in K in a representative case with a two-level hierarchy. By adopting appropriate selection strategies as subroutines at all levels, the HLMC policy achieves sublinear regret orders in T under notions of both weak regret and shifting regret. We further establish a memory-dependent regret bound for the general case with a D-level hierarchy to characterize the tradeoff between the memory complexity and the regret order of HLMC. By properly selecting the depth D of the adopted hierarchy, the HLMC policy adapts to different sizes of the available memory space and achieves no-regret learning.

1.2.3 Bandits in Dynamic Environments

The third part of this dissertation focuses on the contextual bandit problem with dynamic reward models. In applications such as online recommendation, user interests are dynamically changing and the preference changes toward different items may be asynchronous and distinct. To characterize such phenomena, we study two reward models: the disjoint and the hybrid reward models.

In the disjoint reward model, the expected reward of playing an arm is the inner product of the given context vector and an arm-specific unknown coefficient vector, which represents the preference of the user towards the arm. The preference vector is assumed to be piecewise-stationary and the change points are different across arms. We propose an upper confidence bound (UCB) based algorithm—*PSLinUCB (Piecewise-Stationary Linear UCB)*—that selects arms based on estimates of the unknown preference vectors from past observations. To address the challenge of time-varying interests, the algorithm adopts a change-detection procedure to identify potential changes on the preference vectors, and an efficient restart procedure after detected changes to re-estimate the preference vectors using up-to-date observations.

We further extend the algorithm to the general hybrid reward model. In addition to the arm-specific preference vector, the expected reward in the hybrid model also depends on a joint coefficient vector shared by all arms, which corresponds to the time-invariant component of user interests. We conduct experiments on both synthesized data and real-world datasets to evaluate the performance of the proposed algorithms in both models.

We also provide theoretical guarantee on the regret performance of the pro-

posed algorithm. To avoid certain technical difficulties in analysis, we introduce a modified PSLinUCB algorithm and analyze its regret performance in the disjoint reward model. We show that a near-optimal regret order in T is achieved in the problem-independent setting.

1.3 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we discuss bandits with many arms in the stochastic setting. We introduce the LSDT learning structure that fully exploits the topological structure of the side information. In Chapter 3, we consider bandits with memory constraints in the adversarial setting. We present the HLMC learning structure that trades off the regret performance with memory complexity. In Chapter 4, we study bandits in dynamic environments under the contextual bandit framework. We propose the PSLinUCB algorithm that adapts to the time-varying environment in both disjoint and hybrid reward models. Chapter 5 concludes the dissertation. Additional results and all proofs are included in the Appendices.

CHAPTER 2

BANDITS WITH MANY ARMS

In addressing the issue of a massive number of arms, there has been a growing body of studies aiming at exploiting certain side information on the relations among the large number of arms. Among various formulations of the side information (see a more detailed discussion in Sec. 2.1), one notable example is the statistical similarity and dissimilarity among arms. For instance, in recommendation systems and information retrieval, products, ads, and documents in the same category (more generally, close in some feature space) have similar expected rewards. At the same time, it may also be known *a priori* that some arms have considerably different mean rewards, e.g., news with drastically different opinions, products with opposite usage, documents associated with key words belonging to distant categories in the taxonomy.

The side information on arm similarity and dissimilarity opens the possibility of efficient solutions that scale well with the large action space. In this chapter, we introduce a mathematical formulation of such arm relations in the stochastic bandit setting, and provide a UIG representation of the action space. The central question we seek to answer in this chapter is: *how to fully exploit the topological structure of the action space with side information to achieve an optimal regret order in terms of both the size of the action space and the time length?*

2.1 Literature Review of MAB with Structured Action Space

Existing studies on MAB with structured action space and reward models can be categorized based on the types of arm relations adopted in the MAB models. The first type is realization-based relation that assumes a certain known probabilistic dependency across arms. Examples include combinatorial bandits [52, 33, 23, 45], linearly parameterized bandits [30, 57, 1], and spectral bandits for smooth graph functions [64, 39]. The second type of arm relation can be termed as observation-based relation [18, 15, 6]. Specifically, playing an arm provides additional side observations about its neighboring arms. See [63] for a survey on various bandit models with structured action spaces.

The problem studied in this chapter considers another type of relation among arms: ensemble-based relation that aims to capture the relations on ensemble behaviors (i.e., mean rewards) across arms, rather than probabilistic dependencies in their realizations. Related work includes Lipschitz bandits [2, 44, 55], taxonomy bandits [59] and unimodal bandits [26]. Specifically, in Lipschitz bandits, the mean reward is assumed to be a Lipschitz function of the arm parameter. Taxonomy bandits have a tree-structured action space where arms in the same subtree are close in their mean rewards. In unimodal bandits, the action space is represented by a graph where from every sub-optimal arm, there exists a path to the optimal arm along which the mean reward increases.

Different from these existing studies, the bandit model studied in this chapter considers an action space represented by a UIG indicating not only similarity but also dissimilarity relations across actions. Besides, the structure of the proposed learning policy consists of a two-level exploitation of the UIG structure,

which is fundamentally different from the existing ones. Recently, a general formulation of structured bandits was proposed in [25], which includes a variety of known bandit models (e.g., Lipschitz bandits, unimodal bandits, linear bandits, etc.) as well as the bandit model studied in this work as special cases. The learning policy developed in [25], however, was given only implicitly in the form of a linear program (LP) that needs to be solved at every time step. For the problem studied in this chapter, the LP does not admit polynomial-time solutions (unless P=NP).

2.2 Problem Formulation

Consider a stochastic K-armed bandit problem. At each time t, a player chooses one arm to play. Playing an arm i yields a reward $X_i(t)$ drawn i.i.d. from an unknown distribution f_i with mean μ_i. We assume that f_i belongs to the family of sub-Gaussian distributions[1] for all i. Extensions to other distribution types will be discussed later in Sec. 2.6.

Across K arms, the similarity and dissimilarity relations are defined through a parameter $\epsilon > 0$: two arms are similar (dissimilar) if the difference between their mean rewards is below (above) ϵ. The similarity-dissimilarity structure of the action space can be represented by an undirected graph $\mathcal{G}_\epsilon^* = (\mathcal{V}, \mathcal{E}_\epsilon^*)$. In the graph representation, every node $i \in \mathcal{V}$ represents an arm with reward distribution f_i and the presence (absence) of an edge (i, j) corresponds to a similar (dissimilar) arm pair. Throughout this chapter, $1 \leq i \leq K$ is used to refer to an arm or a node, exchangeably. We first show that $\mathcal{G}_\epsilon^*$ is a UIG.

[1] A random variable Y with mean μ is sub-Gaussian with parameter σ (or σ sub-Gaussian) if $\mathbb{E}[e^{\lambda(Y-\mu)}] \leq e^{\sigma^2\lambda^2/2}$, for all $\lambda \in \mathbb{R}$ [16].

Definition 1 (Unit interval graph and unit interval model) *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a unit interval graph if there exists a set of unit length intervals $\{I_i\}_{i\in\mathcal{V}}$ on the real line such that each interval I_i corresponds to a node $i \in \mathcal{V}$ and there exists an edge $(i, j) \in \mathcal{E}$ if and only if $I_i \cap I_j \neq \emptyset$. The set of intervals $\{I_i\}_{i\in\mathcal{V}}$ is a unit interval model (UIM) for the UIG.*

It should be noted that if a UIG is finite (with a finite number of nodes), there is no difference between taking open intervals or closed intervals to represent nodes [32]. Without loss of generality, we assume that $I_i = (l_i, r_i)$ where l_i, r_i are the left and right coordinates of interval I_i.

Through a mapping from every node $i \in \mathcal{V}$ to an ϵ-length interval $I_i = (\mu_i, \mu_i + \epsilon)$, it is not difficult to see that

$$|\mu_i - \mu_j| < \epsilon \Leftrightarrow I_i \cap I_j \neq \emptyset, \tag{2.1}$$

which indicates that $\mathcal{G}_\epsilon^*$ is a UIG (see an example in Fig. 2.1). Without loss of generality, we assume that $\mathcal{G}_\epsilon^*$ is connected. Extensions to the disconnected case will be discussed in Sec. 2.6.

We define $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$ as the side information on arm similarity and dissimilarity. Based on whether $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$ fully reveal the UIG $\mathcal{G}_\epsilon^*$, we consider the following two cases separately. In the case of complete side information, $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$ are identical to the edge set and the complement edge set of $\mathcal{G}_\epsilon^*$, i.e., $\mathcal{E}_\epsilon^S = \mathcal{E}_\epsilon^*, \mathcal{E}_\epsilon^D = \overline{\mathcal{E}_\epsilon^*}$. In the case of partial side information, they are subsets of the latter, i.e., $\mathcal{E}_\epsilon^S \subseteq \mathcal{E}_\epsilon^*, \mathcal{E}_\epsilon^D \subseteq \overline{\mathcal{E}_\epsilon^*}$.

The objective is an online learning policy π that specifies a sequential arm selection rule at each time t based on both past observations of selected arms and the side information $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$. The performance of policy π is measured by regret $R_\pi(T; \mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D)$ defined as the expected reward loss against a player who

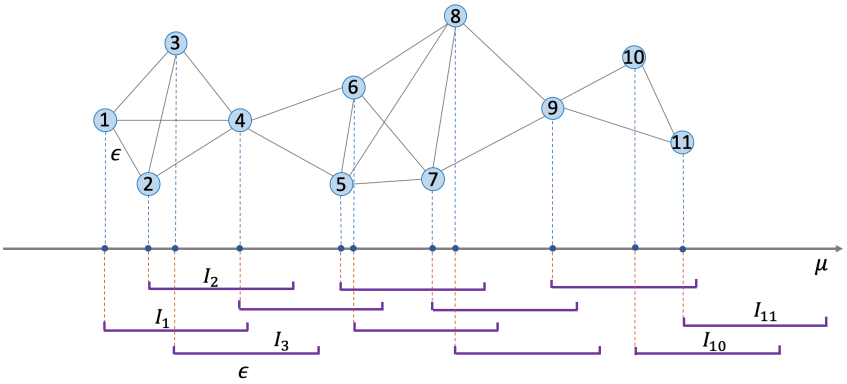

Figure 2.1: Action space as a UIG.

knows the reward model and always plays the best arm $i_{\max}$ (chosen arbitrarily in the case of multiple optimal arms), i.e.,

$$R_\pi(T; \mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \mu_{i_{\max}}(t) - \sum_{t=1}^{T} X_{\pi_t}(t) \right], \tag{2.2}$$

where $\mu_{i_{\max}}$ is the largest mean reward and π_t is the arm selected by policy π at time t. In this chapter, we consider the problem-specific regret measure, i.e., the regret is a function of the unknown reward distributions $\boldsymbol{f} = (f_1, ..., f_K)$. When there is no ambiguity, the dependency of regret on $\boldsymbol{f}$ is omitted and the notation is simplified to $R(T)$.

Let $\tau_i(T)$ denote the number of times that arm i has been selected up to time T. We rewrite the regret as:

$$R(T) = \mu_{i_{\max}} T - \sum_{i=1}^{K} \mu_i \mathbb{E}[\tau_i(T)] = \sum_{i=1}^{K} \Delta_i \mathbb{E}[\tau_i(T)], \tag{2.3}$$

where $\Delta_i = \mu_{i_{\max}} - \mu_i$. The objective of maximizing the expected cumulative reward is equivalent to minimizing the regret over a time horizon of length T. In order to minimize regret, it can be inferred from (2.3) that every sub-

optimal arm ($\Delta_i > 0$) should be distinguished from the optimal one with the least number of plays.

2.3 Two-Step Learning Structure

While classic stochastic bandit algorithms have to try out every arm sufficiently often to distinguish the sub-optimal arms from the optimal one, which induces a linear scaling of regret in the number of arms, the side information on arm similarity and dissimilarity allows the possibility of identifying a set of sub-optimal arms without even playing them. To be specific, we define a candidate set $\mathcal{B}$ determined by the side information $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$ as follows.

Definition 2 (Candidate Arm and Candidate Set) *Given the side information $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$, an arm i is a candidate arm if there exists an assignment of reward distributions with means $\boldsymbol{\mu} = (\mu_1, ..., \mu_K)$ conforming to $\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D$ and $\mu_i = \max_{1 \le j \le K} \mu_j$. The candidate set $\mathcal{B}$ is the set consisting of all candidate arms.*

Note that the optimal arm $i_{\max}$ under the ground truth assignment of reward distributions in the bandit problem always belongs to the candidate set $\mathcal{B}$. It is clear that if we can find the candidate set $\mathcal{B}$ from the side information efficiently, the action space can be reduced to $\mathcal{B}$. Only arms in $\mathcal{B}$ need to be explored. Furthermore, certain topological structures of the revealed UIG on the reduced action space can be further exploited to accelerate learning. In estimating the mean reward of every arm in the candidate set, observations from similar arms can also be leveraged as approximations, which reduces the number of plays required to distinguish sub-optimal arms from the optimal one.

The aforementioned facts motivate a general two-step learning structure: *Learning from Similarity-Dissimilarity Topology (LSDT)* for both cases of complete and partial side information. Specifically, LSDT consists of (1) an offline elimination step that reduces the action space to the candidate set and (2) online learning of the optimal arm by aggregating observations from similar ones. We specify each step for the cases of complete and partial side information separately in Sec. 2.4 and Sec. 2.5.

2.4 Complete Side Information

We first consider the case of complete side information that fully reveals the UIG $\mathcal{G}_\epsilon^*$. We follow the two-step learning structure proposed in Sec. 2.3 and develop a learning policy: *LSDT-CSI (Learning from Similarity-Dissimilarity Topology with Complete Side Information)* along with theoretical analysis on its regret performance. While restrictive in applications, this case provides useful insights for tackling the general case of partial side information addressed in Sec. 2.5.

2.4.1 Offline Elimination

The first step of LSDT-CSI is an offline preprocessing that aims at identifying the candidate set from the complete side information. Since the UIG $\mathcal{G}_\epsilon^*$ is fully revealed, we denote the candidate set in this case as $\mathcal{B}^*$ to distinguish from the case of partial side information. We show that $\mathcal{B}^*$ is identical to the set of *left anchors* of the UIG $\mathcal{G}_\epsilon^*$.

Definition 3 (Left Anchor) *Given a UIG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $i \in \mathcal{V}$ is a left anchor if there exists a UIM for $\mathcal{G}$ where i corresponds to the leftmost interval along the real line.*

Since the mirror image of an UIM with respect to the origin is also an UIM for the same UIG, the node corresponding to the rightmost interval in a UIM is also a left anchor. Based on the definition of the UIG $\mathcal{G}_\epsilon^*$ that represents the similarity-dissimilarity structure of the arm set in Sec. 2.2, it is not difficult to see that the candidate set $\mathcal{B}^*$ is identical to the set of left anchors of $\mathcal{G}_\epsilon^*$, which can be identified through a BFS-based algorithm proposed in [27]. The BFS-based algorithm starts from an arbitrary node in a UIG and returns a set of left anchors. We apply the algorithm two times: in the first time, we start from an arbitrary node in $\mathcal{G}_\epsilon^*$ and obtain a set of left anchors. In the second time, we re-apply the algorithm starting from one of the returned node in the last time. One can directly infer from Proposition 2.1 and Theorem 2.3 in [27] that the obtained set is the candidate set $\mathcal{B}^*$. The detailed algorithm is summarized in Algorithm 1.

Algorithm 1: **Offline Elimination of LSDT-CSI**

Input: Fully revealed UIG $\mathcal{G}_\epsilon^*$.

Output: Candidate set $\mathcal{B}^*$.

Initialization: $\mathcal{B}^* = \emptyset$.

Start from an arbitrary node i and perform a BFS on $\mathcal{G}_\epsilon^*$.

Let $\mathcal{L}$ be the set of nodes in the last level of the BFS.

for each $j \in \mathcal{L}$ **do**

  if $\deg(j) = \min_{k \in \mathcal{L}} \deg(k)$ **then**

    $\mathcal{B}^* \leftarrow \mathcal{B}^* \cup \{j\}$.

Start from a node $j \in \mathcal{B}^*$ and repeat the previous steps.

Note that the computation complexity of the offline elimination step is $O(|\mathcal{E}_\epsilon^*|)$, which is polynomial in the problem size.

2.4.2 Online Aggregation

We now present the second step of online learning that further exploits topological structures of the candidate set $\mathcal{B}^*$. We first introduce an equivalence relation between nodes in the UIG $\mathcal{G}_\epsilon^*$.

Definition 4 (Neighborhood Equivalence) *Two nodes i, j in $\mathcal{G}_\epsilon^*$ are (neighborhood) equivalent if $\mathcal{N}[i] = \mathcal{N}[j]$, where $\mathcal{N}[i]$ is the set of neighbors of i in $\mathcal{G}_\epsilon^*$, including i. Moreover, let $\{\mathcal{B}_i^*\}$ denote the partition of the arm set $\mathcal{V}$ in $\mathcal{G}_\epsilon^*$ with respect to the neighborhood equivalence relation.*

Note that arms within the same equivalence class have the same set of neighbors and thus, they are topologically indistinguishable in the UIG. Based on the equivalence class partition, we obtain a closed-form expression for $\mathcal{B}^*$.

Theorem 1 *When the side information fully reveals the UIG $\mathcal{G}_\epsilon^*$ (assumed to be connected), the candidate set $\mathcal{B}^*$ is the union of two equivalence classes containing the optimal arm $i_{\max}$ and the worst arm $i_{\min}$ (with minimum mean reward). Note that the two equivalence classes containing the optimal and the worst arm are identical in the special case where $\mathcal{G}^*$ is fully connected. The proposed algorithm and analysis still apply in this case. Without loss of generality, we assume that $\mathcal{G}_\epsilon^*$ is not fully connected., i.e.,*

$$\mathcal{B}^* = \mathcal{B}_{i_{\max}}^* \cup \mathcal{B}_{i_{\min}}^*, \tag{2.4}$$

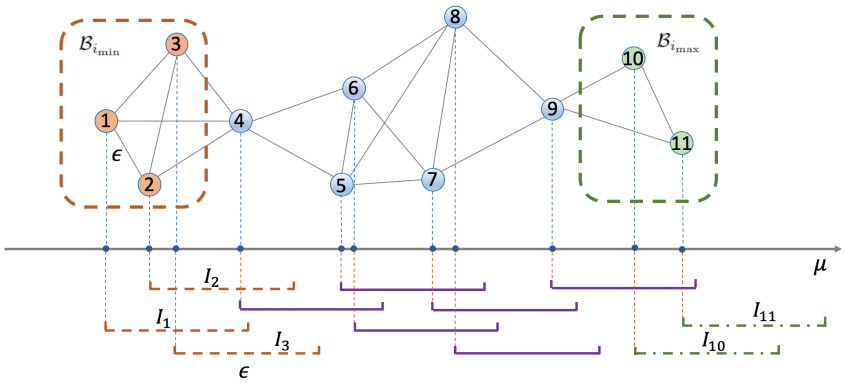

Figure 2.2: Left anchors and the candidate set

where

$$\mathcal{B}^*_{i_{\max}} = \{j : \mathcal{N}[j] = \mathcal{N}[i_{\max}]\}, \tag{2.5}$$

$$\mathcal{B}^*_{i_{\min}} = \{j : \mathcal{N}[j] = \mathcal{N}[i_{\min}]\}. \tag{2.6}$$

Proof 1 *See Appendix A.1.*

The result is also illustrated in Fig. 2.2 : the node corresponding to I_1 (or I_{11}) is the left anchor under the current UIM (or its mirroring). Switching I_1, I_2, I_3 (or I_{10}, I_{11}) does not change the graph connectivity, i.e., each node in $\mathcal{B}^*_{i_{\min}} = \{1, 2, 3\}$ (or $\mathcal{B}^*_{i_{\max}} = \{10, 11\}$) is a left anchor. Hence the candidate set $\mathcal{B} = \{1, 2, 3\} \cup \{10, 11\}$ is the union of the two equivalence classes, which can be directly obtained through the offline elimination step.

Based on the topological structure of the candidate set, we develop a hierarchical online learning policy that aggregates observations from arms within the same equivalence class. By considering each class as a super node (arm), we reduce the problem to a simple two-armed bandit problem.

Specifically, the second step of LSDT-CSI carries out a hierarchical UCB-based online learning on the candidate set $\mathcal{B}^*$ by maintaining a class index $H_i(t)$ for each equivalence class $\mathcal{B}_i^*$ and an arm index $L_j(t)$ for each individual arm j in $\mathcal{B}^*$. The arm index is defined as:

$$L_j(t) = \bar{x}_j(t) + \sqrt{\frac{8 \log t}{\tau_j(t)}}, \tag{2.7}$$

where $\bar{x}_j(t)$, $\tau_j(t)$ are the empirical average of observations from arm j and the number of times that arm j has been played up to time t. The class index $H_i(t)$ aggregates the same statistics across arms in the class:

$$H_i(t) = \frac{\sum_{j \in \mathcal{B}_i^*} \bar{x}_j(t) \tau_j(t)}{\sum_{j \in \mathcal{B}_i^*} \tau_j(t)} + \sqrt{\frac{8 \log t}{\sum_{j \in \mathcal{B}_i^*} \tau_j(t)}}. \tag{2.8}$$

At each time, the online learning procedure selects the equivalence class with the largest class index and plays the arm with the largest arm index within the selected class. Once the reward has been observed, both class indices and arm indices are updated.

Algorithm 2: **Online Aggregation of LSDT-CSI**

Input: Candidate set $\mathcal{B}^* = \mathcal{B}_1^* \cup \mathcal{B}_2^*$ where $\mathcal{B}_1^*, \mathcal{B}_2^*$ are two disjoint equivalence classes.

Initialization: Play each arm in $\mathcal{B}^*$ once, update all the arm indices $\{L_j(t)\}_{j \in \mathcal{B}^*}$ and class indices $H_1(t), H_2(t)$ defined in (2.7) and (2.8).

for $t = |\mathcal{B}^*| + 1, |\mathcal{B}^*| + 2, \ldots$ **do**

Let $i_t^* = \text{argmax}_{i \in \{1,2\}} H_i(t-1)$.

Play arm $j_t^* = \text{argmax}_{j \in \mathcal{B}_{i_t^*}^*} L_j(t-1)$.

It should be noted that the two-step learning structure LSDT is independent of the specific arm selection rule adopted at the online learning step. In

particular, different arm selection techniques developed for the original bandit problems may be incorporated into the second step of LSDT, except based on aggregated observations. In Sec. 2.6.3, we discuss the use of Thompson Sampling (TS), another representative strategy in stochastic bandits (see [60, 21, 3, 4] and references therein), with LSDT to fully exploit the side information.

2.4.3 Order Optimality

We first present the regret analysis of LSDT-CSI, which focuses on upper bounding the expected number of times that each suboptimal arm has been played up to time T. We show that when the total number of times that arms in $\mathcal{B}^*_{i_{\min}}$ have been played is greater than $\Omega(\log T)$, the class index $H_{i_{\min}}(t)$ will not be chosen with high probability. Besides, if each suboptimal arm $j \in \mathcal{B}^*_{i_{\max}}$ has been played more than $\Omega(\log T)$ times, the arm index $L_j(t)$ will not be chosen with high probability. The following theorem provides the performance guarantee for LSDT-CSI.

Theorem 2 *Suppose that $\mathcal{G}^*_\epsilon$ is connected. Assume that the reward distribution for each arm is sub-Gaussian with parameter $\sigma = 1$ [2]. Then the regret of LSDT-CSI up to time T is upper bounded as follows:*

$$R(T) \leq \left(\frac{32 \max_{i \in \mathcal{B}_{i_{\min}}} \Delta_i}{(\min_{j \in \mathcal{B}^*_{i_{\min}}} \Delta_j - \max_{k \in \mathcal{B}^*_{i_{\max}}} \Delta_k)^2} + \sum_{i \in \mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}} \frac{32}{\Delta_i} \right) \log T + O(|\mathcal{B}^*|), \tag{2.9}$$

where $\mathcal{A}$ is the set of arms with largest mean rewards ($i_{\max} \in \mathcal{A}$).

Proof 2 *See Appendix A.2.*

[2] See Sec. 2.6 for extensions to general σ.

Remark 1 *For fixed Δ_i, the regret of LSDT-CSI is of order*

$$O\Big((1 + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) \log T\Big), \tag{2.10}$$

as $T \to \infty$. In certain scenarios (e.g., $\mathcal{G}^*_\epsilon$ is a line graph), $|\mathcal{B}^*_{i_{\max}}| \ll K$, which indicates a sublinear scaling of regret in terms of the number of arms given such side information.

Remark 2 *If $\mathcal{G}^*_\epsilon$ is fully connected (e.g., ϵ is large), then $\mathcal{B}^*_{i_{\max}} = \mathcal{B}^*_{i_{\min}} = \mathcal{V}$. In this case, LSDT-CSI degenerates to the classic UCB policy and $R(T) \sim O(K \log T)$.*

We discuss in Sec. 2.7 that if the mean reward of each arm is independently and uniformly chosen from $[0, 1]$ and ϵ is bounded away from 0 and 1, the expected value of $|\mathcal{B}^*|$ is smaller than $O(K^{1/2} \log K)$, which indicates a sublinear scaling of regret in terms of the size of the action space. We also use a numerical example to verify the result in Sec. 2.7.

To establish the order optimality of LSDT-CSI, we further derive a matching lower bound on regret. We focus here on the case that the unknown mean reward of each arm is unbounded (i.e., can be any value on the real line). We adopt the same parametric setting as in [46] on classic MAB where the rewards are drawn from a specific parametric family of distributions with known distribution type. It should be clarified that although the upper bound on regret of LSDT-CSI is derived under the non-parametric setting (the distribution type is unknown), the non-parametric lower bound suffices to show the order optimality of LSDT-CSI since it is no smaller than that in the parametric one.

Specifically, we assume that the reward distribution of arm i has a univariate density function $f(\cdot; \theta_i)$ with an unknown parameter θ_i from a set of parameters Θ. Let $I(\theta||\lambda)$ be the Kullback-Leibler (KL) distance between two distributions

with density functions $f(\cdot;\theta)$ and $f(\cdot;\lambda)$ and with means $\mu(\theta)$ and $\mu(\lambda)$ respectively. We assume the same regularity assumptions on the finiteness of the KL divergence and its continuity with respect to the mean values as in [46].

Assumption 1 *For every $f(\cdot;\theta), f(\cdot;\lambda)$ such that $\mu(\lambda) > \mu(\theta)$, we have $0 < I(\theta||\lambda) < \infty$.*

Assumption 2 *For every $\epsilon > 0$ and $\theta, \lambda \in \Theta$ with $\mu(\lambda) > \mu(\theta)$, there exists $\eta > 0$ for which $|I(\theta||\lambda) - I(\theta||\rho)| < \epsilon$ whenever $\mu(\lambda) < \mu(\rho) < \mu(\lambda) + \eta$, $\rho \in \Theta$.*

Note that the regret measure studied in this chapter is problem-specific. There exist an issue of trivial lower bounds on regret caused by policies that heavily bias toward specific arms. For example, a policy that always plays arm 1 incurs 0 regret if arm 1 is indeed optimal in certain given problem instances. To avoid such trivial lower bounds, we focus on the set of uniformly good policies as did in [46]. A policy π is called uniformly good if for every $\boldsymbol{f}$, the regret of π satisfies $R(T) = o(T^\alpha), \forall \alpha > 0$, as $T \to \infty$. We then establish the lower bound on regret in the following theorem.

Theorem 3 *Suppose $\mathcal{G}_\epsilon^*$ is connected. Assume that Assumptions 1, 2 hold and the mean reward of each arm can be any value in $\mathbb{R}$. For any uniformly good policy, the regret up to time T is lower bounded as follows:*

$$\lim_{T\to\infty} \frac{R(T)}{\log T} \geq C_1, \tag{2.11}$$

where C_1 is the optimal value of an LP that only depends on $f_1, ..., f_K$ and ϵ (see (A.32) in Appendix A.3 for details). It can be shown that for fixed Δ_i, $I(\theta_i||\theta_i')$ and $I(\theta_i||\theta_{i_{\max}})$, the regret for any uniformly good policy is of order

$$\Omega\Big((1 + |\mathcal{B}_{i_{\max}}^* \setminus \mathcal{A}|) \log T\Big),$$

as $T \to \infty$.

Proof 3 *See Appendix A.3.*

Remark 3 *LSDT-CSI is order optimal since its upper bound on regret matches the lower bound shown in Theorem 3.*

Remark 4 *If there is a unique optimal arm, i.e.,* $|\mathcal{A}| = 1$, $R(T) \sim \Theta\Big(|\mathcal{B}^*_{i_{\max}}| \log T\Big)$, *as* $T \to \infty$.

2.5 Partial Side Information

In this section, we consider the general case of partial side information where the UIG $\mathcal{G}^*_\epsilon$ is partially revealed. We develop a learning policy: *LSDT-PSI (Learning from Similarity-Dissimilarity Topology with Partial Side Information)* following the two-step structure proposed in Sec. 2.3 and provide theoretical analysis on the regret performance.

2.5.1 Offline Elimination

A partially revealed UIG can be represented by an undirected edge-labeled multigraph $\mathcal{G}_\epsilon = (\mathcal{V}, \mathcal{E}^S_\epsilon, \mathcal{E}^D_\epsilon)$ (see Fig. 2.3). Specifically, $\mathcal{G}_\epsilon$ consists of two types of edges: type-S edges ($\mathcal{E}^S_\epsilon$) and type-D edges ($\mathcal{E}^D_\epsilon$) indicating the presence and the absence of the corresponding UIG edges. The absence of an edge between two nodes indicates an unknown relation between the two arms.

We first show that finding the candidate set under partial side information $\mathcal{E}^S_\epsilon, \mathcal{E}^D_\epsilon$ is NP-complete. We notice that finding the candidate set is equivalent to considering every node i individually and deciding if i can be a left

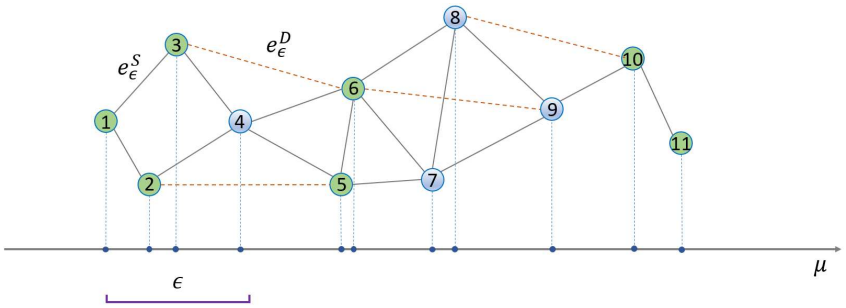

Figure 2.3: Partially revealed UIG as an undirected edge-labeled multigraph: black solid lines represent type-S edges and red dash lines represent type-D edges.

anchor of a UIG $\mathcal{G}'_\epsilon = (\mathcal{V}, \mathcal{E}^P_\epsilon)$ consisting of the same set of nodes with $\mathcal{G}_\epsilon$ and the potential edge set $\mathcal{E}^P_\epsilon$ satisfying

$$\mathcal{E}^S_\epsilon \subseteq \mathcal{E}^P_\epsilon, \tag{2.12}$$

$$\mathcal{E}^P_\epsilon \cap \mathcal{E}^D_\epsilon = \emptyset. \tag{2.13}$$

For example in Fig. 2.3, if we take $\epsilon = 0.15$, there exists a graph conforming assignment of mean rewards $\boldsymbol{\mu} = (0.8, 0.8, 0.8, 0.9, 1, 1, 0.9, 0.9, 0.8, 0.7, 0.6)$ of which the resulting UIG satisfies (2.12) and (2.13). In this assignment, node 5 and 6 are left anchors and thus belong to the candidate set. However, finding the candidate set is difficult in general. Specifically, we show the NP-completeness of the following decision problem.

LEFTANCHOR

[INPUT]: A multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_1, \mathcal{E}_2)$ knowing that there exists a UIG $\mathcal{G}' = (\mathcal{V}, \mathcal{E}_3)$ where $\mathcal{E}_1 \subseteq \mathcal{E}_3$ and $\mathcal{E}_3 \cap \mathcal{E}_2 = \emptyset$, and a specific node i.

[QUESTION]: Does there exist a UIG $\mathcal{G}'' = (\mathcal{V}, \mathcal{E}_4)$ where $\mathcal{E}_1 \subseteq \mathcal{E}_4$ and $\mathcal{E}_4 \cap \mathcal{E}_2 = \emptyset$ such that node i is a left anchor of $\mathcal{G}''$?

Theorem 4 *LEFTANCHOR is NP-complete.*

Proof 4 *To show the NP-completeness of LEFTANCHOR, we give a reduction from a variant of the 3-SAT problem: CONSISTENT-NAE-3SAT.*

> *CONSISTENT-NAE-3SAT*
>
> *[INPUT]: A not-all-equal satisfiable 3-SAT instance: there exists a truth assignment such that every clause contains one or two true literals .*
>
> *[QUESTION]: Does there exist a consistent truth assignment, i.e., every clause contains only one true literal OR every clause contains only two true literals?*

The NP-completeness of CONSISTENT-NAE-3SAT is proved in Appendix A.4. The reduction to LEFTANCHOR and the remaining proof are presented in Appendix A.5.

It should be noted that LEFTANCHOR is similar to the so-called UIG Sandwich Problem [37] where two graphs $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2)$ are given satisfying $\mathcal{E}_1 \subseteq \mathcal{E}_2$. The question is whether a UIG $\mathcal{G}_3 = (\mathcal{V}, \mathcal{E}_3)$ exists satisfying $\mathcal{E}_1 \subseteq \mathcal{E}_3 \subseteq \mathcal{E}_2$. It is not difficult to see that the type-S edge set $\mathcal{E}_\epsilon^S$ corresponds to $\mathcal{E}_1$ in the sandwich problem and the complement of $\mathcal{E}_\epsilon^D$ corresponds to $\mathcal{E}_2$. However, LEFTANCHOR is different from the sandwich problem as we know that the sandwich problem is satisfied by the ground truth UIG $\mathcal{G}_\epsilon^*$, and what we are interested in is whether a specific node i can be a left anchor.

To address the challenge of finding the candidate set in polynomial time, we exploit the following topological property of $\mathcal{G}_\epsilon$ to obtain an approximation solution.

Proposition 1 *Given $\mathcal{G}_\epsilon$, an arm i is sub-optimal if it is similar to two dissimilar arms, i.e., if there exist j, k, such that $(i, j), (i, k) \in \mathcal{E}_\epsilon^S$ but $(j, k) \in \mathcal{E}_\epsilon^D$, then $i \notin \mathcal{B}$.*

Based on this property, we develop the offline elimination step of LSDT-PSI with $O(K|\mathcal{E}_\epsilon^D|)$ complexity in Algorithm 3.

Algorithm 3: **Offline Elimination of LSDT-PSI**

Input: $\mathcal{G}_\epsilon = (\mathcal{V}, \mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D)$.

Output: $\mathcal{B}_0$.

Initialization: $\mathcal{B}_0 = \mathcal{V}$.

for $i = 1, 2, ..., K$ **do**

$\mathcal{B}_0 \leftarrow \mathcal{B}_0 \setminus \{i\}$ if there exist $j, k \in \mathcal{V}$ such that

$$(i, j), (i, k) \in \mathcal{E}_\epsilon^S, (j, k) \in \mathcal{E}_\epsilon^D.$$

It is clear that in general, $\mathcal{B}^* \subseteq \mathcal{B} \subseteq \mathcal{B}_0$. However, in certain scenarios, the partially revealed UIG provides sufficient topological information to identify the ground truth candidate set $\mathcal{B}^*$ obtained from the fully revealed UIG. We show that such information is fully exploited by the offline elimination step of LSDT-PSI to achieve the same performance as that of LSDT-CSI for the case of complete side information.

Specifically, we make the following assumptions on $\mathcal{G}_\epsilon^*$ and its equivalence classes $\{\mathcal{B}_i^*\}_{i=1}^m$ assuming that the neighbor set of every arm $i \notin \mathcal{B}^*$ is diverse enough. Without loss of generality, we assume an increasing order of the equivalence classes along the real line, i.e., $\forall 1 \leq i < j \leq m$ and $\forall k_i \in \mathcal{B}_i^*, k_j \in \mathcal{B}_j^*$, we have $\mu_{k_i} < \mu_{k_j}$. Note that $\mathcal{B}^* = \mathcal{B}_1^* \cup \mathcal{B}_m^*$.

Assumption 3 *For every $1 < i < m$, assume that there exist j, k such that $j < i < k$ and $\mathcal{B}_j^*, \mathcal{B}_k^*$ are connected to $\mathcal{B}_i^*$ but mutually disconnected in $\mathcal{G}_\epsilon^*$.*[3]

Assumption 4 *Assume that there exists a constant $\kappa > 0$ and for every i, $|\mathcal{B}_i^*| \geq \kappa \log K$.*

We further make a probabilistic assumption on the partial side information.

Assumption 5 *The presence and the absence of an edge in the UIG $\mathcal{G}_\epsilon^*$ are revealed by the partial side information $\mathcal{E}_\epsilon^S$ and $\mathcal{E}_\epsilon^D$ independently with probabilities p_S and p_D. Assume that $p_S^2 p_D \geq 1 - e^{-2/\kappa}$, where κ is defined in Assumption 4.*

Note that as κ increases, for every arm $i \notin \mathcal{B}^*$, the number of dissimilar arm pairs that are similar to i increases. Therefore, smaller probabilities of observing edges can still guarantee that arm i is elilminated with high probability.

Based on these assumptions, we provide performance guarantee for the offline elimination step of LSDT-PSI through the following theorem. We also verify the results through numerical examples in Sec. 2.7.

Theorem 5 *Given a UIG $\mathcal{G}_\epsilon^*$, under Assumptions 3-5, with probability at least $1 - \frac{1}{K^2}$, every arm $i \notin \mathcal{B}^*$ is eliminated by the offline elimination step of LSDT-PSI and thus,*

$$\mathbb{E}_{\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D}\left[|\mathcal{B}_0|\right] = |\mathcal{B}^*| + o(1), \tag{2.14}$$

as $K \to \infty$, where $\mathcal{B}_0$ is the arm set remaining after the offline elimination step of LSDT-PSI.

Proof 5 *See Appendix A.6.*

[3]Two equivalence classes are connected if and only if at least one pair of arms from the two classes are adjacent in the UIG. It can be inferred from the equivalence relation that if there exists an adjacent arm pair from the two classes, all arm pairs are adjacent.

2.5.2 Online Aggregation

Now we present the second step, the online learning procedure of LSDT-PSI. We first define a similarity graph $\mathcal{G}'_\epsilon = (\mathcal{V}', \mathcal{E}_\epsilon^{S\prime})$ restricted to the remaining arm set $\mathcal{B}_0$: $\mathcal{V}' = \mathcal{B}_0$ and $\mathcal{E}_\epsilon^{S\prime} = \{(i,j)|i,j \in \mathcal{B}_0, (i,j) \in \mathcal{E}_\epsilon^S\}$. For every arm $i \in \mathcal{B}_0$, we define an exploration value $z_i \in [0,1]$, which measures the topological significance of node i in the similarity graph $\mathcal{G}'_\epsilon$ and determines the frequency of playing arm i. Intuitively, a node with a higher degree has a higher exploration value since playing this node provides information about more (neighboring) nodes. Specifically, we define exploration values $\{z_i\}_{i\in\mathcal{B}_0}$ as the optimal solution to the following LP.

$$\mathcal{P}_2: \quad C_2 = \min_{\{z_i\}_{i\in\mathcal{V}'}} \sum_{i\in\mathcal{V}'} z_i, \tag{2.15}$$

$$s.t. \quad \sum_{j\in\mathcal{N}'[i]} z_j \geq 1, \ \forall i \in \mathcal{V}', \tag{2.16}$$

$$z_i \geq 0, \ \forall i \in \mathcal{V}', \tag{2.17}$$

where $\mathcal{N}'[i]$ is the set of neighbors of node i in $\mathcal{G}'_\epsilon$ (including i). In the online learning procedure, the number of times arm i is played is proportional to its exploration value z_i. Note that if at least n_i plays are necessary to distinguish a suboptimal arm i from the optimal one in the classic MAB problem, now if suffices to play only $z_i n_i$ times by aggregating observations from every neighboring arm $j \in \mathcal{N}'[i]$ that is played $z_j n_i$ times. Note that $z_i \leq 1, \forall i$ and C_2 is upper bounded by the size of the minimum dominating set of $\mathcal{G}_\epsilon{}'$.

We briefly summarize the second step of LSDT-PSI: the algorithm is played in epochs and during epoch m, arms are played up to $\tau_i(m) \sim \Theta(z_i \log T)$ times. Arms less likely to be optimal are eliminated at the end of every epoch and only two types of arms will be played in the next epoch: 1) non-eliminated arms and 2) arms with non-eliminated neighbors. After a sufficient num-

ber of epochs, only arms close to the optimal one remain and we use single arm indices for selection. Let $\bar{x}_i(m)$ be the average reward from arm i up to epoch m.

Algorithm 4: **Online Aggregation of LSDT-PSI**

Input: $\mathcal{G}'_\epsilon = (\mathcal{V}', \mathcal{E}_\epsilon^{S\prime})$, time horizon T, parameter $\lambda > 0$.

Initialization: Let $\tilde{\Delta}_0 = 1$, $\mathcal{S}_0 = \mathcal{B}_0$, $\{z_i\}_{i\in\mathcal{V}'}$ be the solution to $\mathcal{P}_2$, $m_f = \min\left\{\left\lceil \log_2\left(\frac{8}{\sqrt{2\lambda\epsilon}}\right)\right\rceil, \left\lfloor \frac{1}{2}\log_2\frac{T}{e}\right\rfloor\right\}$.

for $m = 0, 1, ..., m_f$ **do**

if $|\mathcal{B}_m| = 1$ **then** Play $i \in \mathcal{B}_m$ until time T.

else

for each arm $i \in \mathcal{S}_m$ **do**

Play arm i until $\tau_i(m) = \left\lceil \frac{\lambda z_i \log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2}\right\rceil$.

Let $\mathcal{B}_{m+1} = \mathcal{B}_m$.

for each arm $i \in \mathcal{B}_m$ **do**

$\mathcal{B}_{m+1} \leftarrow \mathcal{B}_{m+1} \setminus \{i\}$ if

$$\frac{\sum_{j\in\mathcal{N}'[i]}\bar{x}_j(m)\tau_j(m)}{\sum_{j\in\mathcal{N}'[i]}\tau_j(m)} + \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2\sum_{j\in\mathcal{N}'[i]}\tau_j(m)}} + \epsilon \leq \max_{k\in\mathcal{B}_m}\left\{\frac{\sum_{j\in\mathcal{N}'[k]}\bar{x}_j(m)\tau_j(m)}{\sum_{j\in\mathcal{N}'[k]}\tau_j(m)} - \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2\sum_{j\in\mathcal{N}'[k]}\tau_j(m)}}\right\}. \tag{2.18}$$

Let $\mathcal{S}_{m+1} = \{i : \mathcal{N}'[i] \cap \mathcal{B}_{m+1} \neq \emptyset\}$.

Let $\tilde{\Delta}_{m+1} = \tilde{\Delta}_m/2$.

for $t = \sum_{i\in\mathcal{V}'}\tau_i(m_f) + 1, ..., T$ **do**

Play arm $i_t^* = \text{argmax}_{i\in\mathcal{B}_{m_f+1}}\bar{x}_i(t-1) + \sqrt{\frac{2\log(t-1)}{\tau_i(t-1)}}$.

2.5.3 Order Optimality

The following theorem provides an upper bound on regret of LSDT-PSI for any given partially revealed UIG.

Theorem 6 *Given a partially revealed UIG $\mathcal{G}_\epsilon$. Assume that the reward distribution of reach arm is $\sigma = 1/2$ sub-Gaussian*[4]*. Let $\mathcal{Q} = \{i \in \mathcal{B}_0 : \Delta_i > 4\epsilon\}$. Then the regret of LSDT-PSI up to time T is upper bounded by:*

$$R(T) \leq \sum_{j \in \mathcal{B}_0 \setminus (\mathcal{Q} \cup \mathcal{A})} \Delta_j \max\left\{\frac{8\log T}{\Delta_j^2}, \frac{32 z_j \log(T\epsilon^2)}{\epsilon^2}\right\} + \sum_{i \in \mathcal{Q}} \Delta_i z_i \frac{32\log(T\hat{\Delta}_i^2)}{\hat{\Delta}_i^2} + O(|\mathcal{V}'|), \tag{2.19}$$

where $\hat{\Delta}_i = \max\{\min_{j \in \mathcal{N}'[i]} \Delta_j - 3\epsilon, \epsilon\}$.

Proof 6 *See Appendix A.7.*

Remark 5 *For fixed Δ_i, the regret of LSDT-PSI is of order*

$$O\Big((\gamma(\mathcal{G}'_\epsilon) + |\mathcal{B}_0 \setminus (\mathcal{Q} \cup \mathcal{A})|)\log T\Big), \tag{2.20}$$

as $T \to \infty$, where $\gamma(\mathcal{G}'_\epsilon)$ is the size of the minimum dominating set of graph $\mathcal{G}'_\epsilon$ and $|\mathcal{B}_0 \setminus (\mathcal{Q} \cup \mathcal{A})|$ is the number of sub-optimal arms that are 4ϵ-close to the optimal one. It is not difficult to see that as ϵ increases, $\gamma(\mathcal{G}'_\epsilon)$ decreases and $|\mathcal{B}_0 \setminus (\mathcal{Q} \cup \mathcal{A})|$ increases. For an appropriate ϵ, a sublinear scaling of regret in the number of arms can be achieved.

Recall in Theorem 5, we show that under certain assumptions, the offline elimination step of LSDT-PSI achieves the same performance as LSDT-CSI for the case of complete side information. The following corollary further establishes the order optimality of LSDT-PSI in terms of both K and T.

[4]Certain sub-Gaussian distributions (e.g. Bernoulli distribution, uniform distribution on $[0, 1]$) have parameters $\sigma = 1/2$. See Sec. 2.6 for extensions to general σ.

Corollary 1 *Assume that Assumptions 3-5 hold and* $\Delta_i > 4\epsilon, \forall i \in \mathcal{B}^*_{i_{\min}}$. *For fixed* Δ_i, p_S, p_D, *the expectation of regret of LSDT-PSI taken over random realizations of the partial side information* $\mathcal{E}^S_\epsilon, \mathcal{E}^D_\epsilon$ *is upper bounded as follows:*

$$\mathbb{E}_{\mathcal{E}^S_\epsilon, \mathcal{E}^D_\epsilon}[R(T)] \leq O\Big((1 + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) \log T\Big), \tag{2.21}$$

as $T \to \infty$, *which matches the lower bound on regret for the case of complete side information established in Theorem 3.*

Proof 7 *See Appendix A.8.*

2.6 Extensions

In this section, we discuss extensions of the proposed policies: LSDT-CSI and LSDT-PSI as well as their regret analysis to cases with disconnected UIGs and other reward distributions. We also discuss the extension of applying Thompson Sampling techniques to the LSDT learning structure.

2.6.1 Extensions to disconnected UIG

Suppose that the UIG $\mathcal{G}^*_\epsilon$ has M ($M > 1$) connected components. It is not difficult to see that every connected component of $\mathcal{G}^*_\epsilon$ is still a UIG and the set of left anchors of $\mathcal{G}^*_\epsilon$ is the union of left anchors of all components. Therefore, in the case of complete side information, the offline elimination step of LSDT-CSI outputs at most $2M$ equivalences classes and the second step of LSDT-CSI can be directly applied by maintaining a class index for every equivalence class as defined in

(2.8). Moreover, by extending the regret analysis of LSDT-CSI in Theorem 2 as well as the lower bound on regret for uniformly good policies in Theorem 3 to the disconnected case, we can show that LSDT-CSI achieves an order optimal regret, i.e.,

$$R(T) \sim \Theta\Big((M + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) \log T\Big), \tag{2.22}$$

as $T \to \infty$. In the extreme case when $M = K$ (e.g., $\epsilon \to 0$), LSDT-CSI degenerates to the classic UCB policy and $R(T) \sim \Theta(K \log T)$.

In the case of partial side information, the LSDT-PSI policy along with its regret analysis applies to any partially revealed UIG without assumptions on the connectivity of the graph. The upper bound on regret in Theorem 6 still holds when $\mathcal{G}^*_\epsilon$ has M connected components. In the extreme case where $M = K$, the size of the minimum dominating set of the similarity graph $\mathcal{G}'_\epsilon$ equals K and thus, $R(T) \sim O(K \log T)$.

To show the order optimality of LSDT-PSI in the disconnected case, we need certain modifications on the assumptions of the UIG. We consider every connected component m of $\mathcal{G}^*_\epsilon$ with ℓ equivalence classes $\{\mathcal{B}^{*(m)}_i\}^\ell_{i=1}$. We assume that Assumptions 3 and 4 hold for every connected component and without loss of generality, we assume that the optimal arm $i_{\max}$ is in component $m = 1$. Then under Assumption 5, we can extend the regret analysis in Corollary 1 to the case where $\mathcal{G}^*_\epsilon$ has M connected components. It can be shown that the expected regret of LSDT-PSI is upper bounded by

$$O\Big((M + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) \log T\Big), \tag{2.23}$$

as $T \to \infty$, which matches the lower bound in the case of complete side information.

2.6.2 Extensions to Other Distributions

Recall that in the regret analysis of LSDT-CSI and LSDT-PSI, we assume sub-Gaussian reward distributions with parameter $\sigma = 1$ (e.g., standard normal distribution) or $\sigma = 1/2$ (e.g., Bernoulli distribution). We first discuss extensions to general sub-Gaussian distributions with arbitrary parameters σ.

In LSDT-CSI, by replacing the second terms of the UCB indices defined in (2.7) and (2.8) by $\sqrt{\frac{\alpha \log t}{\tau_j(t)}}$ and $\sqrt{\frac{\alpha \log t}{\sum_{j \in \mathcal{B}_i^*} \tau_j(t)}}$ where α is an input parameter, the regret analysis in Theorem 2 still applies and the upper bound on regret is only affected up to a constant scaling factor, as long as $\alpha > 6\sigma^2$. A similar extension also applies to LSDT-PSI if we change the second terms of the UCB indicies in (2.18) to $\sqrt{\frac{\beta \log(T\tilde{\Delta}_m^2)}{\sum_{j \in \mathcal{N}'[i]} \tau_j(m)}}$ where $\beta \geq 2\sigma^2$.

Furthermore, we can extend the results for sub-Gaussian reward distributions to other distribution types such as light-tailed and heavy-tailed distributions. There are standard techniques for such extensions by replacing the concentration result with the corresponding ones for light-tailed and heavy-tailed distributions (the latter also requires replacing sample means with truncated sample means). Similar extensions for classic MAB problems without side information are discussed in [62, 61]. To illuminate the main ideas without too much technicality, most existing work assumes an even stronger assumption of bounded support in $[0, 1]$ (see [9],[34], [47], etc.).

2.6.3 Extensions to Thompson Sampling Techniques

The two-step learning structure LSDT is in general independent of the specific arm selection rule adopted in the online learning step. We discuss here how Thompson Sampling (TS) techniques can be extended and incorporated into the basic structure with aggregation of reward observations.

Specifically, in the case of complete side information, after reducing the action space to the candidate set via the offline step, we adopt a similar hierarchical online learning policy as that used in LSDT-CSI by maintaining two posterior distributions on the reward parameters, one at the equivalence class level, the other at the arm level. At each time, the policy first randomly selects an equivalence class according to its class-level probability of containing the optimal arm and then randomly draws an arm within the class according to its arm-level probability of being optimal.

In the case of partial side information, similar to LSDT-PSI, an eliminative strategy is carried out to sequentially eliminate arms less likely to be optimal. At each time, an arm is randomly drawn according its arm-level posterior distribution of being optimal. The observation from the selected arm is also used to update higher level posterior distributions of its neighbors, which aggregate observations from all similar arms. According to the high level posterior distribution, the arm that is least likely to be optimal is eliminated if it has been explored for sufficient times.

Simulation results in Sec. 2.7.4 show a similar performance gain by exploiting the side information on arm similarity and dissimilarity through the two-step learning structure when TS is incorporated in both cases. To achieve a full

exploitation of the side information and establish the order optimality on regret, however, further studies are required.

2.7 Numerical Examples

In this section, we illustrate the advantages of our policies through numerical examples on both synthesized data and a real dataset in recommendation systems. All the experiments are run 100 times using a Monte Carlo method on MATLAB R2014b.

2.7.1 Reduction of the action space

Complete Side Information

We use two experiments to show how much the action space can be reduced by exploiting the complete side information. In the first experiment, we fix $K = 100$ arms with mean rewards uniformly chosen from $(0, 1)$ and let ϵ vary from 0 to 1. For every ϵ, we obtain a UIG $\mathcal{G}_\epsilon^*$. We apply the offline elimination step of LSDT-CSI to $\mathcal{G}_\epsilon^*$ and compare the size of the candidate set $\mathcal{B}^*$ with K. In the second experiment, we fix $\epsilon = 0.2$ and let K increase from 10 to 200. We generate arms and UIGs in the same way as in the first experiment. We show how $|\mathcal{B}^*|/K$ varies as K increases. The results are shown in Figs. 2.4 and 2.5.

As we can see from Fig. 2.4, when ϵ is small ($\epsilon < 0.1$), the graph is disconnected. As ϵ increases, the number of connected components decreases and thus, $|\mathcal{B}^*|$ decreases. When the graph is connected ($\epsilon > 0.1$), the candidate set $\mathcal{B}^*$

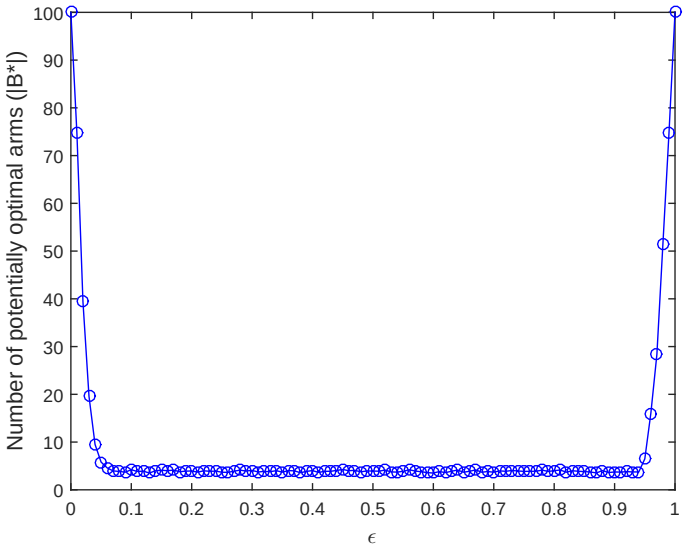

Figure 2.4: Reduction of the action space with complete side information: $|\mathcal{B}^*|$ v.s. ϵ.

only contains two equivalence classes and thus $|\mathcal{B}^*|$ is much smaller than K. When ϵ is large ($\epsilon > 0.9$), the probability that the graph is complete increases as ϵ increases. In this case, the candidate set contains all the arms. Thus, $|B^*|$ increases to K as ϵ grows to 1. In Fig. 2.5, we notice that $\mathcal{B}^*$ has a diminishing cardinality compared with K. Since the mean rewards are uniformly chosen from $(0, 1)$, the set of arms becomes denser on the interval $(0, 1)$ as K grows. It can be inferred from [43] that the maximum distance d between two consecutive points uniformly chosen from $(0, 1)$ is in the order of $O(\frac{\log K}{\sqrt{K}})$ with probability $1 - 1/K$. If we choose $\epsilon = \frac{\rho \log K}{\sqrt{K}}$ for some $\rho > 0$, $\mathcal{G}_\epsilon$ will be connected with high probability. Moreover, it can be shown that the cardinality of $\mathcal{B}^*_{i_{\max}}$ ($\mathcal{B}^*_{i_{\min}}$) is smaller than the number of nodes whose distance to $i_{\max}$ ($i_{\min}$) is smaller than d. Therefore, it follows that the cardinality of the candidate set in this setting is smaller than $O(K^{1/2} \log K)$.

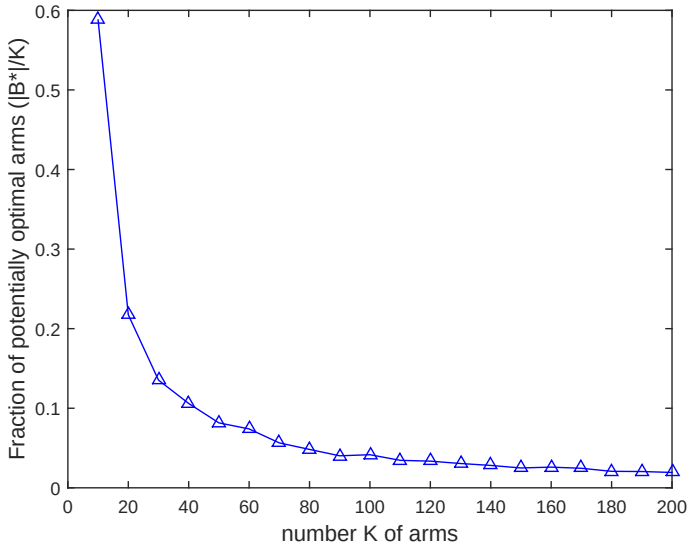

Figure 2.5: Reduction of the action space with complete side information: $\frac{|\mathcal{B}^*|}{K}$ v.s. K.

Partial Side Information

We use two other experiments to show the reduction of the action space with partial side information. In the first experiment, we fix $K = 100$ arms with mean rewards uniformly chosen from $(0, 1)$. We choose $\epsilon = 0.2$ and obtain the UIG $\mathcal{G}_\epsilon^*$. We let $p_S = p_D = p$ vary from 0.1 to 1 and for every p, we observe the presence and the absence of edges in $\mathcal{G}_\epsilon^*$ independently with probability p. We apply the offline elimination step of LSDT-PSI on $\mathcal{G}_\epsilon$ and compare the size of the output set $\mathcal{B}_0$ with K. Note that when $p = 1$, $\mathcal{G}_\epsilon^*$ is fully revealed and we use the offline elimination step of LSDT-CSI to obtain $\mathcal{B}^*$. In the second experiment, we fix $\epsilon = 0.2, p_S = p_D = 0.5$ and let K increase from 10 to 150. We generate arms and side information graphs in the same way as in the first experiment and show how $|\mathcal{B}_0|/K$ varies as K increases. The results of the two experiments are shown in Figs. 2.6 and 2.7 .

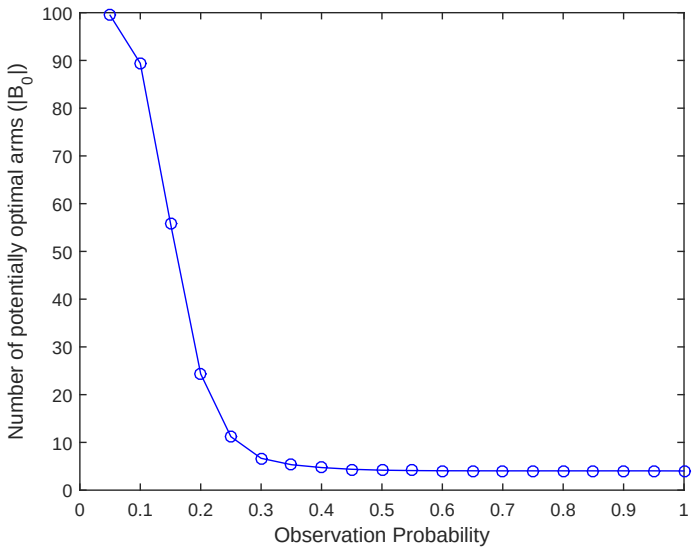

Figure 2.6: Reduction of the action space with partial side information: $|\mathcal{B}_0|$ v.s. p.

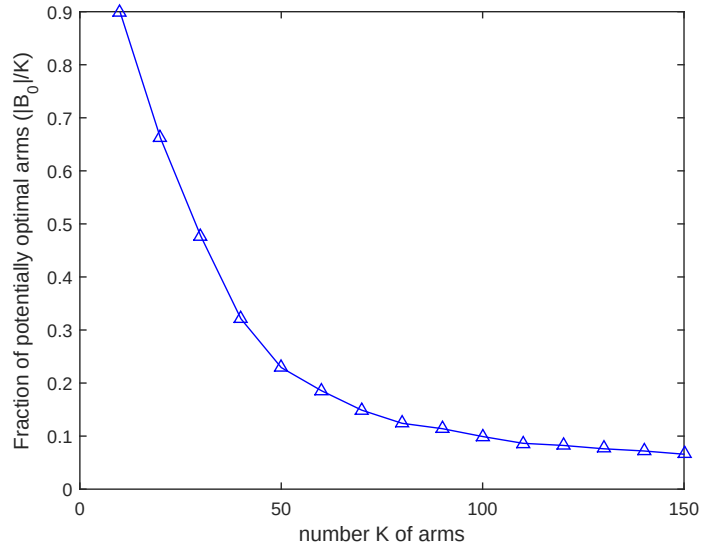

Figure 2.7: Reduction of the action space with partial side information: $\frac{|\mathcal{B}_0|}{K}$ v.s. K.

It can be seen from Fig. 2.6 that as p increases, $|\mathcal{B}_0|$ decreases to $|\mathcal{B}^*|$. Besides, when $p > 0.5$, the performance of the offline elimination step of LSDT-PSI is as good as that of LSDT-CSI, which is optimal, i.e. only arms in $\mathcal{B}^*$ remain. Moreover, in Fig. 2.7, we see that $|\mathcal{B}_0|/K$ decreases as K increases which indicates a diminishing cardinality of the reduced action space in terms of K.

2.7.2 Regret on Randomly Generated Graphs

Complete Side Information

We compare LSDT-CSI with existing algorithms on a set of randomly generated arms. We obtain the UIG $\mathcal{G}_\epsilon^*$ on $K = 100$ nodes with means uniformly chosen from $[0.1, 1]$ and $\epsilon = 0.1$. Every time an arm i is played, a random reward is drawn independently from a Gaussian distribution with mean μ_i and variance 1. We let T vary from 10 to 1000 and compare the regret of LSDT-CSI and four baseline algorithms:

1. UCB1: classic UCB policy proposed in [9] assuming no relation among arms.
2. TS: classic Thompson Sampling algorithm proposed in [60] assuming Beta prior and Bernoulli likelihood on the reward model.
3. CKL-UCB: proposed in [55] for Lipschitz bandit exploiting only similarity relations.
4. OSUB: proposed in [26] for unimodal bandits. Note that if the UIG $\mathcal{G}_\epsilon^*$ is connected, it satisfies the unimodal structure: for every sub-optimal arm$i$,

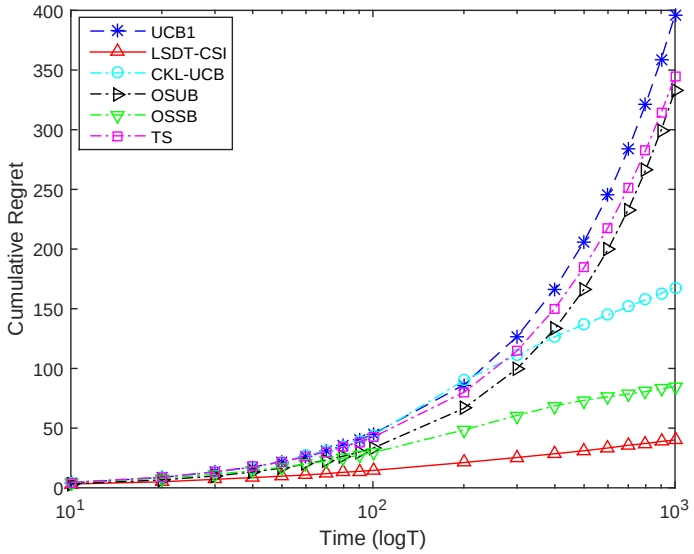

Figure 2.8: Regret performance on randomly generated arms with complete side information ($K = 100, \epsilon = 0.1$): comparison with existing algorithms.

there exists a path $P = (i_1 = i, i_2, ..., i_n = i_{\max})$ such that for every $t \in [1, n-1]$, $\mu_{i_t} \leq \mu_{i_{t+1}}$.

5. OSSB: proposed in [25] for general structured bandits. At each time, OSSB estimates the minimum number of times that every arm has to be played by solving a LP.

The results shown in Fig. 2.8 indicate that LSDT-CSI outperforms the baseline algorithms. In particular, when $T < K$, LSDT-CSI has already started to exploit the optimal arm while the other algorithms are still exploring. We also compare LSDT-CSI with a heuristic algorithm applying UCB1 on the candidate set in Fig. 2.9. With the same setup, we see performance gain due to the online step.

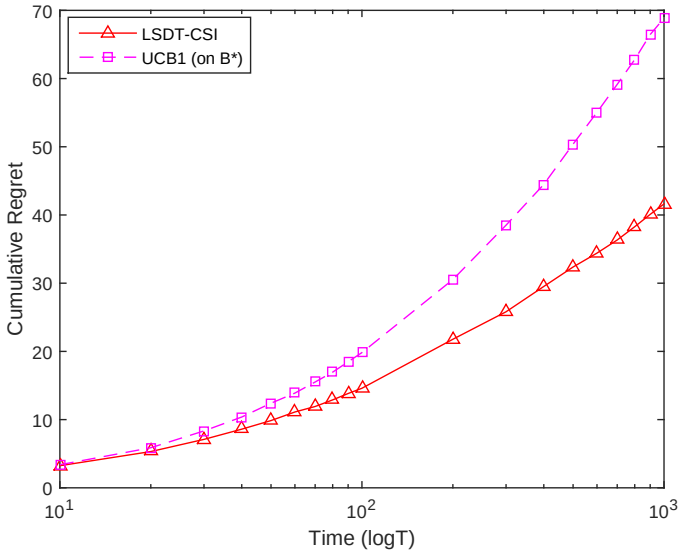

Figure 2.9: Regret performance on randomly generated arms with complete side information ($K = 100, \epsilon = 0.1$): comparison with a heuristic algorithm.

Partial Side Information

We compare LSDT-PSI with existing algorithms. We obtain the UIG $\mathcal{G}_\epsilon^*$ on $K = 100$ arms with means uniformly chosen from $[0.1, 0.9]$ and $\epsilon = 0.1$. We let $p_S = p_D = p = 0.5$ and get the partially observed UIG $\mathcal{G}_\epsilon$ based on Assumption 5. The random rewards for every arm i are independently generated from a Bernoulli distribution with mean μ_i. We consider $T \in [100, 1000]$.

Given that finding the candidate set is NP-complete, the OSSB policy is not applicable since the LP is unspecified. Besides, OSUB is also inapplicable since $\mathcal{G}_\epsilon$ is not unimodal in general. Therefore, we only compare LSDT-PSI with three baseline algorithms: UCB1, TS and CKL-UCB. In LSDT-PSI, we choose the input parameter $\lambda = 1/8$. Note that the choice of λ does not affect the theoretical upper bound on regret. However, in practice, it is better to use a smaller λ to avoid

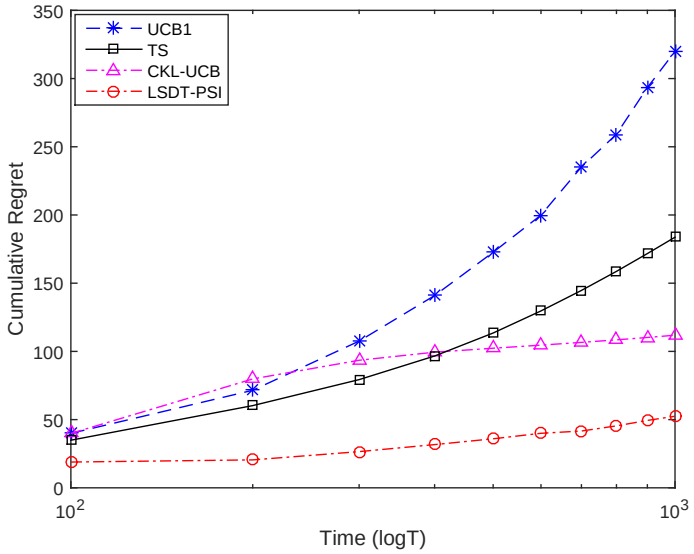

Figure 2.10: Regret performance on randomly generated arms with partial side information ($K = 100, \epsilon = 0.1, p = 0.5$): comparison with existing algorithms

excessive plays of suboptimal arms. The simulation results shown in Fig. 2.10 indicates that LSDT-PSI outperforms the other algorithms. Similar to the case of complete side information, we compare LSDT-PSI with a heuristic algorithm applying UCB1 on $\mathcal{B}_0$. A similar performance gain is observed in Fig. 2.11.

2.7.3 Online Recommendation Systems

In this subsection, we apply LSDT-PSI to a problem in online recommendation systems. We test our policy on a dataset from Jester, an online joke recommendation and rating system [36], consisting of 100 jokes and 25K users and every joke has been rated by at least 34% of the entire population.[5]. Ratings are real values between -10.00 and 10.00. In the experiment, we recommend a joke

[5]Available on http://eigentaste.berkeley.edu/dataset/.

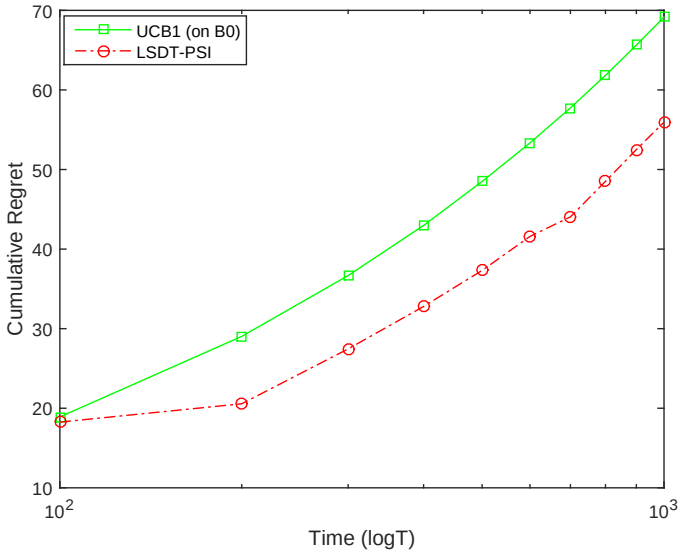

Figure 2.11: Regret performance on randomly generated arms with partial side information ($K = 100, \epsilon = 0.1, p = 0.5$): comparison with a heuristic algorithm

(modeled as an arm) to a new user at each time and observe the rating, which corresponds to playing an arm and receiving the reward. Note that although different users have different preference towards items, every item exhibits certain internal quality that is represented by the mean reward, i.e., the average rating from all users. The variations of ratings from different users correspond to the randomness of rewards. Notice that the algorithms we propose work for any reward distribution as long as it is sub-Gaussian, Jester is a suitable dataset for the purpose of evaluating the performance of our algorithms since any distribution with bounded support is sub-Gaussian. In accordance with the assumptions of the policy, all ratings are normalized to $[0, 1]$.

To test our policy using side information, we partition the dataset into a training set (5% or 10% of the users) and a test set (20K users). We obtain the partially revealed UIG from the training set as follows: we estimate the dis-

tance between two jokes i, j by calculating the difference between their average ratings from users in the training set who have rated both jokes. We define a confidence parameter $\alpha > 0$. If the distance between (i, j) is larger than $(1 + \alpha)\epsilon$, we add (i, j) to $\mathcal{E}_\epsilon^D$. Otherwise if the distance is smaller than $(1 - \alpha)\epsilon$, we add (i, j) to $\mathcal{E}_\epsilon^S$. It is clear that there exist certain pairs of arms whose relations are unknown. We let $\alpha = 0.2$ if the size of the training set is 2% of the entire dataset or $\alpha = 0.1$ if the size of the training set is 5%. Note that as the size of the training set increases, the estimation of distances between jokes becomes more accurate and thus, the confidence parameter can be smaller. As a consequence, the number of joke pairs whose relations are known increases. For the hyper-parameter ϵ, we use an iterative approach to find the best ϵ that minimize the size of $\mathcal{B}_0$, i.e., the set of arms that need to be explored. Intuitively, as ϵ increases, $|\mathcal{B}_0|$ first decreases since the side information graph becomes more connected and more similarity relations can be observed. Therefore, the probability of eliminating sub-optimal arms by the offine step becomes higher. When ϵ is large, the graph approaches a complete graph and less dissimilarity relations are observed. As a consequence, the probability of eliminating sub-optimal arms decreases and thus $|\mathcal{B}_0|$ increases. A similar tendency of variation can be observed on the overall regret performance of the learning policy. Based on this, the iterative approach starts from a small $\epsilon(0)$ (i.e., 0.01) at time $t = 0$ and find $\mathcal{B}_0(0)$. It keeps doubling the value of ϵ at each step until time t when $|\mathcal{B}_0(t)| > |\mathcal{B}_0(t-1)|$. Then a binary search method is applied to find the best ϵ^* (with a minimum increment of 0.01) between $\epsilon(t-1)$ and $\epsilon(t)$ that achieves the minimum $|\mathcal{B}_0|$.

We use an unbiased offline evaluation method introduced in [49] and [50] to evaluate algorithms including LSDT-PSI, UCB1, TS, CKL-UCB and UCB1 on $\mathcal{B}_0$, on the test set. Fig. 2.12 shows the average rating per user with con-

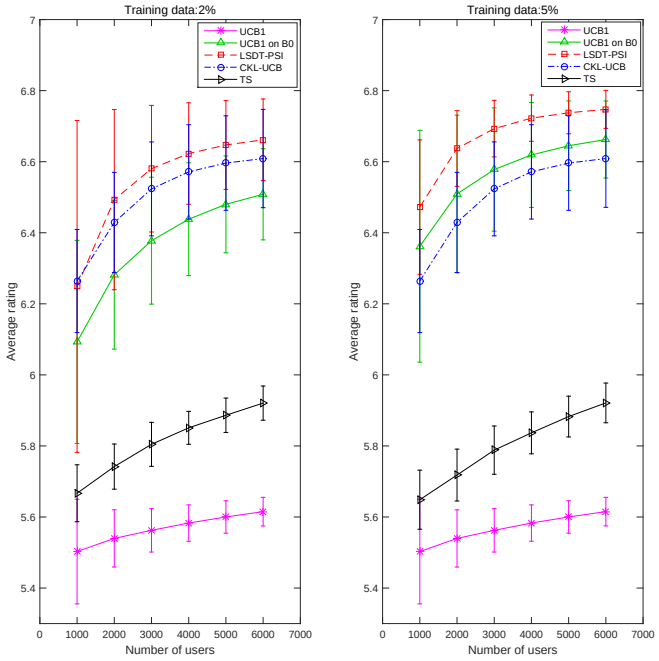

Figure 2.12: Joke recommendation on Jester.

fidence intervals (scaled back to $[0, 10.00]$) of every policy. Note that CKL-UCB needs to estimate the KL-divergence between two distributions. Since the distribution type in the real dataset is unknown, we can only use Δ^2 to approximate the KL-divergence where Δ is the distance between the average ratings. For LSDT-PSI, we choose the input parameter $\lambda = 1/32$. Simulation results in Fig. 2.12 show that LSDT-PSI has the best performance with relatively small variations. Besides, the effectiveness of the adaptive approach on selecting the hyper-parameter ϵ is verified. Moreover, it can be observed that as the size of the training data increases, the performance of LSDT-PSI and UCB1 on $\mathcal{B}_0$ get improved since more side information is available.

2.7.4 LSDT with Thompson Sampling Techniques

As discussed in Sec. 2.6.3, we use numerical examples to show the performance of applying TS techniques in the two-step learning structure: LSDT. We adopt the same experiment setup with that in the simulation of regret analysis on randomly generated graphs with complete side information (Sec. 2.7) and compare LSDT-TS (CSI) (applying TS in LSDT learning structure in the case of complete side information, which is introduced in Sec. 2.6.3) with classic TS that ignores side information.

The results in Fig. 2.13 verify the advantage of our two-step learning structure, which fully exploits the topological structure of the side information graph. Besides, we also compare LSDT-TS (CSI) with another heuristic algorithm, which simply applies classic TS on the reduced action space $\mathcal{B}^*$ without aggregation observations from similar arms in the second step of online learning. The results in Fig. 2.14 further indicates that the online aggregations step in the two-step learning structure improves the performance.

In the case of partial side information, we conduct an experiment similar with that in Sec. 2.7 to evaluate the performance of LSDT-TS (PSI), which applies TS in LSDT learning structure in the case of partial side information as discussed in Sec. 2.6. We compare LSDT-TS (PSI) with classic TS ignoring side information and another heuristic algorithm applying TS on the reduced action space without online aggregation. The results shown in Fig. 2.15 and Fig 2.16 verify the performance gain through both offline and online steps of LSDT.

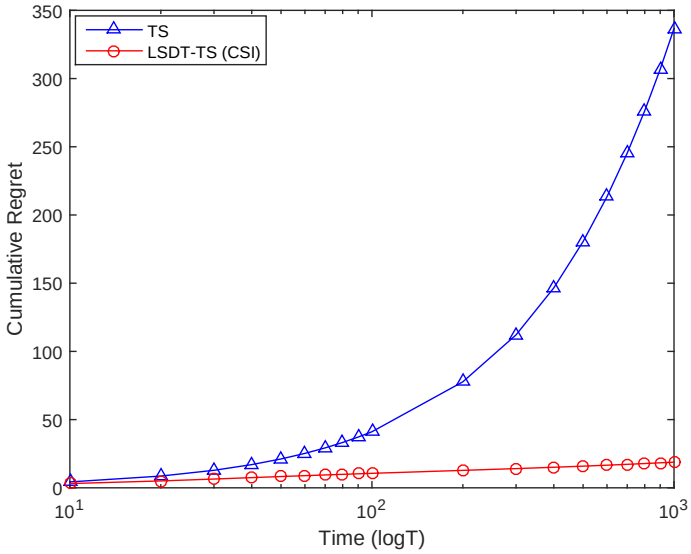

Figure 2.13: Regret performance with complete side information: LSDT-TS (CSI) v.s. classic TS.

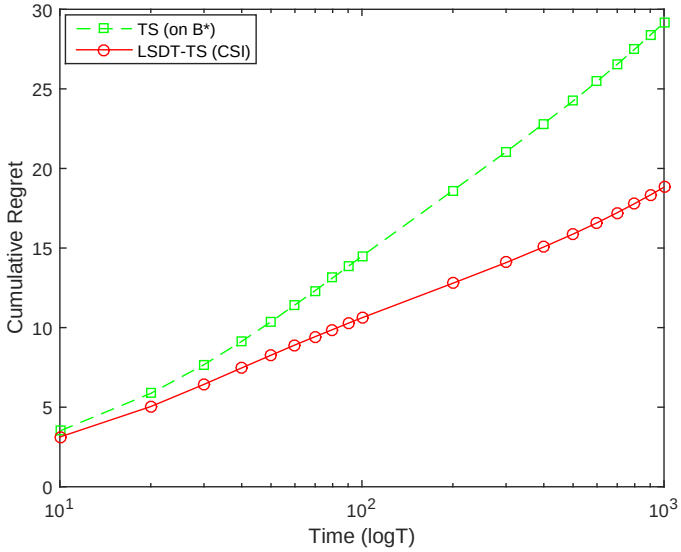

Figure 2.14: Regret performance with complete side information: LSDT-TS (CSI) v.s. TS on $\mathcal{B}^*$.

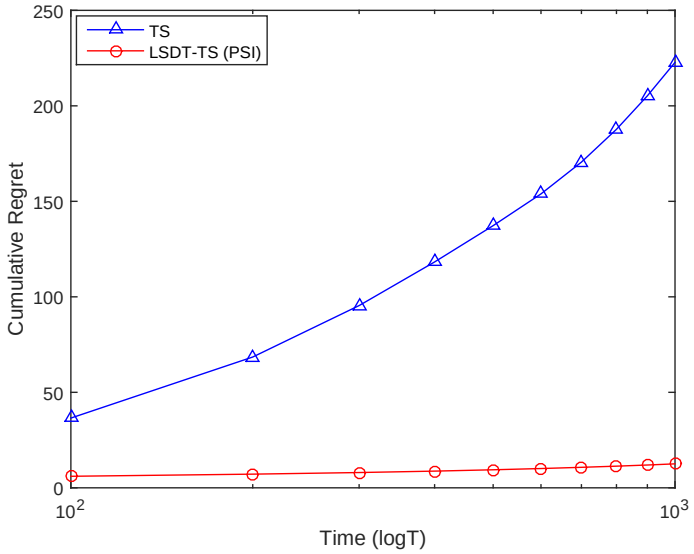

Figure 2.15: Regret performance with partial side information: LSDT-TS (PSI) v.s. classic TS.

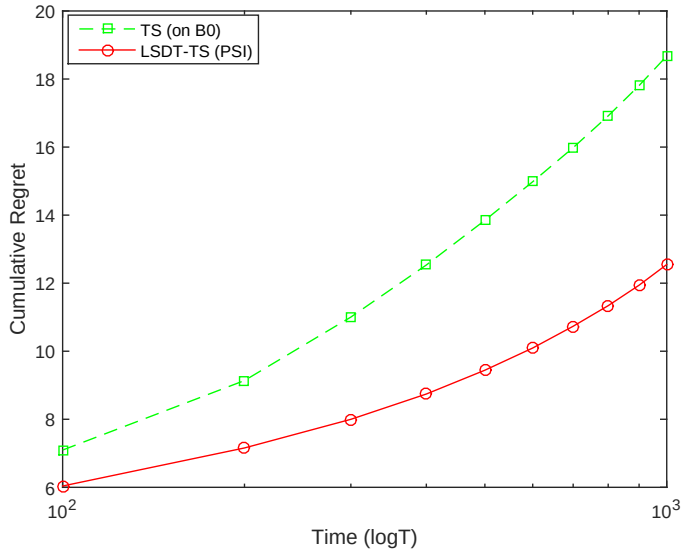

Figure 2.16: Regret performance with partial side information: LSDT-TS (PSI) v.s. TS on $\mathcal{B}_0$.

2.7.5 Comparison of Running Times

We compare the running time of LSDT-CSI as well as the baseline algorithms in Table 2.1 for the case of complete side information. It is not difficult to see that

LSDT-CSI has a relatively low computation complexity in contrast to algorithms with comparable performance, i.e., CKL-UCB and OSSB. Note that CKL-UCB and OSSB are time consuming since they have to solve an optimization problem at each time step. Besides, it can be seen that the time complexity of the offline reduction step is not too high to be applied.

Algorithm	UCB1	TS	CKL-UCB
Running Time (ms)	9.7	37.6	849.5
Algorithm	OSUB	OSSB	UCB1 on $\mathcal{B}^*$
Running Time (ms)	880.3	3.3×10^5	9.1
Algorithm	LSDT-CSI (offline)	LSDT-CSI (online)	
Running Time (ms)	14.1	18.6	

Table 2.1: Running times in the case of complete side information.

For the case of partial side information, we summarize the running times of LSDT-PSI and the other baselines in Table 2.2. Note that the running times of UCB1 and TS are smaller than LSDT-PSI since they ignore the similarity-dissimilarity relations across arms and have worse performance. When compared with CKL-UCB (achieves a comparable performance by exploiting the similarity relations), LSDT-PSI has a smaller computation complexity.

Algorithm	UCB1	TS	CKL-UCB
Running Time (ms)	10.3	38.3	354.2
Algorithm	LSDT-PSI (offline)	LSDT-CSI (online)	UCB1 on $\mathcal{B}_0$
Running Time (ms)	12.6	161.4	10.1

Table 2.2: Running times in the case of partial side information.

CHAPTER 3

BANDITS WITH MEMORY CONSTRAINTS

In canonical bandit models under both stochastic and adversarial settings, existing studies only focus on the regret performance of the learning policies without considering their memory complexity. Classical learning policies require a memory space with size linear in the number K of arms to store a statistic of every arm summarizing its reward history throughout the entire time horizon, which is infeasible in applications involving a large action space but limited memory. In this chapter, we study the memory-constrained adversarial bandit problem where a learning policy can only store the statistics of a subset of arms in the memory at any given time.

The memory constraint gives rise to two new problems in addition to arm selection: (i) *which arms' statistics should be stored in the memory at every time step*, and (ii) *how to memorize the reward history of arms whose statistics are not stored?*. In this chapter, we develop a hierarchical learning policy as a solution to the two new problems, which trades off the regret order with memory complexity.

3.1 Literature Review of MAB with Memory Constraints

Memory-constrained bandit problems have been recently studied under the stochastic reward models in [51, 22]. Two policies adopting best-arm identification techniques [14] in deciding which arm to store in the RMS were developed. Specifically, both policies partition arms into groups (the group size depends on the memory constraint) and time horizon into epochs. Within every epoch,

a new group of arms are selected in a round-robin fashion and the arms are stored in the RMS along with the predicted best arm that has been played so far. The prediction of the best arm is iteratively refined according to a best-arm identification strategy at the end of every epoch after exploring the new arm group, and the statistics of sub-optimal arms are eliminated from the memory. The difference between the two policies is that, the one in [51] is based on an explore-then-exploit structure while the other in [22] conducts exploration and exploitation simultaneously by a UCB policy during every epoch. Sublinear regret orders were established for both policies in the stochastic setting.

In the adversarial setting, however, the above mentioned policies for stochastic bandits are no longer applicable due to the inconsistency between the comparison of arms within a time period and their true rankings over the entire time horizon. Specifically, induced by the memory constraint, only a subset of arms can be played and compared during a period of time. In the stochastic setting with fixed reward distributions, the partial views on arm rewards within a time period are consistent with the ground truth. Hence, arm eliminations from the memory with sufficiently high probabilities can be carried out without inflicting a large regret. In the adversarial setting, however, the partial views and the ground truth over the entire time horizon are inconsistent. Therefore, all arms need to remain in the contention until the end of the horizon. Moreover, the policies developed in the stochastic setting are deterministic, and thus suffer linear regret orders in T against adversaries (we verify the claim numerically in Sec. 3.6). New learning policies are needed for the memory-constrained adversarial bandit problem.

Another type of memory constraint that has been studied in the MAB litera-

ture is temporal across time steps: a policy can only make decisions based on the reward outcomes of the m most recent plays. This problem was first considered in [56] where a two-armed bandit problem with Bernoulli rewards was studied. It was later shown in [29] that there exsits a policy with $m = 2$ that achieves an asymptotically optimal average reward in the two-armed bandit instance. The decision process with temporal memory constraints was further modeled as a finite-state machine in [28], where the past reward history was aggregated as a finite-valued statistic. The objective considered in these studies was the asymptotic convergence of the empirical average reward. Analysis on the convergence rate or the regret order, however, was lacking.

The objective of minimizing regret with temporal memory constraints was considered in [53] under the full-information feedback setting (i.e., the rewards of all arms that the player could have played are revealed after every time step). A learning algorithm with $O(m^K)$ states (each arm statistic can take $O(m)$ values) was developed. It was shown that if $m = O(\sqrt{T})$, the algorithm achieves an optimal regret order up to a logarithmic factor. However, the full-information feedback setting is fundamentally different from the bandit setting studied in this dissertation. Moreover, the proposed learning algorithm needs to store a statistic of every arm and the total number of states is exponential in K. Therefore, the algorithm is inapplicable in cases with a massive number of arms.

3.2 Problem Formulation

We consider an adversarial bandit problem with a finite arm set $\mathcal{A} = \{1, 2, ..., K\}$. At each time $t = 1, 2, ..., T$, a player chooses one arm to play. The reward $r_{i,t} \in$

$[0, 1]$ of playing an arm i at time t is assigned by an adversary. We assume that the adversary is *oblivious*, i.e., the assignment of the reward at time t is independent of the player's past actions. Equivalently, an oblivious adversary determines the sequence of rewards $((r_{1,t}, ..., r_{K,t}))_{t=1}^{T}$ ahead of time. We assume that the player can only observe the reward of the selected arm at each time.

The objective of the player is an online learning policy π that specifies a sequential arm selection rule at each time t based on the observation history. We assume that the policy can only use M ($M = o(K)$ as $K \to \infty$) words of memory space to store input values and necessary parameters. We follow the memory model studied in [22] where each of the variables used by the policy takes $O(1)$ word space[1] and thus, a policy with memory size M can only store the statistics of at most M arms (or aggregated statistics of at most M groups of arms) at any given time.

The performance of policy π is measured by regret, which is defined as the reward loss against the best benchmark action sequence $a^T = (a_1, ..., a_T)$ with the greatest cumulative reward, i.e.,

$$R_\pi(T) = \max_{a^T \in \mathcal{A}^T} \sum_{t=1}^{T} r_{a_t,t} - \sum_{t=1}^{T} r_{\pi_t,t}, \tag{3.1}$$

where π_t is the arm selected by policy π at time t. When there is no ambiguity, the notation is simplified to $R(T)$.

As the regret $R(T)$ can be randomized due to the potential randomness of the arm selection policy π, we consider two types of *no-regret learning* conditions in this chapter. A policy π is said to achieve no-regret learning *in expectation* if, for every sequence of rewards $((r_{1,t}, ..., r_{K,t}))_{t=1}^{T}$, the expected regret $E_\pi[R(T)] = o(T)$

[1]The number of bits in a word depends on how real numbers are stored in the memory, which is out of the scope of this chapter.

as $T \to \infty$, where the expectation is taken over the possible randomness of π. The second condition states that a policy π achieves no-regret learning *with high probability* if, for every sequence of rewards and every given $\delta \in (0, 1)$, the regret $R(T) = o(T)$ as $T \to \infty$ with probability at least $1 - \delta$.

It is not difficult to see that achieving no-regret learning, either in expectation or with high probability, is impossible if the benchmark sequence is chosen arbitrarily [11]. Therefore, certain restrictions on the benchmark sequence is necessary to make the problem feasible. In this chapter, we consider two types of regret notions with different restrictions on the benchmark sequence. The first regret notion is the so-called *weak regret* where the benchmark sequence consists of a single arm, i.e.,

$$R_{\mathrm{W}}(T) = \max_{i \in \mathcal{A}} \sum_{t=1}^{T} r_{i,t} - \sum_{t=1}^{T} r_{\pi_t,t}. \tag{3.2}$$

A stronger regret notion is the so-called *shifting regret* where the benchmark sequence is constrained by its *hardness*. Specifically, the hardness of a sequence $a^T = (a_1, ..., a_T)$ measures the total number of action changes over time, i.e.,

$$H(a^T) \triangleq 1 + \sum_{t=1}^{T-1} \mathbb{I}(a_t \neq a_{t+1}), \tag{3.3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The shifting regret with a hardness constraint V is defined as

$$R_{\mathrm{S}}(T, V) = \max_{a^T : H(a^T) \leq V} \sum_{t=1}^{T} r_{a_t,t} - \sum_{t=1}^{T} r_{\pi_t,t}. \tag{3.4}$$

It is clear that the shifting regret is a stronger notion than the weak regret: no-regret learning under the former implies no-regret learning under the latter, but not vice versa.

3.3 Hierarchical Learning with Memory Constraints

In this section, we propose a general learning structure: *HLMC (Hierarchical Learning with Memory Constraints)* for the memory-constrained adversarial bandit problem. We first present the general framework of HLMC with a multi-level hierarchy on the partitions of the arm set and the time horizon. Then we use a representative case with a two-level hierarchy to illustrate the details of the HLMC policy.

3.3.1 A General Framework with Multi-Level Hierarchy

To address the issue that only the statistics of a subset of arms can be stored in the memory at any given time, the key technique is to properly aggregate and store the statistics of a group of arms to memorize their reward history jointly. We introduce a general D-level hierarchy partitioning arms into D levels of groups in an iterative way: the arm set is defined as the level-0 group, every level-d ($d \leq D-1$) group is partitioned into several level-$(d+1)$ groups, and every level-D group consists of a single arm. Similarly, the time horizon is iteratively partitioned into D levels of epochs.

The HLMC policy with a D-level hierarchy is then specified by D selection strategies at all levels in a recursive fashion: at the beginning of every level-d epoch, a level-d group is selected by a level-d strategy. Within that epoch, the policy zooms in to the selected group and conducts a level-$(d+1)$ selection strategy based on the group statistics of all involved level-$(d+1)$ groups. A level-$(d+1)$ group statistics aggregates the reward information of arms within that

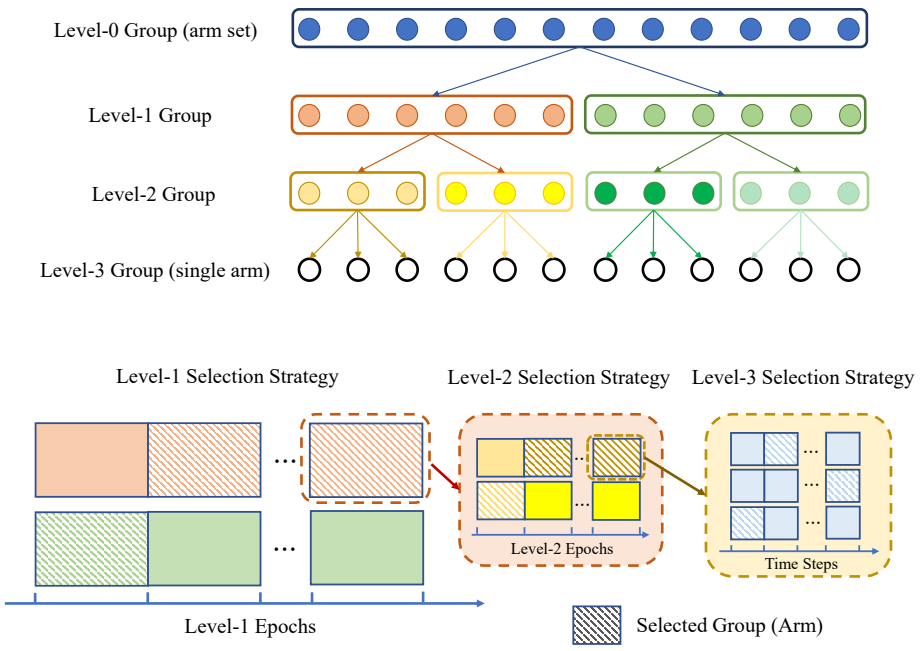

Figure 3.1: HLMC with a three-level hierarchy.

group, and is stored in the memory until the end of the corresponding level-d epoch. At the last level of the hierarchy, the selection strategy decides which arm within the targeted group to play at every time step based on the arm statistics. The set of arms (and arm groups) whose statistics are stored in the memory is called the *reward-memorized set (RMS)*. See Fig. 3.1 for an example of the HLMC policy with a three-level hierarchy.

The HLMC policy solves the problem on deciding which arms' statistics to store in the memory: only arms within the targeted group at the last level is stored in the RMS during the corresponding epoch. For the other arms outside the targeted group, their reward information is jointly stored in the memory through certain group statistics at higher levels.

3.3.2 A Representative Case with Two-Level Hierarchy

We use $D = 2$ as a representative case to present the details of HLMC. The two levels in the hierarchy are referred to as the group and the arm levels. Specifically, the set $\mathcal{A}$ of arms is partitioned into equal-sized groups $\{\mathcal{A}_\ell\}_{\ell=1}^{L}$ where

$$\mathcal{A}_\ell = \{1 + N(\ell - 1), ..., \min(N\ell, K)\}, \tag{3.5}$$

$N = \lceil \frac{M - \sqrt{M^2 - 4K}}{2} \rceil$ is the group size (note that the number of arms in the last group may be smaller than N), and $L = \lceil \frac{K}{N} \rceil$ is the number of groups. The time horizon is partitioned into equal-length epochs $\{\mathcal{T}_s\}_{s=1}^{S}$ where

$$\mathcal{T}_s = [1 + \Delta(s - 1), \min(\Delta S, T)], \tag{3.6}$$

$\Delta \in \mathbb{N}^+$ is the epoch length to be determined later, and $S = \lceil \frac{T}{\Delta} \rceil$ is the number of epochs. Note that the length of the S-th epoch may be smaller than Δ.

By treating each arm group $\mathcal{A}_\ell$ as a "super arm" ℓ and each epoch $\mathcal{T}_s$ a "super time-step" s, we reduce the group selection problem to a classic adversarial bandit problem. The reward of playing a "super arm" ℓ_s at a "super time-step" s is defined as the average per-time reward from the corresponding arm group $\mathcal{A}_{\ell_s}$ during the corresponding epoch $\mathcal{T}_s$, i.e.,

$$y_{\ell_s, s} = \frac{1}{|\mathcal{T}_s|} \sum_{t \in \mathcal{T}_s} r_{i_t, t}, \tag{3.7}$$

where $i_t \in \mathcal{A}_{\ell_s}$ is the arm selected at time t.

The problem on deciding which arm to store in the RMS is then addressed by solving the adversarial bandit problem constructed by the reduction. Once a group of arms is selected and stored in the RMS, the decision problem on playing arms is addressed by conducting a learning algorithm for classic adversarial

bandits as a subroutine on the selected arm group during every epoch. The detailed HLMC policy with a two-level hierarchy is summarized in Algorithm 5.

Algorithm 5: **HLMC**

Input: M the memory size, T the time length, $\mathcal{A}$ the set of K arms, and $\Delta > 0$ the epoch length.

if $M \geq K$ **then**

Run a classic adversarial bandit algorithm on $\mathcal{A}$.

else

Obtain arm group partition $\{\mathcal{A}_\ell\}_{\ell=1}^{L}$ according to (3.5).

Obtain epoch partition $\{\mathcal{T}_s\}_{s=1}^{S}$ according to (3.6).

Initialize and store the statistics of every arm group.

for $s = 1, 2, ..., S$ **do**

Select arm group ℓ_s according to the group-level selection strategy.

Initialize and store the statistics of every arm in $\mathcal{A}_{\ell_s}$.

Initialize $y_{\ell_s,s} = 0, \tau = 0$.

for $t \in \mathcal{T}_s$ **do**

Play arm i_t according to the arm-level selection strategy and receive reward $r_{i_t,t}$.

Update arm statistics in the memory using $r_{i_t,t}$.

Update $y_{\ell_s,s} = \frac{y_{\ell_s,s}\tau + r_{i_t,t}}{\tau+1}$, $\tau = \tau + 1$.

Update all group statistics in the memory using $y_{\ell_s,s}$.

It should be noted that at both group and arm levels in the HLMC learning policy, any adversarial bandit algorithm that achieves no-regret learning can be incorporated as a subroutine to guarantee a sublinear regret order in T. In the next section, we discuss applying three different learning algorithms for classic

adversarial bandits to minimize two notions of regret in expectation and/or with high probability.

3.4 Memory Complexity and Regret Performance in the Two-Level Case

In this section, we analyze the memory complexity and regret performance of the proposed HLMC learning policy in the case with $D = 2$. We notice that in HLMC, the group-level strategy requires L words of memory to store a statistic of every group. Once a groups is selected, the statistics of all arms within the selected group should also be stored, which require N additional words of memory. Therefore, the total memory size required by the HLMC policy is $N+L$. As long as $M \geq 2\sqrt{K}$, the group partition in (3.5) is legitimate and one can verify that $N+L \leq M$. Therefore, the minimum memory space required by HLMC with a two-level hierarchy is or order $\Omega(\sqrt{K})$.

In terms of the regret performance, it is clear that the regret order achieved by HLMC depends on the specific selection strategies adopted as subroutines at both group and arm levels. In the following three subsections, we discuss minimizing weak regret in expectation and with high probability, and minimizing shifting regret in expectation respectively through adopting different learning algorithms as subroutines.

3.4.1 Minimizing Weak Regret in Expectation

We first consider minimizing the expected weak regret through applying EXP3 at both group and arm levels in the HLMC policy. The EXP3 algorithm was first proposed in [10]. It randomly selects an action i_t according to a distribution $(p_{i,t})_{i\in\mathcal{A}}$ at every time t. The probability $p_{i,t}$ is the sum of two components. The first one is proportional to a weight $w_{i,t}$ exponential in the estimated cumulative reward from arm i up to time t, i.e., $w_{i,t} = \prod_{\tau=1}^{t} \exp(\gamma \hat{r}_{i,\tau}/|\mathcal{A}|)$, where $\gamma > 0$ is the learning rate and $\hat{r}_{i,\tau}$ is an unbiased estimator of $r_{i,t}$ with respect to the random choice of i_t. The second component is a random exploration term $\gamma/|\mathcal{A}|$ guaranteeing sufficient exploration of every arm. The EXP3 algorithm achieves a sublinear regret order in the time length if γ is selected appropriately. The details of EXP3 are summarized in Algorithm 6.

Algorithm 6: **EXP3** [10, 11]

Input: $\mathcal{A}$ the arm set and $\gamma \in (0, 1)$.

Initialize $w_{i,1} = 1, \forall i \in \mathcal{A}$.

for $t = 1, 2, ..., T$ **do**

Let

$$p_i(t) = (1-\gamma)\frac{w_{i,t}}{\sum_{j\in\mathcal{A}} w_{j,t}} + \frac{\gamma}{|\mathcal{A}|}, \quad \forall i \in \mathcal{A}.$$

Draw arm i_t according to the probabilities $(p_{i,t})_{i\in\mathcal{A}}$.

Receive reward $r_{i_t,t}$.

Let $\hat{r}_{i,t} = \frac{r_{i,t}}{p_{i,t}}\mathbb{I}(i_t = i)$ and update

$$w_{i,t+1} = w_i(t)\exp\left(\frac{\gamma \hat{r}_{i,t}}{|\mathcal{A}|}\right), \quad \forall i \in \mathcal{A}$$

We show in the following theorem that adopting EXP3 at both group and arm levels in HLMC (with learning rates γ_1 and γ_2 respectively) guarantees a sublinear regret order in T under the notion of expected weak regret.

Theorem 7 *For any T, K, and M, assume $M \geq 2\sqrt{K}$. If the input parameter $\Delta = \left\lceil \sqrt{\frac{TN \ln N}{L \ln L}} \right\rceil$ (where N, L are defined in the algorithm), adopting EXP3 at both group and arm levels with learning rates $\gamma_1 = \sqrt{\frac{L \ln L}{2S}}$ and $\gamma_2 = \sqrt{\frac{N \ln N}{2\Delta}}$ guarantees that, for every assignment of the reward sequence, the expected weak regret of HLMC with a two-level hierarchy is upper bounded as follows:*

$$\mathbb{E}_{HLMC}[R_w(T)] \leq (4 + 2\sqrt{2}) T^{\frac{3}{4}} K^{\frac{1}{4}} (\ln K)^{\frac{1}{2}}. \tag{3.8}$$

To obtain the upper bound in Theorem 7, we decompose the expected weak regret into two parts by introducing an intermediate term $C'_{\max}$ as follows: for every fixed reward sequence, let $i_{\max}$ be the best arm with the greatest cumulative reward over the entire time horizon and $\mathcal{A}_{\ell_{\max}}$ the arm group to which $i_{\max}$ belongs. We define $C'_{\max}$ as the expected cumulative reward obtained by running the arm-level EXP3 algorithm with learning rate γ_2 on $\mathcal{A}_{\ell_{\max}}$ during all epochs, i.e.,

$$C'_{\max} = \sum_{s=1}^{S} \mathbb{E}_{\text{Arm-EXP3}(\mathcal{A}_{\ell_{\max}})} \left[\sum_{t \in \mathcal{T}_s} r_{i_t,t} \right], \tag{3.9}$$

where $\mathbb{E}_{\text{Arm-EXP3}(\mathcal{A}_{\ell_{\max}})}[\cdot]$ denotes the expectation taken over the randomness of the arm-level EXP3 algorithm when conducted on group $\mathcal{A}_{\ell_{\max}}$. Then the expected weak regret of HLMC is decomposed as:

$$\mathbb{E}_{\text{HLMC}}[R_{\text{W}}(T)] = \underbrace{(C'_{\max} - C_{\text{HLMC}})}_{R_1(T)} + \underbrace{(C_{\max} - C'_{\max})}_{R_2(T)}, \tag{3.10}$$

where

$$C_{\text{HLMC}} = \mathbb{E}_{\text{HLMC}} \left[\sum_{t=1}^{T} r_{i_t,t} \right],$$

$$C_{\max} = \sum_{t=1}^{T} r_{i_{\max},t}. \tag{3.11}$$

Note that in the decomposition, $R_1(T)$ corresponds to the group-level reward loss due to not selecting $\mathcal{A}_{\ell_{\max}}$ at every epoch, and $R_2(T)$ corresponds to the arm-level reward loss due to playing suboptimal arms in $\mathcal{A}_{\ell_{\max}}$ assuming that group $\mathcal{A}_{\ell_{\max}}$ is selected at all epochs.

We first upper bound the group-level reward loss $R_1(T)$. Noticing that the arm selection process during every epoch is independent of the group and arm selection history in the past, we can thus rewrite the expected reward of the HLMC policy as follows:

$$\begin{aligned} & \mathbb{E}_{\text{HLMC}}\left[\sum_{t=1}^{T} r_{i_t,t}\right] \\ = \; & \mathbb{E}_{\text{Group-EXP3}}\left[\sum_{s=1}^{S} \mathbb{E}_{\text{Arm-EXP3}(\mathcal{A}_{\ell_s})}\left[\sum_{t\in\mathcal{T}_s} r_{i_t,t}\right]\right], \end{aligned} \tag{3.12}$$

where $\mathbb{E}_{\text{Group-EXP3}}[\cdot]$ denotes the expectation taken over the randomness of the group-level EXP3 algorithm, and $\mathcal{A}_{\ell_s}$ is the group selected at epoch s. To ease the analysis, we assume without losing generality that all epochs have an equal length Δ. We further define

$$x_{\ell,s} = \mathbb{E}_{\text{Arm-EXP3}(\mathcal{A}_\ell)}\left[\frac{1}{|\mathcal{T}_s|}\sum_{t\in\mathcal{T}_s} r_{i_t,t}\right]. \tag{3.13}$$

It is not difficult to see that

$$R_1(T) = \Delta\left(\sum_{s=1}^{S} x_{\ell_{\max},s} - \mathbb{E}_{\text{Group-EXP3}}\left[\sum_{s=1}^{S} x_{\ell_s,s}\right]\right). \tag{3.14}$$

It is then clear that upper bounding $R_1(T)$ is equivalent to upper bounding the weak regret of applying the group-level EXP3 algorithm to the adversarial bandit problem constructed by the reduction in Sec. 3.3. Specifically, the

reward of selecting a group $\mathcal{A}_\ell$ at epoch $\mathcal{T}_s$ is defined as $y_{\ell,s}$ according to (3.7) where i_t is randomly selected by the arm-level EXP3 algorithm. Therefore, $y_{\ell,s}$ is a random reward with mean $x_{\ell,s}$. The group selection problem is reduced to a classic adversarial bandit problem with noisy observations. It should be noted that after fixing an assignment of the reward sequence $((r_{1,t}, ..., r_{K,t}))_{t=1}^T$, the expected reward $x_{\ell,s}$ is fixed. Meanwhile, the realization of $y_{\ell,s}$ is independent across ℓ, s and is independent of the arm (group) selection history up to epoch s. We obtain the following result on applying the group-level EXP3 algorithm to the constructed adversarial bandit problem.

Lemma 1 *By choosing $\gamma_1 = \sqrt{\frac{L \ln L}{2S}}$, the group-level EXP3 algorithm guarantees that, for every assignment of the reward sequence $((r_{1,t}, ..., r_{K,t}))_{t=1}^T$,*

$$\max_{1\le\ell\le L} \sum_{s=1}^{S} x_{\ell,s} - \mathbb{E}_{\textit{Group-EXP3}}\left[\sum_{s=1}^{S} x_{\ell_s,s}\right] \le 2\sqrt{2SL\ln L}, \tag{3.15}$$

where ℓ_s is the arm group selected by the group-level EXP3 algorithm at epoch s.

Proof 8 *See Appendix B.1.*

For the arm-level reward loss $R_2(T)$, we notice that

$$R_2(T) = \sum_{s=1}^{S}\left(\sum_{t\in\mathcal{T}_s} r_{i_{\max},t} - \mathbb{E}_{\text{Arm-EXP3}(\mathcal{A}_{\ell_{\max}})}\left[\sum_{t\in\mathcal{T}_s} r_{i_t,t}\right]\right). \tag{3.16}$$

It suffices to upper bound each term in the summation, that is, the weak regret of conducting the arm-level EXP3 algorithm on group $\mathcal{A}_{\ell_{\max}}$ during each epoch $\mathcal{T}_s$.

Lemma 2 (Corollary 3.2 in [11]) *By choosing $\gamma_2 = \sqrt{\frac{|\mathcal{A}_\ell| \ln |\mathcal{A}_\ell|}{2|\mathcal{T}_s|}}$, the arm-level EXP3 algorithm conducted on arm group $\mathcal{A}_\ell$ during epoch $\mathcal{T}_s$ guarantees that, for every assignment of the reward sequence,*

$$\max_{i\in\mathcal{A}_\ell} \sum_{t\in\mathcal{T}_s} r_{i,t} - \mathbb{E}_{\textit{Arm-EXP3}(\mathcal{A}_\ell)}\left[\sum_{t\in\mathcal{T}_s} r_{i_t,t}\right] \le 2\sqrt{2\Delta N\ln N}, \tag{3.17}$$

where i_t is the arm selected by the arm-level EXP3 algorithm at time t.

Theorem 7 is then proved by applying Lemma 1 and Lemma 2 to $R_1(T)$ and $R_2(T)$, respectively.

Proof 9 (Proof of Theorem 1) *Combining (3.14) with Lemma 1, and (3.16) with Lemma 2, we can derive that*

$$\begin{aligned} R_1(T) &\leq 2\Delta\sqrt{2SL\ln L} = 2\sqrt{2T\Delta L\ln L}, \\ R_2(T) &\leq 2S\sqrt{2\Delta N\ln N} = 2\sqrt{\frac{2T^2}{\Delta}N\ln N}. \end{aligned} \tag{3.18}$$

By choosing $\Delta = \left\lceil \sqrt{\frac{TN\ln N}{L\ln L}} \right\rceil$, we obtain the upper bound in Theorem 7.

It should be noted that although the proposed learning policy requires the knowledge of the total time length T for choosing input parameters to achieve no-regret learning, the issue of unknown T can be easily addressed by the doubling technique as used in the classic adversarial bandit problem [11]. Specifically, we partition the time horizon into phases with length $T_r = 2^r, r = 0, 1, ...,$ and run the HLMC policy as a subroutine in every phase. The input parameters are chosen accordingly by letting $T = T_r$. It is not difficult to show that the same regret order still holds.

3.4.2 Minimizing Weak Regret with High Probability

In this subsection, we further discuss applying EXP3.P [11, 13] at both group and arm levels of HLMC to achieve no-regret learning with high probability under the notion of weak regret.

Different from EXP3 that uses an unbiased estimate $\hat{r}_{i,t} = \frac{r_{i,t}\mathbb{I}(i_t=i)}{p_{i,t}}$ in updating arm weights, the EXP3.P algorithm adopts an upper confidence bound $\tilde{r}_{i,t}$ instead, which guarantees that the true reward is upper bounded by the new estimate with high probability. The detailed EXP3.P algorithm is summarized in Algorithm 7.

Algorithm 7: **EXP3.P** [13]

Input: $\mathcal{A}$ the arm set, $\eta > 0$, and $\gamma, \beta \in (0, 1)$.

Initialize $w_{i,1} = 1, \forall i \in \mathcal{A}$.

for $t = 1, 2, ..., T$ **do**

Let

$$p_i(t) = (1-\gamma)\frac{w_{i,t}}{\sum_{j\in\mathcal{A}} w_{j,t}} + \frac{\gamma}{|\mathcal{A}|}, \quad \forall i \in \mathcal{A}.$$

Draw arm i_t according to the probabilities $(p_{i,t})_{i\in\mathcal{A}}$.

Receive reward $r_{i_t,t}$.

Let

$$\tilde{r}_{i,t} = \frac{r_{i,t}\mathbb{I}(i_t = i) + \beta}{p_{i,t}}.$$

Update

$$w_{i,t+1} = w_i(t)\exp\left(\eta\tilde{r}_{i,t}\right), \quad \forall i \in \mathcal{A}$$

We show in the following theorem that by adopting EXP3.P at both group and arm levels with parameters $(\eta_1, \gamma_1, \beta_1)$ and $(\eta_2, \gamma_2, \beta_2)$ respectively, the weak regret of HLCM has a sublinear growth rate in T with high probability.

Theorem 8 *For any T, K, M, assume that $M > 2\sqrt{K}$. For any $\delta \in (0, 1)$, if the input parameter $\Delta = \left\lceil \sqrt{\frac{TN\ln(2KT/\delta)}{L\ln(2L/\delta)}} \right\rceil$ (where N, L are defined in the algorithm), and the EXP3.P algorithm is adopted at both the group level with $\beta_1 = \sqrt{\frac{\ln(2L/\delta)}{LS}}, \eta_1 = 0.95\sqrt{\frac{\ln L}{LS}}, \gamma_1 =$*

$1.05\sqrt{\frac{L\ln L}{S}}$, *and the arm level with* $\beta_2 = \sqrt{\frac{\ln(2KS/\delta)}{N\Delta}}, \eta_2 = 0.95\sqrt{\frac{\ln N}{N\Delta}}, \gamma_2 = 1.05\sqrt{\frac{N\ln N}{\Delta}}$, *then for any assignment of the reward sequence, the weak regret of HLCM with a two-level hierarchy is upper bounded by*

$$R_{\mathcal{W}}(T) \leq 12.5T^{\frac{3}{4}}K^{\frac{1}{4}}(\ln(2KT/\delta))^{\frac{1}{2}}, \tag{3.19}$$

with probability at least $1-\delta$.

Theorem 8 is proved through a similar structure with that used in analyzing the expected weak regret of HLMC in Sec. 3.4.1. Specifically, the weak regret is decomposed as:

$$\begin{aligned} R_{\mathrm{W}}(T) &= \sum_{s=1}^{S}\sum_{t\in\mathcal{T}_s} r_{i_{\max},t} - \sum_{s=1}^{S}|\mathcal{T}_s|y_{\ell_{\max},s} \\ &\quad + \sum_{s=1}^{S}|\mathcal{T}_s|y_{\ell_{\max},s} - \sum_{s=1}^{S}\sum_{t\in\mathcal{T}_s} r_{i_t,t} \\ &= R_1(T) + R_2(T), \end{aligned} \tag{3.20}$$

where $i_{\max}$ is the arm with the greatest cumulative reward in hindsight, $\ell_{\max}$ is the group index of $i_{\max}$, $y_{\ell_{\max},s}$ is the average reward obtained by running the arm-level EXP3.P algorithm on $\mathcal{A}_{\ell_{\max}}$ during epoch s, and i_t is the arm selected by the algorithm at time t.

We first upper bound $R_1(T)$, which corresponds to the arm-level reward loss due to playing suboptimal arms in $\mathcal{A}_{\ell_{\max}}$ assuming that $\mathcal{A}_{\ell_{\max}}$ is selected at all epochs. It suffices to upper bound

$$\sum_{t\in\mathcal{T}_s} r_{i_{\max},t} - |\mathcal{T}_s|y_{\ell_{\max},s}, \tag{3.21}$$

for every s. It is clear that (3.21) is equivalent to the weak regret of applying the arm-level EXP3.P algorithm on $\mathcal{A}_{\ell_{\max}}$ during epoch $\mathcal{T}_s$, which is upper bounded in the following lemma proved in [13].

Lemma 3 (Theorem 3.2 in [13]) *For every* $\delta_0 \in (0,1)$*, by choosing* $\beta_2 = \sqrt{\frac{\ln(N/\delta_0)}{N\Delta}}, \eta_2 = 0.95\sqrt{\frac{\ln N}{N\Delta}}, \gamma_2 = 1.05\sqrt{\frac{N\ln N}{\Delta}}$*, the arm-level EXP3.P algorithm conducted on arm group* $\mathcal{A}_\ell$ *during epoch* $\mathcal{T}_s$ *guarantees that, for every assignment of the reward sequence,*

$$\sum_{t\in\mathcal{T}_s} r_{i_{\max},t} - |\mathcal{T}_s| y_{\ell_{\max},s} \leq 5.15\sqrt{N\Delta\ln(N/\delta_0)} \tag{3.22}$$

with probability at least $1-\delta_0$*.*

To upper bound $R_2(T)$, which corresponds to the group-level reward loss due to not selecting $\mathcal{A}_{\ell_{\max}}$ at every epoch, we rewrite $R_2(T)$ as

$$R_2(T) = \Delta\left(\sum_{s=1}^{S} y_{\ell_{\max},s} - \sum_{s=1}^{S} y_{\ell_s,s}\right) \tag{3.23}$$

where ℓ_s is the group selected by the group-level EXP3.P algorithm at epoch s (we assume without loss of generality that every epoch has equal length Δ).

As argued in Sec. 3.4.1, the realization of $y_{\ell,s}$ is independent across ℓ, s and is independent of the past group selection history. Once we fixed a sequence of realizations of $((y_{1,s},...y_{L,s}))_{s=1}^{S}$, Lemma 3 can be applied to upper bound the group-level regret $R_2(T)$ with high probability.

Proof 10 (Proof of Theorem 8) *For any* $\delta > 0$*, we apply Lemma 3 to all groups* $\ell = 1,...,L$ *and all epochs* $s = 1,...,S$ *by choosing* $\delta_0 = \frac{\delta}{2LS}$*. Then using the union bound, we obtain that with probability at least* $1-\delta/2$*, the upper bound in (3.21) holds for all* ℓ *and* s*. As a result, the arm-level regret* $R_1(T)$ *is upper bounded as:*

$$\begin{aligned} R_1(T) &\leq 5.15S\sqrt{N\Delta\ln(2NLS/\delta)} \\ &= 5.15\sqrt{\frac{T^2}{\Delta}N\ln\left(\frac{2KS}{\delta}\right)}, \end{aligned} \tag{3.24}$$

with probability at least $1-\delta/2$*.*

Moreover, we apply Lemma 3 again to the group-level selection strategy by choosing $\delta_0 = \delta/2$. *We obtain that with probability at least* $1 - \delta/2$,

$$\begin{aligned} R_2(T) &\leq 5.15\Delta\sqrt{LS\ln(2L/\delta)} \\ &= 5.15\sqrt{T\Delta L\ln(2L/\delta)}. \end{aligned} \tag{3.25}$$

The upper bound on $R_{\mathcal{W}}(T)$ *in Theorem 8 is obtained by choosing* $\Delta = \left\lceil \sqrt{\frac{TN\ln(2KT/\delta)}{L\ln(2L/\delta)}} \right\rceil$ *and combining (3.24) and (3.25) using the union bound.*

3.4.3 Minimizing Shifting Regret in Expectation

To achieve no-regret learning under a stronger regret notion: shifting regret, we consider applying EXP3.S, a variant of the EXP3 algorithm, at the group level of the HLMC policy.

In the EXP3.S algorithm, a fixed share is added to the update process of arm weights, i.e., $g_{\ell,s+1} = g_{\ell,s}\exp(\gamma'\hat{y}_{\ell,s}) + \alpha' G_s$. One step forward gives that $g_{\ell,s+2} = g_{\ell,s}\exp(\gamma'(\hat{y}_{\ell,s} + \hat{y}_{\ell,s+1})) + \alpha' G_s\exp(\gamma'\hat{y}_{\ell,s+1})$. It is not difficult to see that $\hat{y}_{\ell,s+1}$ has a greater impact than $\hat{y}_{\ell,s}$ on future arm weights. As a result, the arm selection relies more on recent rewards. The detailed EXP3.S algorithm is summarized in Algorithm 8.

At the arm-level, we still adopt the EXP3 algorithm for arm selection. It should be noted that the arm-level strategy in the HLMC policy is restarted at the beginning of every epoch, which guarantees quick elimination of the past experience. Therefore, the hierarchical structure automatically adapts to the variation of the benchmark sequence. In the following theorem, we provide an upper bound on the expected shifting regret of HLMC when EXP3.S and EXP3

Algorithm 8: **EXP3.S** [11]

Input: $\mathcal{A}$ the arm set, $\gamma \in (0,1)$, and $\alpha > 0$.

Initialize $w_{i,1} = 1, \forall i \in \mathcal{A}$.

for $t = 1, 2, ..., T$ **do**

Let

$$p_i(t) = (1-\gamma)\frac{w_{i,t}}{\sum_{j\in\mathcal{A}} w_{j,t}} + \frac{\gamma}{|\mathcal{A}|}, \quad \forall i \in \mathcal{A}.$$

Draw arm i_t according to the probabilities $(p_{i,t})_{i\in\mathcal{A}}$.

Receive reward $r_{i_t,t}$.

Let $\hat{r}_{i,t} = \frac{r_{i,t}}{p_{i,t}}\mathbb{I}(i_t = i)$ and update

$$w_{i,t+1} = w_i(t)\exp\left(\frac{\gamma\hat{r}_{i,t}}{|\mathcal{A}|}\right) + \frac{e\alpha}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} w_{i,t}.$$

are applied at the group and arm levels, respectively.

Theorem 9 *For any T, K, M, and V, assume that $M \geq 2\sqrt{K}$ and $T \geq VK$. If the input parameter $\Delta = \left\lceil \sqrt{\frac{TN\ln N}{VL\ln(TL)}} \right\rceil$ (where N, L are defined in the algorithm), adopting EXP3.S at the group level with $\gamma_1 = \sqrt{\frac{VL\ln(LS)}{S}}$, $\alpha = 1/S$, and EXP3 at the arm level with $\gamma_2 = \sqrt{\frac{N\ln N}{2\Delta}}$ guarantees that, for every assignment of the reward sequence, the expected shifting regret of HLMC with a two-level hierarchy is upper bounded by:*

$$\mathbb{E}_{HLMC}[R_S(T,V)] \leq 9T^{\frac{3}{4}}V^{\frac{1}{4}}K^{\frac{1}{4}}(\ln(KT))^{\frac{1}{2}}, \tag{3.26}$$

under a hardness constraint V on the benchmark action sequence.

Corollary 2 *If $V = o(T)$ as $T \to \infty$, the HLMC algorithm achieves no-regret learning in expectation under the notion of shifting regret with hardness constraint V.*

To upper bound the expected shifting regret of the HLMC algorithm against an arbitrary benchmark action sequence a^T with a hardness constraint V, the key technique is to construct an alternative benchmark sequence b^T such that: (i) $H(b^T) \leq V$, (ii) the cumulative reward achieved by b^T is close to that achieved by a^T, and (iii) the actions specified by b^T are invariant within each epoch. With such a sequence b^T, it suffices to show that the expected shifting regret of HLMC against b^T has a sublinear growth rate in T.

We follow the same proof structure with that used for analyzing the expected weak regret in Sec. 3.4.1. First note that the constructed sequence b^T is fixed within each epoch. Therefore, the arm-level regret analysis in Lemma 2 directly carries over. At the group-level, the reduction to a new adversarial bandit problem with noisy observations is still legitimate since the group specified by the benchmark sequence is fixed within each epoch. Based on the reduction, we obtain the following result on applying the EXP3.S algorithm at the group level.

Lemma 4 *By choosing* $\gamma_1 = \sqrt{\frac{LV \ln(LS)}{S}}$ *and* $\alpha = 1/S$*, the group-level EXP3.S algorithm guarantees that, for every assignment of the reward sequence* $((r_{1,t}, ..., r_{K,t}))_{t=1}^T$ *and every benchmark sequence of arm groups* $h^S = (h_1, ..., h_S)$ *where* $H(h^S) \leq V$*,*

$$\sum_{s=1}^{S} x_{h_s,s} - \mathbb{E}_{\textit{Group-EXP3.S}} \left[\sum_{s=1}^{S} x_{\ell_s,s} \right] \leq 4\sqrt{VLS \ln(LS)}, \tag{3.27}$$

where ℓ_s *is the arm group selected at epoch* s*.*

Proof 11 *See Appendix B.2*

The upper bound in Theorem 9 on the expected shifting regret of the HLMC algorithm against any benchmark action sequence with hardness constraints V is obtained by combining Lemma 2 in Sec. 3.4.1 and Lemma 4 together.

Proof 12 (Proof of Theorem 9) *For an arbitrary benchmark action sequence a^T such that $H(a^T) \leq V$, we first construct an alternative benchmark sequence b^T as follows: suppose the time horizon is partitioned into V segments:*

$$[T_1, T_2), [T_2, T_3), ..., [T_V, T_{V+1}), \tag{3.28}$$

where $T_1 = 1, T_{V+1} = T + 1$, and a_t is fixed for all $t \in [T_v, T_{v+1})$ (let j_v denote that arm and h_v denote the group it belongs to). Suppose T_v belongs to epoch s_v. The alternative benchmark sequence b^T is defined as

$$b_t = j_v, \text{ if } s(t) \in [s_v, s_{v+1}), \tag{3.29}$$

where $s(t)$ is the epoch to which time t belongs.

One can check that the action specified by b^T is fixed within each epoch and $H(b^T) \leq V$. Moreover, b^T differs from a^T only in the epochs when a change happens in a^T, i.e., $\{s_v\}_{v=1}^V$. Therefore,

$$\sum_{t=1}^{T} (r_{a_t,t} - r_{b_t,t}) \leq V\Delta. \tag{3.30}$$

We decompose the expected shifting regret against a^T as:

$$\begin{aligned} & \mathbb{E}_{HLMC}[R_{a^T}(T)] \\ = & \sum_{t=1}^{T} (r_{a_t,t} - r_{b_t,t}) + \left(\sum_{t=1}^{T} r_{b_t,t} - \sum_{v=1}^{V} \sum_{s=s_v}^{s_{v+1}-1} |\mathcal{T}_s| x_{h_v,s} \right) \qquad (3.31) \\ & + \left(\sum_{v=1}^{V} \sum_{s=s_v}^{s_{v+1}-1} |\mathcal{T}_s| x_{h_v,s} - \mathbb{E}_{HLMC} \left[\sum_{t=1}^{T} r_{i_t,t} \right] \right) \\ = & R_1(T) + R_2(T) + R_3(T). \qquad (3.32) \end{aligned}$$

Note that $R_1(T) \leq V\Delta$. For $R_2(T)$, we have

$$\begin{aligned} R_2(T) &= \sum_{v=1}^{V} \sum_{s=s_v}^{s_{v+1}-1} \sum_{t \in \mathcal{T}_s} r_{b_t,t} - \sum_{v=1}^{V} \sum_{s=s_v}^{s_{v+1}-1} |\mathcal{T}_s| x_{h_v,s} \\ &\leq 2S \sqrt{2\Delta N \ln N}, \qquad (3.33) \end{aligned}$$

where the last inequality uses Lemma 2.

For $R_3(T)$, we can show that

$$\begin{aligned} R_3(T) &= \sum_{v=1}^{V} \sum_{s=s_v}^{s_{v+1}-1} |\mathcal{T}_s| x_{h_v,s} - \mathbb{E}\textit{Group-EXP3.S}\left[\sum_{s=1}^{S} \Delta x_{\ell_s,s}\right] \\ &\leq 4\Delta\sqrt{VLS\ln(LS)}, \end{aligned} \tag{3.34}$$

where the last inequality uses Lemma 3.

Combining the above inequalities together and choosing $\Delta = \left\lceil \sqrt{\frac{TN\ln N}{VL\ln(TL)}} \right\rceil$, we can derive that

$$\mathbb{E}_{HLMC}[R_{a^T}(T)] \leq 8T^{\frac{3}{4}}V^{\frac{1}{4}}K^{\frac{1}{4}}(\ln(KT))^{\frac{1}{2}} + \sqrt{TVK\ln K}. \tag{3.35}$$

Notice that if $T \geq VK$, the first term on the RHS of (3.35) dominates. Since a^T is chosen arbitrarily with hardness constraint V, we obtain the conclusion in Theorem 2.

It should be noted that to achieve the upper bound established in Theorem 9, the knowledge of V is required in selecting input parameters. When V is unknown, we show in the following theorem that no-regret learning under shifting regret can still be achieved by HLMC in expectation under certain conditions.

Theorem 10 *By selecting $\Delta = \left\lceil \sqrt{\frac{TN\ln N}{L\ln(TL)}} \right\rceil$ and $\gamma_1 = \sqrt{\frac{L\ln(LS)}{S}}$ (the other parameters are selected as specified in Theorem 9), the expected shifting regret of HLMC with a two-level hierarchy under hardness constraint V is upper bounded by:*

$$\mathbb{E}_{HLMC}[R_S(T,V)] \leq (V+3)T^{\frac{3}{4}}K^{\frac{1}{4}}(\ln(KT))^{\frac{1}{2}}. \tag{3.36}$$

If $V = o(T^{1/4})$ as $T \to \infty$, no-regret learning is achieved by HLMC in expectation under shifting regret with hardness constraint V, even with V unknown.

Proof 13 *The proof is similar to that of Theorem 9 and thus, we omit the details.*

3.5 Tradeoff Between Memory Complexity and Regret Performance

In this section, we characterize the tradeoff between the memory complexity and the regret order of the HLMC policy through analyzing its performance with a general D-level hierarchy ($D > 2$). For simplicity, we provide detailed analysis for the case with $D = 3$. All the results can be easily generalized to cases with more than three levels.

We first introduce some notations used in the analysis. The three levels in the hierarchy are referred to as the group, subgroup, and arm levels, respectively. In the first level, the arm set $\mathcal{A}$ is evenly partitioned into N_1 groups $\{\mathcal{A}_\ell\}_{\ell=1}^{N_1}$. Within each group $\mathcal{A}_\ell$, arms are further evenly partitioned into N_2 subgroups $\{\mathcal{B}_h^\ell\}_{h=1}^{N_2}$ in the second level. In the last level, each subgroup $\mathcal{B}_h^\ell$ consists of N_3 arms. We assume without losing generality that the size of every group (subgroup) is identical. It is clear that $K = N_1 N_2 N_3$. Similarly, the time horizon $\mathcal{T}$ is evenly partitioned into S_1 epochs $\{\mathcal{T}_s\}_{s=1}^{S_1}$ and every epoch $\mathcal{T}_s$ is evenly partitioned into S_2 subepochs $\{\mathcal{I}_\tau^s\}_{\tau=1}^{S_1}$. We assume that every sub-epoch consists of S_3 time steps and thus, $T = S_1 S_2 S_3$.

The HLMC policy consists of three selection strategies at the group, subgroup, and arm levels. At the beginning of every epoch $\mathcal{T}_s$, the group-level strategy selects an arm group $\mathcal{A}_{\ell_s}$. The statistics of all sub-groups within $\mathcal{A}_\ell$ are stored in the memory during $\mathcal{T}_s$. Within epoch $\mathcal{T}_s$, the subgroup-level strategy selects a subgroup $\mathcal{B}_{h_\tau}^{\ell_s}$ at the beginning of every subepoch $\mathcal{I}_\tau^s$ and the statistics of arms within $\mathcal{B}_{h_\tau}^{\ell_s}$ are stored in the memory during $\mathcal{I}_\tau^s$. The arm-level strategy is conducted on the selected subgroup to play arms at every time step within

the corresponding subepoch.

It is clear that the size of the memory space required by HLMC with a three-level hierarchy equals $N_1 + N_2 + N_3$. As long as $M \geq 3\lceil K^{1/3} \rceil$, there exists an partition of groups and subgroups satisfying that $N_1 N_2 N_3 \geq K$ and $N_1 + N_2 + N_3 \leq M$. Therefore, the minimum memory space required by HLMC is of order $\Omega(K^{1/3})$. More generally, if we adopt a D-level hierarchy where each level d ($d = 1, 2, ..., D$) consists of N_d level-d groups such that $\prod_{d=1}^{D} N_d \geq K$, the minimum memory complexity of HLMC with D levels is of order $\Omega(DK^{1/D})$. It should be noted that a level-d group should contain at least 2 level-$(d+1)$ groups. As a result, the number D of levels is upper bounded by $\lceil \log_2 K \rceil$ and the minimum memory complexity that the general HLMC learning architecture can achieve is $\Omega(\log_2 K)$.

We show that HLMC with a three-level hierarchy achieves no-regret learning in expectation under the notion of weak regret if we adopt EXP3 at all three levels. Using a similar approach with that in analyzing the regret performance of adopting a two-level hierarchy, we prove an upper bound on the expected weak regret of HLMC in the following theorem.

Theorem 11 *For any T, K, M, suppose $M \geq 3\lceil K^{1/3} \rceil$ and there exists an arm partition with parameters N_1, N_2, N_3 such that $N_1 N_2 N_3 = K$ and $N_1 + N_2 + N_3 \leq M$. Then at every level $i = 1, 2, 3$, by choosing $S_i = \left\lceil \frac{T^{1/3}(N_i \ln N_i)^{2/3}}{(\prod_{j \neq i} N_j \ln N_j)^{1/3}} \right\rceil$ and applying EXP3 with parameter $\gamma_i = \sqrt{\frac{N_i \ln N_i}{2S_i}}$, the expected weak regret of HLMC with a three-level hierarchy against every assignment of the reward sequence is upper bounded by*

$$\mathbb{E}_{HLMC}[Rw(T)] \leq 12 T^{5/6} K^{1/6} (\ln K)^{1/2}. \tag{3.37}$$

Proof 14 *See Appendix B.3.*

For the general HLMC policy with a D-level hierarchy ($2 \leq D \leq \lceil \log_2 K \rceil$), the following corollary on the expected weak regret can be directly derived.

Corollary 3 *If EXP3 is applied to all D levels of the general HLMC policy, the expected weak regret is of order*

$$O(DT^{1-\frac{1}{2D}}K^{\frac{1}{2D}}) \tag{3.38}$$

up to a logarithmic factor, as $T \to \infty$.

Proof 15 *The proof is similar to that of Theorem 7 and 11 and thus, we omit the details.*

Corollary 3 indicates that the regret order of the HLMC policy depends on the depth of the adopted hierarchy: with less available memory, a deeper hierarchy is required, and a larger regret order is induced. In particular, with M available words of memory, we define the minimum number $D^*(M)$ of levels required by HLMC to achieve no-regret learning as:

$$D^*(M) = \min\{D \in \mathbb{N}^+ : D\lceil K^{1/D} \rceil \leq M\}. \tag{3.39}$$

The minimum regret achieved by HLMC with M memory space is of order

$$O\left(D^*(M)T^{1-\frac{1}{2D^*(M)}}K^{\frac{1}{2D^*(M)}}\right). \tag{3.40}$$

In the special case without memory constraints (i.e., $M \geq K$), it is clear that $D^*(M) = 1$ and the HLMC policy with a single-level hierarchy reduces to a classical learning policy for standard adversarial bandit problems. In this case, the regret order achieved by HLMC is $O(\sqrt{KT})$ up to a logarithmic factore, which coincides with the classical results. In another extreme case when $M = \Theta(\log_2 K)$,

the size of available memory space reaches the minimum requirement of a legitimate hierarchy where $D^*(M) = \lceil \log_2 K \rceil$. In this case, the regret order achieved by HLMC is still sublinear in T.

One may notice that the theoretical regret performance of HLMC does not improve when M increases but with $D^*(M)$ unchanged since the dependency of the regret order with respect to the available memory is quantized. However, we show in Sec. 3.6.3 through numerical examples that in certain cases, a better performance can be achieved with a larger memory space, even if the number of levels in the hierarchy is unchanged.

3.6 Numerical Examples

In this section, we illustrate the regret performance of the proposed HLMC policy numerically through simulations. All the experiments are run 10 times using a Monte Carlo method on Python 3.7.

3.6.1 Weak Regret Minimization

We conduct two experiments to compare the regret performance of the HLMC policy with baseline ones under the notion of weak regret. Given that this is the first work on memory-constrained adversarial bandits, we consider two baselines: UCB-M (proposed in [22] for memory constrained stochastic bandits) and EXP3 (for classic adversarial bandits without memory constraints).

We first notice that the only randomness of UCB-M comes from the random

shuffle of arm indices before playing arms, which provides no improvement on the performance in the stochastic setting. Without the random shuffle, UCB-M is purely deterministic and thus, we can easily construct a reward sequence such that UCB-M incurs a regret linear in T. Specifically, in the first experiment, we consider the following setup: let $K = 100$, $M = 20$, and $T = 10^7$. In accordance with the UCB-M policy, we partition the time horizon into phases with length $2^i h_0 b_0$ ($i = 0, 2, ...$) and each phase evenly into h_0 sub-phases with length $2^i b_0$. We select $h_0 = \lceil \frac{K-1}{M-1} \rceil$ and $b_0 = M(M+2)$. For each phase, we assign arm rewards as follows: during each subphase $u = 0, 2, ..., h_0 - 1$, we let arm $(M(u+1) \bmod K)$ offer reward 1 and the other arms offer reward 0. Since UCB-M selects arm groups with size M in a round-robin fashion, it is clear that arms selected by UCB-M offers 0 reward at almost all time steps. The weak regret of UCB-M is clearly linear in T. For HLMC, For HLMC, we adopt a two-level hierarchy and apply EXP3 to both group and arm levels. The simulation results on the expected weak regret are presented in Fig. 3.2.

From Fig. 3.2, we can observe that the proposed HLMC policy outperforms the UCB-M policy under the constructed adversarial environment. The error bar indicates that the proposed learning policy is robust with low variance. Note that although the EXP3 algorithm achieves the best performance, it requires $\Theta(K)$ memory size, which is infeasible in the memory-constrained setting. We also plot the theoretical upper bounds on the regret of HLMC and EXP3 (i.e., $T^{\frac{3}{4}} K^{\frac{1}{4}} (\ln K)^{\frac{1}{2}}$ and $\sqrt{TK \ln K}$ where the constant factors are omitted), which verify that the expected weak regret of HLMC is indeed upper bounded by the theoretical bound established in Theorem 7. It should be noted that the gap between the theoretical and the simulated results is due to the fact that the cumulative reward of the best arm is $T/5$ instead of T in this experiment. There-

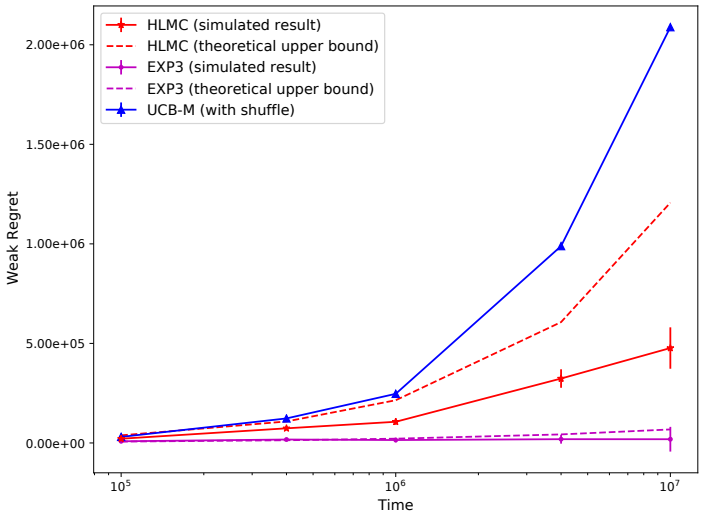

Figure 3.2: Weak regret v.s. time: comparison of UCB-M (without shuffle), HLMC, and EXP3.

fore, the theoretical upper bound is not tight because we use T to upper bound the cumulative reward of the best arm (see the proof of Lemma 1 in Appendix B.1 for details).

We further use another example to show that even with random shuffle, UCB-M still cannot avoid a linear regret in T against adversaries. We consider the same experiment setup with a different reward assignment. Specifically, the phase and subphase partitions are the same with those in the first experiment. During each subphase $u = 0, 2, ..., h_0 - 1$, we let arm 1 offer (u mod 2) reward and the other arms offer $\epsilon = 1 \times 10^{-4}$ rewards. It is not difficult to check that after every time arm 1 is selected by UCB-M and offers reward 1, it will offer 0 reward in the next subphase and will be excluded from the arm memory. As a result, the UCB-M policy suffers a linear regret order in T. For HLMC, we adopt EXP3 at both group and arm levels. The results are shown in Fig. 3.3.

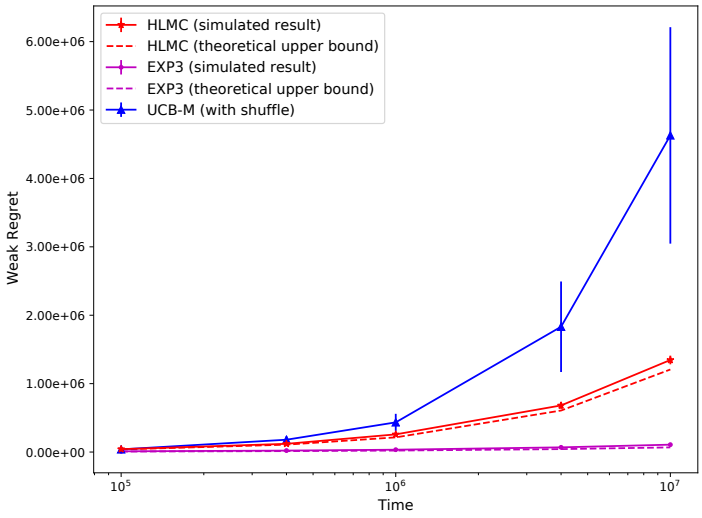

Figure 3.3: Weak regret v.s. time: comparison of UCB-M (with shuffle), HLMC, and EXP3.

The results in Fig. 3.3 illustrate the advantage of HLMC against UCB-M. Moreover, the random shuffle step in UCB-M introduces extremely high variance with little improvement on the expected weak regret. The comparison between the theoretical upper bounds and the simulated results again verifies the correctness of the analysis in Theorem 7.

3.6.2 Shifting Regret Minimization

We further conduct an experiment to show the regret performance of HLMC with a two-level hierarchy under the notion of shifting regret. As discussed in Sec. 3.4.3, by adopting EXP3.S at the group level, HLMC achieves a sublinear scaling of shifting regret in T. In this experiment, we compare the performance of HLMC adopting EXP3.S at the group level and EXP3 at the arm level (referred

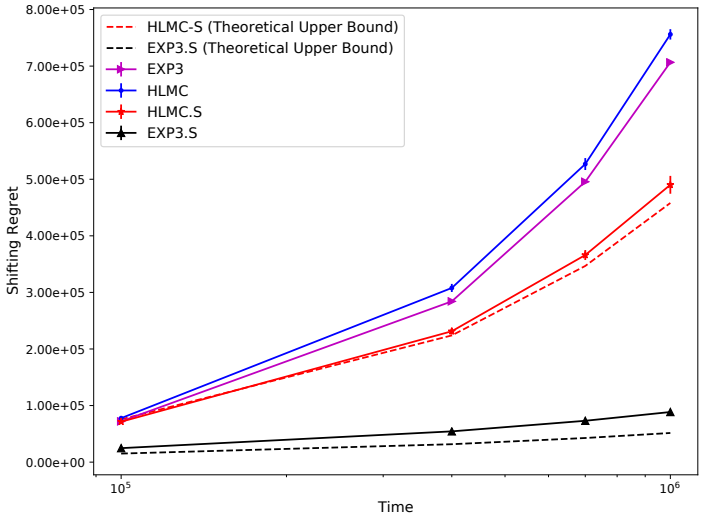

Figure 3.4: Shifting regret v.s. time: comparison of HLMC.S, HLMC, and EXP3.S.

to as HLMC.S in this subsection), HLMC adopting EXP3 at both group and arm levels (referred to as HLMC in this subsection), EXP3, and EXP3.S. The experiment is set up as follows: let $K = 16, M = 8$, and $T = 10^6$. The time horizon is partitioned evenly into $V = 10$ phases. In phase $v = 0, 1, ..., V - 1$, we let arm $i_v = (vN \bmod K)$ offer reward 1 and the other arms offer reward 0 (N is the group size defined in the HLMC algorithm, which equals 4 in this experiment). It is clear that the best benchmark policy in the shifting regret definition with hardness V is to play the best arm i_v within every phase v. The simulation results are presented in Fig. 3.4.

It can be observed from Fig. 3.4 that HLMC.S outperforms HLMC and EXP3, which are designed for weak regret minimization. Adopting EXP3.S at the group level of the HLMC structure improves the regret performance under the notion of shifting regret. Moreover, the error bar verifies the robustness of

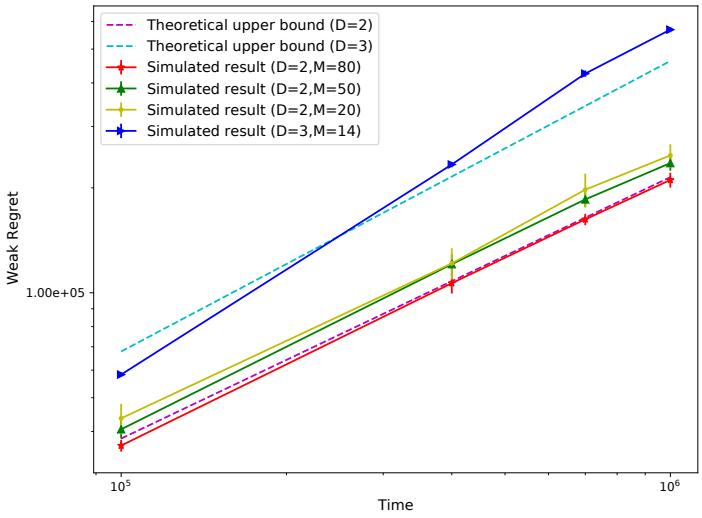

Figure 3.5: Weak regret v.s. time: comparison of HLMC with $M = 14, 20, 50, 80$ memory space.

the proposed policies. It should be noted that although EXP3.S outperforms HLMC and HLMC.S, it requires $\Theta(K)$ memory space, which is inapplicable in the memory-constrained setting.

3.6.3 Impact of Available Memory on Regret Performance

In this subsection, we show the impact of the size of available memory space on the regret performance of HLMC. We use the same experiment setup with that in the first experiment in Sec. 3.6.1. We compare the weak regret of HLMC with $M = 14, 20, 50, 80$. Specifically, when $M = 14$, the HLMC policy adopts a three-level hierarchy with $N_1 = 5, N_2 = 5$, and $N_3 = 4$. When $M = 20, 50, 80$, the HLMC policy adopts two-level hierarchies with $N = \lceil \frac{M - \sqrt{M^2 - 4K}}{2} \rceil$ and $L = \lceil K/N \rceil$.

The results in Fig. 3.5 show that the regret performance of HLMC improves as the size of the memory space increases. In particular, adopting a hierarchy with fewer levels improves the regret order as indicated in Corollary 3. Even with the same number of levels, a smaller regret can be achieved with a larger memory space. Intuitively, as M increases, the group size N and the epoch length Δ decrease. Since the reward sequence assigned in the experiment is stable within a short period but varies vastly in the long run (it has been argued in [70] that such a reward assignment is justified in various real-world applications), the arm-level regret is dominated by the group-level regret and the latter decreases with the epoch length. We also plot the theoretical upper bounds on the regret of HLMC with different levels of hierarchies. The comparison between the theoretical the simulated results verifies our analysis.

CHAPTER 4

BANDITS IN DYNAMIC ENVIRONMENTS

Dynamicity abound in various application domains. A typical example is online recommendation, in which users' interests toward items are dynamically changing and the preference changes are distinct and asynchronous across different items. For example, in news recommendation, changes on the preferences of readers towards different news categories are triggered by the occurrence of related hot events, which are unlikely to happen at the same time. In e-commerce platforms, customers' life-long interests over different products also exhibit distinct changes: a customer is more likely to purchase toys in his childhood while in adulthood, he may become more interested in sport-related products. However, the preference changes over the two categories can happen asynchronously as there may exist a time period (e.g., adolescence) when the customer likes both toys and sports. Moreover, it is possible that the customer's preferences towards other products (e.g., snakes) remain unchanged over time.

To characterize the above phenomena in real applications, we introduce two reward models within the contextual bandit framework in this chapter. Under each model, we develop a learning strategy that adapts to the changing environment. We provide performance guarantees of the proposed algorithms in both theory and practice.

4.1 Literature Review of MAB with Dynamic Reward Models

In addressing the issue of dynamic environments, various reward models have been studied in the bandit literature. The adversarial bandit model discussed in Chapter 3 is a typical example with dynamicity where the reward at every time step is arbitrarily assigned. However, in applications such as online recommendation, users' preferences toward items are usually stationary within a short period of time, but may change abruptly in the long term.

Therefore, a more appropriate model is the piecewise-stationary stochastic reward model, which allows arbitrary changes on the reward distributions at certain unknown time points but remains fixed between two consecutive change points. Under the piecewise-stationary assumption, the problem has been well studied in the classical context-free setting. A number of learning algorithms have been developed that adapts to the abrupt reward changes by either triggering a reset of the learning algorithm after the detected changes [41, 69, 17] or applying a discount factor on past observations [35]. Theoretical regret analysis showed that a sublinear scaling of regret in T is achieved.

Within the contextual bandit setting, however, only a few recent studies have taken the issue of non-stationary environment into consideration. In [40], a contextual Thompson sampling algorithm with a change detection module was proposed but theoretical regret analysis is lacking. In [65], a hierarchical bandit algorithm was developed that detects and adapts to changes by maintaining a suite of contextual bandit models and a regret sublinear in T was proved. However, the existing results assumed a uniform reward model where all arms share a common coefficient vector representing the user interests, which fails to char-

acterize the fact that users' preferences towards different items vary differently. Recently, a so-called context-dependent property was considered in [66] where arms are partitioned into change-invariant and change-sensitive ones based on their context vectors to characterize the distinct reward changes. However, the changes are not completely asynchronous across arms. A more detailed comparison between various models is discussed in the next section.

4.2 Problem Formulation

Consider a contextual bandit problem with K arms and a time horizon of length T. At each time t, a recommender system observes the current user u_t with a d-dimensional feature vector x_{u_t}. A subset $\mathcal{A}_t \subseteq [K]$ of arms is available for selection and each arm $a \in \mathcal{A}_t$ is associated with an m-dimensional feature vector y_a. The system recommends an arm a_t to the user u_t and observes a random reward $r_{u_t,a_t}(t)$ (i.e., clicks, ratings, etc.), which is drawn from an unknown distribution $f(\cdot; x_{u_t}, y_{a_t}, W(t))$ where $W(t) = (w_1(t), ..., w_m(t)) \in \mathbb{R}^{d\times m}$ is a time-varying unknown weight matrix representing the preferences of users towards items in the feature space. The conditional expectation of the reward $r_{u_t,a_t}(t)$ given the feature vectors and the weight matrix is defined as

$$\mathbb{E}[r_{u_t,a_t}(t)|x_{u_t}, y_{a_t}, W(t)] = x_{u_t}^T W(t) y_{a_t}. \tag{4.1}$$

Without loss of generality, we assume that the probability distribution of the random reward $r_{u_t,a_t}(t)$ is sub-Gaussian with parameter σ. The objective is an arm selection policy π that maximizes the expected cumulative reward over the entire time horizon, i.e., $\mathbb{E}[\sum_{t=1}^{T} r_{u_t,\pi_t}(t)]$ where π_t is the arm selected by policy π at time t. Equivalently, we may find a policy π that minimizes the expected

cumulative regret defined as the expected reward loss of policy π against the best policy in the known model case, i.e.,

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} r_{u_t,a_t^*}(t) - r_{u_t,\pi_t}(t)\right], \tag{4.2}$$

where a_t^* is the arm with the largest expected reward at t. It should be noted that the benchmark policy is different from that in the stochastic bandit setting discussed in Chapter 2: the best arm at every time step is specific to the given context information and can be different across time. It is also stronger than the benchmark policies adopted in the adversarial bandit setting discussed in Chapter 3 where a best fixed action or a best action sequence with a hardness constraint are considered. Moreover, in this chapter, we focus on the problem-independent setting in analyzing the regret performance. In particular, the regret is measure under the worst-case assignment (satisfying certain regularity assumptions) of the context vectors of both users and items, the unknown preference matrix, and the resulting reward distributions.

In the stationary scenario where $W(t)$ is fixed over time (i.e., $W(t) \equiv W$), the above formulation is equivalent to the standard contextual bandit model with linear rewards as studied in the literature [8, 24, 5]. Specifically, let $z_{u_t,a} = \text{vec}(x_{u_t} y_a^T)$ be the context vector[1] associated with arm a at time t and $\beta = \text{vec}(W)$ be an unknown preference vector. It is clear that $\mathbb{E}[r_{u_t,a}(t)|x_{u_t}, y_a, W] = z_{u_t,a}^T \beta$. The unknown preference vector β can be efficiently estimated in an online fashion at each time t via ridge regression (see the LinUCB algorithm in [49]), and is applied to the reward estimation and the arm selection at time $t+1$.

In the non-stationary scenario, however, estimating $W(t)$ is in general challenging if elements of $W(t)$ vary arbitrarily: without constraints on the variation

[1] $\text{vec}(\cdot)$ is the vectorization operator that concatenates columns of a matrix to a single vector.

of the parameters, estimating $W(t)$ is impossible. Moreover, to characterize the fact that the preferences of users towards different items vary asynchronously and distinctly, elements of $W(t)$ should exhibit different varying patterns. However, the effects of different elements of $W(t)$ on the obtained rewards are difficult to be distinguished, which leads to the challenge of detecting unknown changes on each element from reward observations. To address the two challenges, we turn to consider approximated reward models to simplify the problem, and adopt certain assumptions on the varying patterns of the reward parameters. Specifically, we study two reward models, i.e., the *disjoint reward model* and the *hybrid reward model*.

4.2.1 Disjoint Reward Model

In the disjoint reward model, we let the combination of $W(t)$ and y_a, i.e., $\theta_a(t) = W(t)y_a$ be the unknown preference vector associated with arm a at time t. The expected reward of recommending item a to user u at time t is then equivalent to the inner product of x_u and $\theta_a(t)$, i.e.,

$$\mathbb{E}[r_{u,a}(t)|x_u, \theta_a(t)] = x_u^T \theta_a(t). \tag{4.3}$$

We adopt a piecewise-stationary assumption on $\theta_a(t)$. To be specific, the time horizon is partitioned into M_a stationary segments with $M_a + 1$ change points $\{\nu_a^{(\ell)}\}_{\ell=0}^{M_a}$ where $\nu_a^{(0)} = 0$ and $\nu_a^{(M_a)} = T$. Within each segment, $\theta_a(t)$ is assumed to be fixed, i.e., $\theta_a(t) \equiv \theta_a^{(\ell)}$, $\forall t \in [\nu_a^{(\ell-1)} + 1, \nu_a^{(\ell)}]$, $0 \leq \ell \leq M_a$. The sequence of changes points may be different across arms, which characterizes the fact that users' preferences towards different items may change asynchronously.

4.2.2 Hybrid Reward Model

In a more general model with hybrid rewards, we further assume that $W(t)$ consists of both a time-varying component $W_v(t)$ and a time-invariant component W_c, i.e., $W(t) = W_v(t) + W_c$. In particular, $W_v(t)$ represents the dynamically changing preferences of users towards items and W_c represents the stationary internal interests of users that are unaffected by the external environment.

For the time-varying component $W_v(t)$, we adopt the same approximation method as the one used in the disjoint setting and define $\theta_a(t) = W_v(t)y_a$ be the arm-specific preference vector of arm a. For the time-invariant component, we define $\beta = \text{vec}(W_c)$ be the joint coefficient vector shared by all arms. It is not difficult to see that the expected reward of recommending arm a to user u at time t satisfies that

$$\mathbb{E}[r_{u,a}(t)|x_u, z_{u,a}, \theta_a(t), \beta] = x_u^T \theta_a(t) + z_{u,a}^T \beta, \tag{4.4}$$

where $z_{u,a} = \text{vec}(x_u y_a^T)$ is a k-dimensional ($k = d \times m$) cross-feature vector of the user-item pair. We adopt the same piecewise-stationary assumption on the arm-specific vectors $\theta_a(t)$ as that assumed in the disjoint setting, which allows asynchronous changes across different arms.

4.2.3 Comparisons with Existing Models

We first compare the two reward models with the stationary ones in the classical contextual bandit setting. It is clear that both models are direct extensions of the stationary reward models studied in [49] where the preference vectors $\theta_a(t), \forall a$ are assumed to be fixed over time. As discussed in the introduction section, it

is more realistic to consider non-stationary preferences in real applications as users' interests are in general time-varying.

In considering the non-stationary environment within the contextual bandit setting, the majority of existing studies [65, 66] assumed a uniform (joint) reward model where all arms share a common coefficient vector $\theta_u(t)$ representing the interests of user u. The expected reward is thus defined as

$$\mathbb{E}[r_{u,a}(t)|y_a, \theta_u(t)] = y_a^T \theta_u(t). \tag{4.5}$$

Notice that the uniform reward model is another approximation of the bilinear model defined in (4.1): $\theta_u(t)$ is the combination of x_u and $W(t)$, i.e., $\theta_u(t) = W^T(t)x_u$. In the literature, $\theta_u(t)$ is assumed to be piecewise-stationary to model the time-varying interests of users. The fact that users' preferences change differently towards different items is, however, not characterized.

The issue was partially addressed in [66] where the so-called *context-dependent* property was considered. It has been assumed that the expected rewards of certain arms are insensitive to the changes of $\theta_u(t)$ (i.e., for some stationary periods i and j, $|y_a^T \theta_u^{(i)} - y_a^T \theta_u^{(j)}| \leq \Delta_L$, where Δ_L is a small constant), while the other arms are change-sensitive. The partition of arms based on their context vectors models the distinct reward changes on different arms. However, the change points across arms are not completely asynchronous: it has been assumed in [66] that between any two stationary periods, there should be a sufficient number of change-sensitive arms undergo perceivable changes to distinguish the two periods. As a result, the user preferences towards a large fraction of arms change simultaneously at the change points of $\theta_u(t)$.

Moreover, we further study a general hybrid reward model consisting of both arm-specific and joint preference vectors that correspond to the time-

varying and the time-invariant interests of users respectively. To the best of our knowledge, the hybrid reward model with dynamically changing user interests has not been studied in the literature.

4.3 PSLinUCB Algorithm in the Disjoint Reward Model

We first consider the disjoint reward model in this section. The key to achieving the objective of minimizing regret under the assumption of piecewise-stationary rewards is to i) estimate the preference vectors accurately, and ii) detect the abrupt changes timely and correctly. We propose a *PSLinUCB (Piecewise-Stationary Linear Upper Confidence Bounds)* algorithm to address the two issues.

To estimate the preference vectors, we adopt a learning structure similar to that of the LinUCB algorithm (proposed in [49] in the stationary contextual bandit setting). In particular, the unknown preference vectors $\theta_a(t), \forall a$ are estimated through ridge regression and can be updated incrementally at each time t. To detect the preference changes timely and correctly, the key technique adopted in the algorithm is to maintain a sliding window for each arm consisting of the most recent reward observations from the arm. If the preference vector learned from observations before the sliding window cannot accurately predict the rewards observed within the window, it is likely that the preference vector has changed. A new model should then be rebuilt based on the observations after the change point.

To be more specific, the estimation and the change detection of the preference vector $\theta_a(t)$ of every arm a can be executed independently in the disjoint reward model. For every arm a, the algorithm maintains a sliding window SW_a

and three different models M_a^{pre}, M_a^{cur}, and M_a^{cum}. The sliding window SW_a of length ω consists of the ω latest observations from arm a (including the observed context vectors and the obtained rewards). M_a^{pre} consists of necessary statistics for estimating the preference vector $\theta_a(t)$. It is learned from observations after the last detected change point and before the sliding window SW_a. Similarly, M_a^{cur} with the same set of statistics is learned from observations within the sliding window, and M_a^{cum} is learned from all observations from the last detected change point to the current time step. In the following subsections, we describe the details of the three models and their usage in the two key components of the PSLinUCB-Disjoint algorithm: (i) parameter estimation and arm selection, and (ii) change detection and model update.

4.3.1 Parameter Estimation and Arm Selection

In each of the three models M_a^{pre}, M_a^{cur}, and M_a^{cum}, the preference vector $\theta_a(t)$ can be estimated by applying ridge regression to the associated set of observations. Without loss of generality, we take M_a^{cum} for an example to illustrate the estimation process. Denote $\{(x_{u_t}, r_{u_t,a})\}_{t \in \mathcal{I}_a^{cum}}$ as the set of observations where $\mathcal{I}_a^{cum}$ is the set of time steps when arm a is played from its last detected change time (initialized to be 0) to the current time step. $\hat{\theta}_a^{cum}$ can be estimated as $\hat{\theta}_a^{cum} = (\mathbf{A}_a^{cum})^{-1}\mathbf{b}_a^{cum}$ where $\mathbf{A}_a^{cum} = \mathbf{I}_d + \sum_{t \in \mathcal{I}_a^{cum}} x_{u_t} x_{u_t}^T$, $\mathbf{I}_d$ is a $d \times d$ identity matrix, and $\mathbf{b}_a^{cum} = \sum_{t \in \mathcal{I}_a^{cum}} r_{u_t,a}(t) x_{u_t}$. The statistics $\mathbf{A}_a^{cum}$ and $\mathbf{b}_a^{cum}$ can be updated incrementally as described in[49].

Based on the estimated preference vector $\hat{\theta}_a^{cum}$ of every arm $a \in \mathcal{A}_t$, we select arms according to the UCB principle to balance the tradeoff between exploration

and exploitation. Similar to the LinUCB algorithm, we define a UCB index for every arm a at time t as $x_{u_t}^T \hat{\theta}_a^{cum} + \alpha \sqrt{x_{u_t}^T (\mathbf{A}_a^{cum})^{-1} x_{u_t}}$. The arm with the greatest index is selected and the reward observations from the selected arm is used to update the corresponding models.

4.3.2 Change Detection and Model Update

To detect potential changes on an arm a, we use M_a^{pre} to predict the rewards of playing arm a at the time steps within the sliding window. We compare the predicted rewards with the observed ones to test if the model learned from earlier data still fits the current observations. To be specific, let $\{(x_s, r_s)\}_{s=1}^{\omega}$ be the set of observations within the sliding window. We test if $\left|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T \hat{\theta}_{a_t}^{pre} - r_s)\right| \geq \delta$, where δ is an input threshold.

If a change is detected on arm a, i.e., the average distance between the predicted rewards and the observed ones in the sliding window exceeds the threshold, we have to restart the learning process of arm a using only observations after the detected change point. Instead of re-constructing a new model without using history data, we exploit the observations within the sliding window again as a warm-start to accelerate learning. In particular, we initialize M_a^{cum}, M_a^{pre}, which are used for arm selection and change detection respectively, with M_a^{cur}, which is the model learned from the latest observations after the change point (i.e., within the sliding window). The sliding window is then emptied to collect new observations until its length reaches ω again.

If no change is detected on arm a, i.e., the earlier and the current reward observations follow the same model, we should keep both sets of data to en-

hance the estimation accuracy. Therefore, M_a^{cum} keeps unchanged and the sliding window is right-shifted by one time step. Note that M_a^{pre} and M_a^{cur} should be updated accordingly after the right-shifting of SW_a.

The detailed implementation of the entire algorithm is summarized in Algorithm 9. Note that the computation complexity in each time step is $O(Kd^3)$ (a finite number of matrix operations for each arm) and the memory size required for learning is $O(K(d^2 + d\omega))$ (three sets of statistics and a sliding window for each arm).

4.4 PSLinUCB Algorithm in the Hybrid Reward Model

In the hybrid reward model, the preference of a user towards an arm a is determined by both an arm-specific preference vector $\theta_a(t)$ and a joint coefficient vector β, which should be estimated simultaneously. Therefore, in addition to a sliding window SW_a and three models M_a^{pre}, M_a^{cur}, and M_a^{cum} for each arm a, the PSLinUCB-Hybrid algorithm maintains two global models G^{pre} and G^{cum} to estimate β. Specifically, G^{pre} is the model learned from the observations from all arms before their sliding windows and is used for change detection. G^{cum} is the model learned from the observations from all arms up to the current time step and is used for arm selection. The statistics in the two global models are obtained by applying ridge regression to the associated data. We omit the tedious theoretical derivations of ridge regression and describe the key process of updating the arm-specific and the global parameters below.

Algorithm 9: **PSLinUCB-Disjoint**

Input: $\alpha > 0, \omega \in \mathbb{N}^+, \delta > 0$.

for $t = 1, 2, ..., T$ **do**

Observe the feature vector x_{u_t} of the current user u_t and the set of available arms $\mathcal{A}_t$.

//Parameter Estimation and Arm Selection

for $a \in \mathcal{A}_t$ **do**

if a is new **then**

$\mathbf{A}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{I}_d$, $\mathbf{b}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times 1}$, $SW_a \leftarrow \emptyset$.

$\hat{\theta}_a^{cum} \leftarrow (\mathbf{A}_a^{cum})^{-1}\mathbf{b}_a^{cum}$.

$p_{t,a} \leftarrow x_{u_t}^T \hat{\theta}_a^{cum} + \alpha \sqrt{x_{u_t}^T (\mathbf{A}_a^{cum})^{-1} x_{u_t}}$.

Play $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$, obtain reward $r_{u_t,a_t}(t)$.

Append $(x_{u_t}, r_{u_t,a_t}(t))$ to the end of SW_{a_t}.

$\mathbf{A}_{a_t}^{\{cur,cum\}} \leftarrow \mathbf{A}_{a_t}^{\{cur,cum\}} + x_{u_t} x_{u_t}^T$.

$\mathbf{b}_{a_t}^{\{cur,cum\}} \leftarrow \mathbf{b}_{a_t}^{\{cur,cum\}} + r_{u_t,a_t}(t) x_{u_t}$.

//Change Detection and Model Update

if $|SW_{a_t}| \geq \omega$ **then**

$\hat{\theta}_{a_t}^{pre} \leftarrow (\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{b}_{a_t}^{pre}$.

Let $SW_{a_t} = \{(x_s, r_s)\}_{s=1}^{\omega}$.

if $|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T \hat{\theta}_{a_t}^{pre} - r_s)| \geq \delta$ **then**

$\mathbf{A}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{A}_{a_t}^{cur}$, $\mathbf{b}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{b}_{a_t}^{cur}$,

$\mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{I}_d$, $\mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times 1}$, $SW_{a_t} \leftarrow \emptyset$.

else

$(x_1, r_1) \leftarrow \text{Popleft}(SW_{a_t})$.

$\mathbf{A}_{a_t}^{pre} \leftarrow \mathbf{A}_{a_t}^{pre} + x_1 x_1^T$, $\mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{A}_{a_t}^{cur} - x_1 x_1^T$,

$\mathbf{b}_{a_t}^{pre} \leftarrow \mathbf{b}_{a_t}^{pre} + r_1 x_1$, $\mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{b}_{a_t}^{cur} - r_1 x_1$.

4.4.1 Parameter Estimation and Arm Selection

By applying ridge regression to the observed data, it can be shown that the joint coefficient vector $\hat{\beta}^{cum}$ is estimated as $\hat{\beta}^{cum} = (\mathbf{A}_0^{cum})^{-1}\mathbf{b}_0^{cum}$ where $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are

coupled with arm-specific parameters $\mathbf{A}_{a_t}^{cum}$, $\mathbf{B}_{a_t}^{cum}$ and $\mathbf{b}_{a_t}^{cum}$. Therefore, the global and the arm-specific parameters should be updated simultaneously. Specifically, $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are initialized to $\mathbf{I}_m, \mathbf{0}_{m\times k}$ respectively and the parameters are updated as follows:

$$\begin{aligned}
\mathbf{A}_0^{cum} &\leftarrow \mathbf{A}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum},\\
\mathbf{b}_0^{cum} &\leftarrow \mathbf{b}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum},\\
\mathbf{A}_{a_t}^{cum} &\leftarrow \mathbf{A}_{a_t}^{cum} + x_{u_t}x_{u_t}^T,\\
\mathbf{B}_{a_t}^{cum} &\leftarrow \mathbf{B}_{a_t}^{cum} + x_{u_t}z_{u_t,a_t}^T,\\
\mathbf{b}_{a_t}^{cum} &\leftarrow \mathbf{b}_{a_t}^{cum} + r_{u_t,a_t}(t)x_{u_t},\\
\mathbf{A}_0^{cum} &\leftarrow \mathbf{A}_0^{cum} + z_{u_t,a_t}z_{u_t,a_t}^T - (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum},\\
\mathbf{b}_0^{cum} &\leftarrow \mathbf{b}_0^{cum} + r_{u_t,a_t}(t)z_{u_t,a_t} - (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum}.
\end{aligned} \tag{4.6}$$

The update procedures of $\mathbf{A}_{a_t}^{cur}$, $\mathbf{B}_{a_t}^{cur}$ and $\mathbf{b}_{a_t}^{cur}$ are similar to the ones of $\mathbf{A}_{a_t}^{cum}$, $\mathbf{B}_{a_t}^{cum}$, and $\mathbf{b}_{a_t}^{cum}$ as described above.

In the arm selection step, we follow [49] to define the UCB index of arm a at time t as $x_{u_t}^T\hat{\theta}_a^{cum} + z_{u_t,a}^T\hat{\beta}^{cum} + \alpha\sqrt{s_{t,a}}$ where $\hat{\theta}_a^{cum} = (\mathbf{A}_a^{cum})^{-1}(\mathbf{b}_a^{cum} - \mathrm{B}_a^{cum}\hat{\beta}^{cum})$. The exploration term $s_{t,a} = s_{t,a}^{(1)} + s_{t,a}^{(2)} + s_{t,a}^{(3)}$ is computed as follows:

$$\begin{aligned}
s_{t,a}^{(1)} &= z_{u_t,a}^T(\mathbf{A}_0^{cum})^{-1}z_{u_t,a} + x_{u_t}^T(\mathbf{A}_a^{cum})^{-1}x_{u_t},\\
s_{t,a}^{(2)} &= -2z_{u_t,a}^T(\mathbf{A}_0^{cum})^{-1}(\mathbf{B}_a^{cum})^T(\mathbf{A}_a^{cum})^{-1}x_{u_t},\\
s_{t,a}^{(3)} &= x_{u_t}^T\mathbf{P}(\mathbf{A}_0^{cum})^{-1}\mathbf{P}^T x_{u_t},
\end{aligned} \tag{4.7}$$

where $\mathbf{P} = (\mathbf{A}_a^{cum})^{-1}\mathbf{B}_a^{cum}$.

4.4.2 Change Detection and Model Update

We conduct a change detection process similar to the one adopted in PSLinUCB-Disjoint to test if the preference vector $\theta_{a_t}(t)$ of arm a_t changes or not. The occurrence of a change on a_t is equivalent to a_t being replaced by a new arm with a different set of arm-specific parameters specified by $\mathbf{A}_{a_t}^{cur}$, $\mathbf{B}_{a_t}^{cur}$, and $\mathbf{b}_{a_t}^{cur}$. As a result, the global parameters $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are coupled with two sets of arm-specific parameters associated with both the old and the new arm. In particular, the original arm-specific parameters (i.e., $\mathbf{A}_{a_t}^{cum}$, $\mathbf{B}_{a_t}^{cum}$, and $\mathbf{b}_{a_t}^{cum}$) used in estimating $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ should be replaced by the aggregation of the parameters corresponding to the old arm (i.e., $\mathbf{A}_{a_t}^{pre}$, $\mathbf{B}_{a_t}^{pre}$, and $\mathbf{b}_{a_t}^{pre}$) and the new arm (i.e., $\mathbf{A}_{a_t}^{cur}$, $\mathbf{B}_{a_t}^{cur}$, and $\mathbf{b}_{a_t}^{cur}$):

$$\begin{aligned}
\mathbf{A}_0^{cum} &\leftarrow \mathbf{A}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum} - (\mathbf{B}_{a_t}^{pre})^T(\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{B}_{a_t}^{pre} - (\mathbf{B}_{a_t}^{cur})^T(\mathbf{A}_{a_t}^{cur})^{-1}\mathbf{B}_{a_t}^{cur} \\
\mathbf{b}_0^{cum} &\leftarrow \mathbf{b}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum} - (\mathbf{B}_{a_t}^{pre})^T(\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{b}_{a_t}^{pre} - (\mathbf{B}_{a_t}^{cur})^T(\mathbf{A}_{a_t}^{cur})^{-1}\mathbf{b}_{a_t}^{cur}.
\end{aligned} \tag{4.8}$$

Moreover, G^{pre} is re-initialized to the updated G^{cum} after the detected change and the arm-specific parameters are updated in the same way with that in the disjoint reward case.

If no change is detected on a_t, the updating process is similar to that in PSLinUCB-Disjoint. Since the overall structure of PSLinUCB-Hybrid is similar to that in the disjoint case, and the detailed implementation of the algorithm is rather lengthy, we present all details in Algorithm 11 in Appendix C.1.

4.5 Theoretical Regret Analysis of a Modified Algorithm

In this section, we introduce a modified version of PSLinUCB and provide an upper bound on regret in the disjoint reward model. While PSLinUCB under both disjoint and hybrid reward models outperform existing baseline algorithms as shown in Sec. 4.6, there are several technical difficulties in analyzing their regret performance directly.

Specifically, a warm start after a detected change by initializing the parameters based on reward observations in the sliding window introduces statistical dependency between parameter estimation and future change detection. In addition, the change detection process exhibits heavy dependency across different time steps since the sliding windows may overlap. Moreover, in the hybrid reward model, estimations of arm-specific parameters and global ones can hardly be decoupled. To avoid such technical difficulties, we make several modifications on the algorithm without changing the learning structure and key strategies, and analyze its regret performance in the disjoint reward model.

4.5.1 Modified PSLinUCB in the Disjoint Reward Model

The modification includes three steps. First, to avoid dependency between the estimation and the change detection of the underlying parameters, the observations in the sliding-window are not re-used for parameter estimation after a detected change. Also, the change detection procedure only uses observations within the sliding window rather than all observations after the last detected change to get rid of heavy dependency across time.

Second, once a change is detected on an arm, the learning procedures of all arms get restarted. Note that this modification is only for the purpose of simplifying the analysis. Its impact on the regret order is rather limited: instead of only re-exploring the arm with reward changes, re-exploring every arm increases the regret order in K. The regret orders in terms of the time length T and the total number of reward changes are unaffected.

Finally, a round-robin exploration step is added to guarantee sufficient exploration of every arm so that reward changes can be detected timely. To further simplify the algorithm design and regret analysis, we assume that the arm set is fixed throughout the time horizon, i.e., $\mathcal{A}_t = \mathcal{A}$. The details of the modified PSLinUCB-Disjoint algorithm are summarized below in Algorithm 10.

4.5.2 Regret Analysis

Before providing the theoretical regret analysis, we first introduce some notations. Let M be the number of total piecewise-stationary segments, i.e.,

$$M = 1 + \sum_{t=1}^{T-1} \mathbb{I}(\theta_a(t) \neq \theta_a(t-1) \text{ for some } a \in \mathcal{A}). \tag{4.9}$$

Let $\{\nu_i\}_{i=0}^{M}$ be the change points where $\nu_0 = 0, \nu_M = T$. Define $L = \omega\lceil K/\gamma \rceil$ where ω, γ are input parameters of the modified PSLinUCB policy. Let $\Delta_a^{(i)}(x)$ be the amplitude of the preference change of a user with feature vector x towards an arm a at the i-th change point, i.e.,

$$\Delta_a^{(i)}(x) = \left| x^T \theta_a(\nu_i + 1) - x^T \theta_a(\nu_i) \right|. \tag{4.10}$$

Without loss of generality, we assume that the sub-Gaussian parameter σ in the distribution of the random reward is 1 and $\|\theta_a(t)\|_2 \leq 1, \|x_{u_t}\|_2 \leq 1, \forall t, \forall a \in \mathcal{A}$.

Algorithm 10: Modified PSLinUCB-Disjoint

Input: $\alpha > 0, \omega \in \mathbb{N}^+, b, c > 0, \gamma > 0$, and the arm set $\mathcal{A}$.

Initialization: $\tau \leftarrow 0, \mathbf{A}_a^{cum} \leftarrow \mathbf{I}_d, \mathbf{b}_a^{cum} \leftarrow \mathbf{0}_{d\times 1}$, $\mathrm{SW}(a) \leftarrow \emptyset, \forall a \in \mathcal{A}$.

for $t = 1, 2, ..., T$ **do** do

//Round-Robin Exploration

Let $a = (t - \tau) \bmod \lfloor K/\gamma \rfloor$.

if $a \leq K$ **then**

Play arm $a_t = a$.

else

//Parameter Estimation and Arm Selection

Observe the feature vector x_{u_t} of the current user u_t.

for $a \in \mathcal{A}$ **do** do

$\hat{\theta}_a \leftarrow (\mathbf{A}_a^{cum})^{-1}\mathbf{b}_a^{cum}$.

$p_{t,a} \leftarrow x_{u_t}^T\hat{\theta}_a + \alpha\sqrt{x_{u_t}^T(\mathbf{A}_a^{cum})^{-1}x_{u_t}}$.

Play $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$, obtain reward r_{u_t,a_t}.

Append $(x_{u_t}, r_{u_t,a_t}(t))$ to the end of $SW(a_t)$.

$\mathbf{A}_{a_t}^{cum} \leftarrow \mathbf{A}_{a_t}^{cum} + x_{u_t}x_{u_t}^T$, $\mathbf{b}_{a_t}^{cum} \leftarrow \mathbf{b}_{a_t}^{cum} + r_{u_t,a_t}x_{u_t}$.

//Change Detection and Model Update

if $|SW_{a_t}| \geq \omega$ **then**

Let $SW_{a_t} = \{(x_s, r_s)\}_{s=1}^{\omega}$.

$\mathbf{A}_{a_t}^{pre} = \sum_{s=1}^{\lfloor\omega/2\rfloor} x_s x_s^T$, $\mathbf{b}_{a_t}^{pre} = \sum_{s=1}^{\lfloor\omega/2\rfloor} r_s x_s^T$, $\hat{\theta}_{a_t}^{pre} \leftarrow (\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{b}_{a_t}^{pre}$.

if $|\frac{2}{\omega}(\sum_{s=\lfloor\omega/2\rfloor+1}^{\omega} x_s^T\hat{\theta}_{a_t}^{pre} - r_s)| \geq b + c$ **then**

$\forall a \in \mathcal{A} : \mathbf{A}_a^{cum} \leftarrow \mathbf{I}_d, \mathbf{b}_a^{cum} \leftarrow \mathbf{0}_{d\times 1}, SW_a \leftarrow \emptyset, \tau \leftarrow t$.

Moreover, to guarantee that the reward changes are discernible to the learning process, we further assume that the lengths of stationary segments and the magnitude of reward changes are sufficiently large.

Assumption 6 *Assume that* $v_{i+1} - v_i \geq L, \forall 1 \leq i \leq M-1$ *and* $v_1 \geq L/2$.

Assumption 7 *Assume that there exists* $\Delta > 0$ *such that for every user vector* x *and change point* i, $\Delta_a^{(i)}(x) \geq \Delta$.

We provide an upper bound on regret of the modified PSLinUCB-Disjoint algorithm in the following theorem.

Theorem 12 *Suppose Assumptions 6 and 7 holds. With appropriate choices of the input parameters, the cumulative regret of the modified PSLinUCB algorithm under the disjoint reward model satisfies:*

$$R(T) \leq \tilde{C}_1 \sqrt{TMK\omega} + \tilde{C}_2 \sqrt{TMKd^2 \log^2 T}, \tag{4.11}$$

where $\tilde{C}_1, \tilde{C}_2$ *are constants independent of* T, M, *and* K.

Proof 16 *See the Appendix C.2.*

Remark 6 *The cumulative regret achieved by the modified PSLinUCB-Disjoint algorithm has a sublinear scaling in* T *and* M, *i.e.,* $R(T) \sim \tilde{O}(\sqrt{MT})$ *where the* $\tilde{O}$ *notation hides the logarithmic factor. In other words, the average regret per time step diminishes to zero as* $T \to \infty$ *if* $M \sim o(T)$. *Moreover, if we assume that* M, K *are constants, the regret order in* T *is optimal up to a logarithmic factor since the lower bound on regret in the stationary setting is* $\Omega(\sqrt{T})$ *[24].*

We present here a sketch of the proof based on three key lemmas as presented below. We first consider a stationary scenario where the preference vector $\theta_a(t)$ is fixed for all $a \in \mathcal{A}$.

Lemma 5 *Consider a stationary scenario with $M = 1$. For any $\delta_0 \in (0, 1)$ and $\alpha > \sqrt{2d \log \frac{T}{\delta_0}}$, the expected cumulative regret of the modified PSLinUCB algorithm is upper bounded as follows:*

$$\mathbb{E}[R(T)] \leq T\mathbb{P}(\tau_1 \leq T) + (\delta_0 + \gamma)T + K + 2\alpha\sqrt{2TdK \log \frac{T}{d}}, \tag{4.12}$$

where τ_1 is the first detection time.

Proof 17 *See Appendix C.3.*

Second, we upper bound the probability of raising false alarms, i.e., changes are detected in the stationary environment.

Lemma 6 *Consider a stationary scenario with $M = 1$ and let $\delta_1 = 1/(2T^2)$, the probability of false alarm is upper bounded by*

$$\mathbb{P}(\tau_1 \leq T) \leq KT^{-1}. \tag{4.13}$$

if the thresholds b, c are chosen to satisfy (C.22) (in Appendix C.4) for all $a \in \mathcal{A}$ and $c \geq \sqrt{\frac{2}{\omega} \log(2T)}$.

Proof 18 *See Appendix C.4.*

We further upper bound the probability of a late detection.

Lemma 7 *Consider a piecewise-stationary scenario with $M \geq 2$. Assume that $\Delta \geq b + c$. Then we have*

$$\mathbb{P}(\tau_1 > \nu_1 + L/2) \leq 2T^{-2}. \tag{4.14}$$

Proof 19 *See Appendix C.5.*

Theorem 12 can be proved based on the above three lemmas. The detailed proof is presented in Appendix C.2.

4.6 Numerical Examples

We use both synthetic and real-world data to evaluate the performance of the proposed learning algorithms under the disjoint and the hybrid reward models.

4.6.1 Regret Analysis on Synthetic Data

We first use synthetic data to compare the regret performance of the proposed learning algorithms with LinUCB, a representative algorithm for stationary contextual bandits [49, 24]. There are three versions of LinUCB corresponding to three different models with uniform, disjoint, and hybrid rewards. We compare the proposed algorithms with the disjoint and the hybrid versions of LinUCB under the corresponding reward models.

In the first experiment, we generate a dataset under the disjoint reward model. Specifically, we assume a time horizon of length $T = 20000$. We randomly generate $K = 10$ arms. Each arm a is associated with a m-dimensional ($m = 5$) feature vector y_a with $||y_a||_2 \leq 1$. We consider a single user setting where a user u is associated with a d-dimensional ($d = 5$) feature vector x_u with $||x_u||_2 \leq 1$. The d-dimensional preference vectors $\theta_a(t), \forall a$ are randomly generated satisfying the piecewise-stationary assumption (the preference vector $\theta_a(t)$ changes every 2000 time steps) and $||\theta_a(t)||_2 \leq 1$. The reward of playing an arm a at time t is generated according to the disjoint reward model, i.e., $r_a(t) = x_u^T \theta_a(t) + \epsilon$, where

ϵ is a Gaussian noise with $\mu = 0$ (mean) and $\sigma = 0.2$ (standard deviation).

We compare the cumulative regret of PSLinUCB-Disjoint and LinUCB-Disjoint. To guarantee a fair comparison, the parameters α balancing the trade-off between exploration and exploitation in the UCB indices of the two algorithms are equal ($\alpha = 1$). In PSLinUCB-Disjoint, we set $\omega = 100$ and $\delta = 0.35$. The experiment is run 100 times and the simulation results are included in Fig. 4.1. It can be seen that the PSLinUCB-Disjoint algorithm adapts to the changing environment and achieves a lower cumulative regret (30% performance gain).

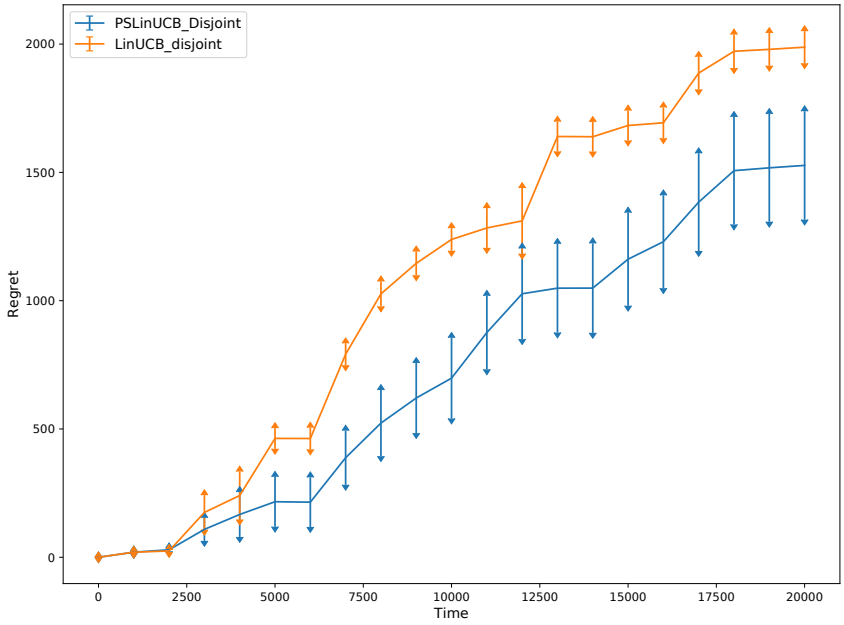

Figure 4.1: Regret v.s. time under the disjoint reward model.

In the second experiment, we consider the hybrid reward model. In addition to the parameters generated in the first experiment, we further construct an $m \times d$-dimensional joint preference vector β. The random reward of playing an

arm a at time t is generated according to the hybrid reward model, i.e., $r_a(t) = x_u^T \theta_a(t) + z_{u,a}^T \beta + \epsilon$, where $z_{u,a} = \text{vec}(x_u y_a^T)$ and ϵ is a Gaussian noise with $\mu = 0$ and $\sigma = 0.2$. We compare the regret performance of PSLinUCB-Hybrid and LinUCB-Hybrid with $\alpha = 1.5$. In PSLinUCB-Hybrid, we set $\omega = 100$ and $\delta = 0.4$. The experiment is also run 100 times and the simulation results are included in Fig. 4.2. Similar performance gain can be observed.

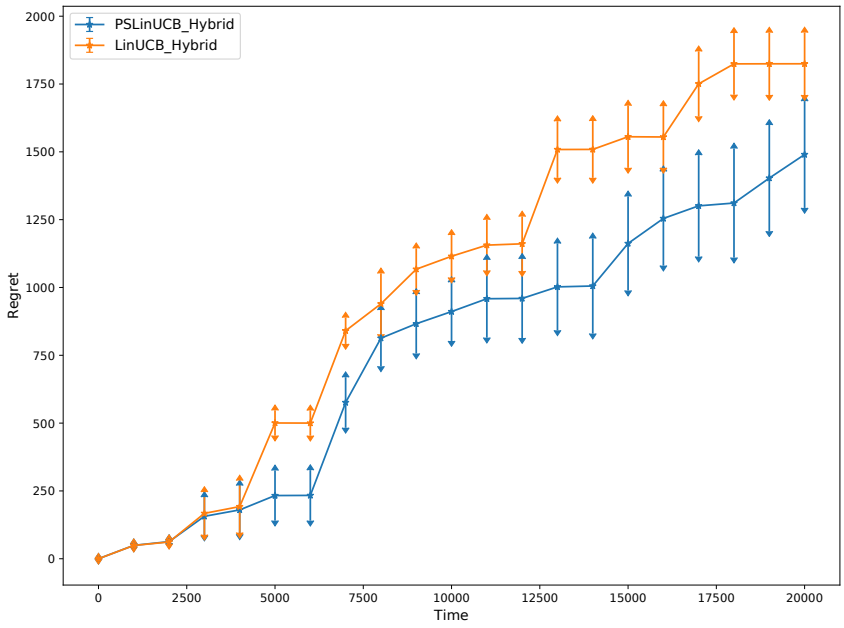

Figure 4.2: Regret v.s. time under the hybrid reward model.

4.6.2 Recommendation Performance on Real-World Datasets

We use two real-world datasets to evalute the recommendation performance of the proposed algorithms. The first dataset is a collection of user-visit log infor-

mation from Yahoo! front page, which is widely used for algorithm evaluation in the contextual bandit setting [49, 50]. The Yahoo! dataset contains 45,811,883 user-visits to Yahoo Today Module in a ten-day period in May 2009. The log information of each user-visit includes a feature vector of the current user, a pool of candidate articles (arms) for recommendation associated with feature vectors, the recommended article, and the feedback from the user (click or not). It has been observed in [66] that the preferences of users towards different items are dynamically changing in this dataset.

The second dataset is extracted from the Last.fm online music system, which is made available on the HetRec 2011 workshop. This dataset contains 1892 users, 17,632 artists (arms), and 92,834 user-artist listening records. Each user may assign multiple tags to the listened artists, which can be preprocessed as the context information to fit into the contextual bandit setting. Following [42], a non-stationary environment can be simulated.

Except LinUCB, we further compare the proposed learning algorithms with the following baselines:

1. *Random*: a policy that selects arms uniformly at random.
2. *UCB* [9]: one of the most well-known algorithms developed in the stationary context-free bandit setting.
3. *MUCB* [17]: an extension of UCB to the context-free setting with piecewise-stationary rewards.
4. *DenBand* [66]: a new algorithm developed under the uniform reward model with piecewise-stationary rewards. Under the assumption of continuous rewards with little noise, the original algorithm only compares

the predicted reward at a single time step with the observed one to detect potential changes. In cases with larger noise (e.g., binary rewards), we modify the algorithm by using observations at multiple time steps for change detection.

Yahoo! Dataset

We randomly sample a subset of data from the original dataset for testing (i.e., each user-visit is selected independently with probability 0.1). We adopt an unbiased offline evaluation method proposed in [49, 50] to evaluate the online performance of the proposed learning algorithms and the baseline ones.

The detailed recommendation performance (i.e., CTR) of the proposed algorithms along with baseline ones are summarized in Table 4.1. In PSLinUCB-Disjoint, we set $\alpha = 0.2$, $\omega = 1000$, and $\delta = 0.025$. In PSLinUCB-Hybrid, we set $\alpha = 0.15$, $\omega = 1200$, and $\delta = 0.03$. In addition to the comparison results discussed in the main file, PSLinUCB-Disjoint and PSLinUCB-Hybrid achieves a performance gain of 59.2% and 61.2% compared with the Random policy, which does not learn from the observation history.

Stationary		Non-Stationary	
Algorithm	CTR	Algorithm	CTR
Random	0.03541	/	/
UCB	0.04002	MUCB	0.04058
LinUCB-uniform	0.04121	DenBand	0.04353
LinUCB-Disjoint	0.05491	PSLinUCB-Disjoint	**0.05639**
LinUCB-Hybrid	0.05638	PSLinUCB-Hybrid	**0.05711**

Table 4.1: Comparison of CTR on Yahoo dataset.

We also illustrate the simulation results in Figure 4.3. We first observe that al-

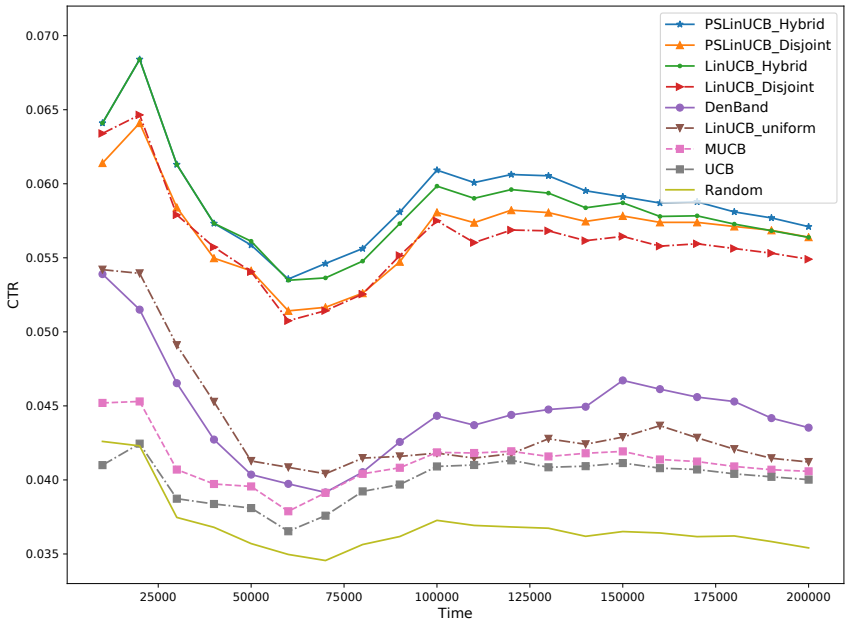

Figure 4.3: Average CTR v.s. time in the Yahoo! dataset.

gorithms exploiting the context information (i.e., PSLinUCB, LinUCB, and DenBand) outperform context-free ones (i.e., UCB and MUCB). This observation is rather intuitive since context vectors provide significant side information on the preferences of users towards items. In addition, under each reward model (i.e., classical context-free bandits and contextual bandits with uniform, disjoint, and hybrid rewards), the algorithm that adapts to reward changes outperforms the one that does not (i.e., MUCB v.s. UCB, DenBand v.s. LinUCB-uniform, PSLinUCB-Disjoint v.s. LinUCB-Disjoint, and PSLinUCB-Hybrid v.s. LinUCB-Hybrid). In particular, PSLinUCB-Disjoint achieves a performance gain of 2.7% (2.9% at the peak) against LinUCB-Disjoint and PSLinUCB-Hybrid achieves an improvement of 1.3% (2% at the peak) against LinUCB-Hybrid. The comparison results verify the assumption that users' interests are dynamically changing and should be taken into consideration in learning.

Moreover, within the contextual bandit setting, algorithms developed under the hybrid reward model (i.e., PSLinUCB-Hybrid and LinUCB-Hybrid) or the disjoint reward model (i.e., PSLinUCB-Disjoint and LinUCB-Disjoint) achieve better performance compared with the ones developed under the uniform reward model (i.e., DenBand and LinUCB-Uniform). This is because the uniform reward model fails to exploit the personalized interests of different users. An alternative approach is to learn the preferences of every user individually. However, the amount of data associated with a single user is rather limited. Furthermore, the performance gain of PSLinUCB over DenBand (31.2% under the hybrid model and 29.5% under the disjoint model) verifies the fact that users' preferences towards different items vary differently.

LastFM Dataset

Given that the original LastFM dataset dose not provide context vectors of neither users nor items, we first preprocess the dataset to fit into the contextual bandit setting. Specifically, following the settings in [19, 66], we treat the 'listened artists' of each user as positive feedback. For each artist, we use its associated tags to create a TF-IDF feature vector and then apply PCA to reduce the dimension to 10. For each user, we adopt a method similar to the one used in [49] to generate a feature vector: we use matrix factorization to obtain a raw feature vector and then apply the K-means method to group users into 10 clusters. The final user feature is a 10-dimensional vector corresponding to the soft-membership of the user in the 10 clusters (computed with a Gaussian kernel and then normalized). In the experiment, we only consider artists that have been listened by at least 100 users and we follow [65] to generate the log data.

We summarize the simulation results of the proposed algorithms and their corresponding opponents in the stationary setting in Table 4.2. Note that in PSLinUCB-Disjoint, $\alpha = 0.15, \omega = 1200, \delta = 0.035$. In PSLinUCB-Hybrid, $\alpha = 0.2$, $\omega = 1000$, $\delta = 0.02$.

Stationary		Non-Stationary	
Algorithm	CTR	Algorithm	CTR
LinUCB-Disjoint	0.03341	PSLinUCB-Disjoint	0.03408
LinUCB-Hybrid	0.04046	PSLinUCB-Hybrid	**0.04143**

Table 4.2: Comparison of CTR on LastFM dataset.

The results are also presented in Figure 4.4 and similar conclusions with those in the experiment on the Yahoo! dataset can be drawn. In particular, PSLinUCB-Disjoint achieves a performance gain of 2% against LinUCB-Disjoint and PSLinUCB-Hybrid achieves a performance gain of 2.4% against LinUCB-Hybrid, which again verify the advantages of the proposed algorithms.

Sensitivity Analysis

We further test the sensitivity of the proposed algorithms against hyper-parameters: ω and δ on both the Yahoo! dataset and the LastFM dataset. Since the effect of users' changing interests on the recommendation performance emerges after a sufficient time of learning, we use the first 1/2 of the Yahoo dataset and the entire LastFM dataset for testing. From the results shown in Figure 4.5 and Figure 4.6, we observe that both PSLinUCB-Disjoint and PSLinUCB-Hybrid are relatively robust towards the change of the hyper-parameter within certain ranges.

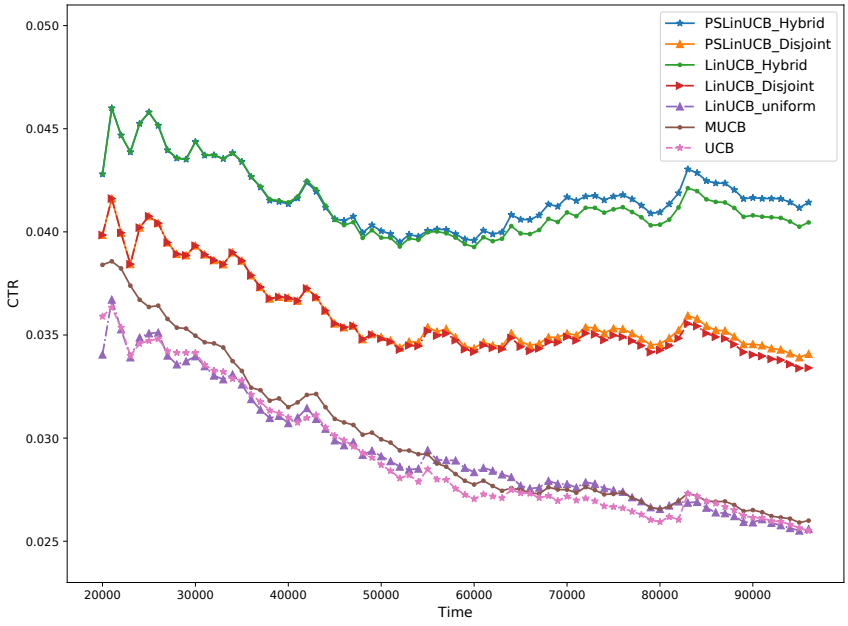

Figure 4.4: Average CTR v.s. time in the LastFM dataset.

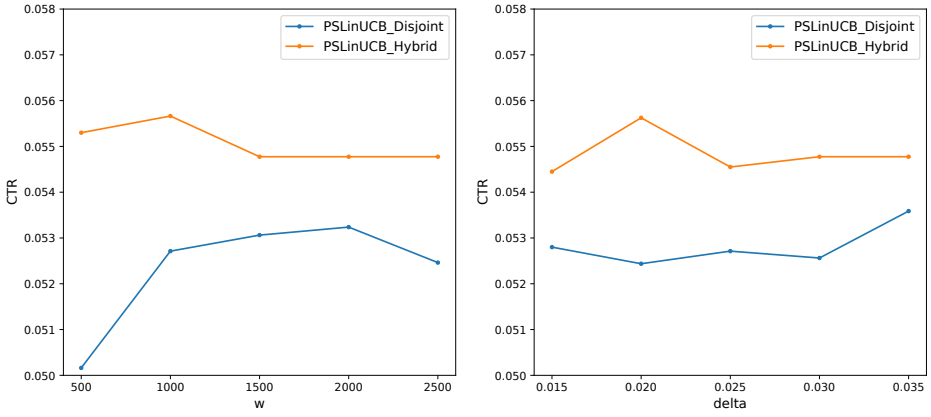

Figure 4.5: Sensitivity analysis on Yahoo! dataset.

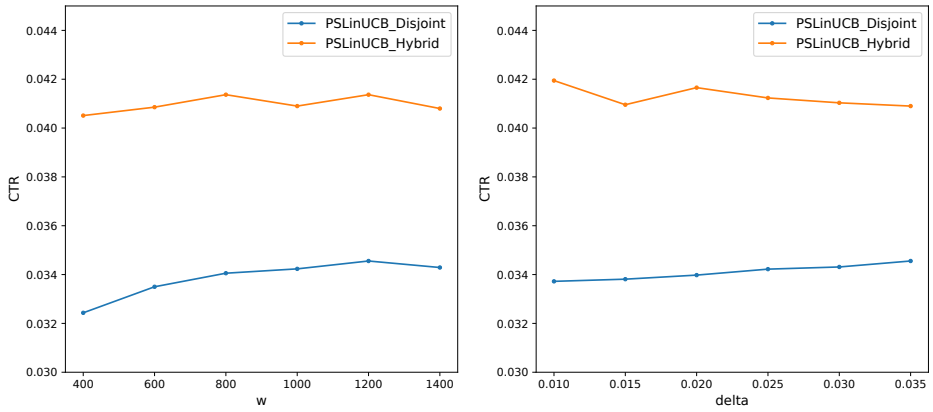

Figure 4.6: Sensitivity analysis on LastFM dataset.

CHAPTER 5

CONCLUSION

This dissertation focused on the online learning problem within the framework of multi-armed bandits. Three emerging issues in terms of the massive number of actions, memory constraints on learning strategies, and dynamicity in reward models were studied under various bandit models.

In the first part of the dissertation, we studied a stochastic multi-armed bandit problem with side information on the similarity and dissimilarity across arms to address the issue of a large action space. The similarity-dissimilarity structure is represented by a UIG where every node represents an arm and the presence (absence) of an edge between two nodes represents similarity (dissimilarity) of their mean rewards. We considered two settings with complete and partial side information based on whether the UIG is fully revealed, and proposed a general two-step learning structure: LSDT consisting of an offline reduction of the action space to the candidate set and online aggregation of observations from similar arms. Theoretical regret analysis along with matching lower bounds in both settings showed the order optimality of LSDT in both the size of the action space and the length of the time horizon. Extensive simulation experiments were conducted to verify the performance of LSDT numerically.

In the second part of the dissertation, we studied the problem of adversarial multi-armed bandits with memory constraints. We proposed a general hierarchical learning architecture that adopts a multi-level hierarchy to partition the arm set into groups and the time horizon into epochs. By adopting appropriate selection strategies as subroutines at all levels, we showed that the proposed HLMC policy achieves no-regret learning under two regret notions using

a memory space with size sublinear in the number of arms. We further characterize the tradeoff between the regret order and the memory complexity by establishing a memory-dependent regret bound of HLMC. We conducted numerical experiments to verify the advantages of the proposed learning policies against existing baselines.

In the third part of the dissertation, we studied a contextual bandit problem for personalized recommendation in a non-stationary environment. To characterize the fact that users' interests towards different items vary asynchronously and distinctly, two models with disjoint and hybrid piecewise-stationary rewards were considered. For each model, we proposed a PSLinUCB learning algorithm that adapts to the changing environment via change detection and restart. We further introduced a modified version of the learning algorithm with theoretical analysis validating a near-optimal regret order in the time length under the disjoint reward model. Numerical results on both synthetic data and real-world datasets verified the advantages of the proposed learning algorithms against baseline ones.

APPENDIX A

PROOFS OF LEMMAS AND THEOREMS IN CHAPTER 2

A.1 Proof of Theorem 1

We first show that $\mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}} \subseteq \mathcal{B}^*$. Clearly $i_{\max} \in \mathcal{B}^*$. For each $j \in \mathcal{B}^*_{i_{\max}}$, $\mathcal{N}[j] = \mathcal{N}[i_{\max}]$. Thus if we construct a new set of mean rewards $(\mu'_1, ..., \mu'_K)$ where the mean values of j and $i_{\max}$ get switched and the others remain the same, the UIG $\mathcal{G}^*_\epsilon$ remains unchanged. Thus, $j \in \mathcal{B}^*$. Similar result holds for $\mathcal{B}^*_{i_{\min}}$. Therefore $\mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}} \subseteq \mathcal{B}^*$.

Next, we show that $\mathcal{B}^* \subseteq \mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}}$. For each $j \notin \mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}}$, consider two cases:

1. $j \in \mathcal{N}[i_{\max}] \cup \mathcal{N}[i_{\min}]$: without loss of generality, assume that $j \in \mathcal{N}[i_{\max}]$. Since $j \notin \mathcal{B}^*_{i_{\max}}$, there exists an arm k such that $k \in \mathcal{N}[j]$ but $k \notin \mathcal{N}[i_{\max}]$. Now suppose there exists an assignment of mean rewards $(\mu'_1, ..., \mu'_K)$ conforming to $\mathcal{G}^*_\epsilon$ such that arm j is optimal, then $\mu'_k, \mu'_{i_{\max}} \in (\mu'_j - \epsilon, \mu'_j]$ and thus, arm k and $i_{\max}$ are neighbors. This contradicts the assumption that $k \notin \mathcal{N}[i_{\max}]$. Hence, there doesn't exists a set of mean rewards conforming to $\mathcal{G}^*_\epsilon$ where j is optimal. Thus $j \notin \mathcal{B}^*$. Similar result holds for the case when $j \in \mathcal{N}[i_{\min}]$.

2. $j \notin \mathcal{N}[i_{\max}] \cup \mathcal{N}[i_{\min}]$: define

$$k_1 = \underset{k \notin \mathcal{N}[j], \mu_k > \mu_j}{\operatorname{argmin}} \mu_k, \tag{A.1}$$

$$k_2 = \underset{k \notin \mathcal{N}[j], \mu_k < \mu_j}{\operatorname{argmax}} \mu_k. \tag{A.2}$$

Notice that k_1, k_2 are not neighbors. However, since the component is connected, k_1, k_2 must connect with arms in $\mathcal{N}[j]$. Now suppose there exists an assignment of mean rewards $(\mu'_1, ..., \mu'_K)$ conforming to $\mathcal{G}^*_\epsilon$ such that j is optimal, then $\mu'_{k_1}, \mu'_{k_2} \in (\mu'_j - 2\epsilon, \mu'_j - \epsilon]$. This contradicts the assumption that k_1, k_2 are not neighbors. Thus, $j \notin \mathcal{B}^*$.

Therefore, we have that if $j \notin \mathcal{B}^*{}_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}}$, then $j \notin \mathcal{B}^*$. This implies that $\mathcal{B}^* \subseteq \mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}}$. In summary,

$$\mathcal{B}^* = \mathcal{B}^*_{i_{\max}} \cup \mathcal{B}^*_{i_{\min}}. \tag{A.3}$$

A.2 Proof of Theorem 2

When $\mathcal{G}^*_\epsilon$ is connected, $\mathcal{B}^* = \mathcal{B}^*_{i_{\min}} \cup \mathcal{B}^*_{i_{\max}}$ where $\mathcal{B}^*_{i_{\min}}$ and $\mathcal{B}^*_{i_{\max}}$ are disjoint if $\mathcal{G}^*_\epsilon$ is not complete. We upper bound the number of times that arms in $\mathcal{B}^*_{i_{\min}}$ have been played up to time T. Let $\tau_{\mathcal{B}^*_{i_{\min}}}(T) = \sum_{j \in \mathcal{B}^*_{i_{\min}}} \tau_j(T)$, $\tau_{\mathcal{B}^*_{i_{\max}}}(T) = \sum_{j \in \mathcal{B}^*_{i_{\max}}} \tau_j(T)$. Let $c_{t,s} = \sqrt{(8 \log t)/s}$. Let π_t be the arm selected at time t and $\mathbb{I}\{\cdot\}$ be the indicator function. Let $\ell > |\mathcal{B}^*_{i_{\min}}|$ be an arbitrary integer, then with $H_i(t)$ defined in (2.8),

$$\begin{aligned}
\mathbb{E}[\tau_{\mathcal{B}^*_{i_{\min}}}(T)] &= \mathbb{E}\left[|\mathcal{B}^*_{i_{\min}}| + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{I}\{\pi_t \in \mathcal{B}^*_{i_{\min}}\}\right] \\
&\leq \ell + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{P}\left(\pi_t \in \mathcal{B}^*_{i_{\min}}, \tau_{\mathcal{B}^*_{i_{\min}}}(t-1) \geq \ell\right) \\
&\leq \ell + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{P}\Big(H_{i_{\min}}(t-1) \geq H_{i_{\max}}(t-1), \tau_{\mathcal{B}^*_{i_{\min}}}(t-1) \geq \ell\Big) \\
&\leq \ell + \sum_{t=|\mathcal{B}^*|}^{T-1} \sum_{s=\ell}^{t} \sum_{r=1}^{t} \mathbb{P}(H_{i_{\min}}(t) \geq H_{i_{\max}}(t), \tau_{\mathcal{B}^*_{i_{\min}}}(t) = s, \tau_{\mathcal{B}^*_{i_{\max}}}(t) = r).
\end{aligned} \tag{A.4}$$

To upper bound each term on the RHS of the last inequality in (A.4), we consider

$$\begin{aligned}
&\mathbb{P}\left(\frac{\sum_{j\in\mathcal{B}^*_{i_{\min}}}\tau_j(t)\bar{x}_j(t)}{s}+c_{t,s}\geq\frac{\sum_{j\in\mathcal{B}^*_{i_{\max}}}\tau_j(t)\bar{x}_j(t)}{r}+c_{t,r}\right)\\
\leq\ &\mathbb{P}\left(\frac{\sum_{j\in\mathcal{B}^*_{i_{\min}}}\tau_j(t)\bar{x}_j(t)}{s}\geq\frac{\sum_{j\in\mathcal{B}^*_{i_{\min}}}\tau_j(t)\mu_j}{s}+c_{t,s}\right)\\
&+\mathbb{P}\left(\frac{\sum_{j\in\mathcal{B}^*_{i_{\max}}}\tau_j(t)\bar{x}_j(t)}{r}\leq\frac{\sum_{j\in\mathcal{B}^*_{i_{\max}}}\tau_j(t)\mu_j}{r}-c_{t,r}\right)\\
&+\mathbb{P}\left(\frac{\sum_{j\in\mathcal{B}^*_{i_{\max}}}\tau_j(t)\mu_j}{r}<\frac{\sum_{j\in\mathcal{B}^*_{i_{\min}}}\tau_j(t)\mu_j}{s}+2c_{t,s}\right),
\end{aligned}\tag{A.5}$$

where $\tau_{\mathcal{B}^*_{i_{\min}}}(t)=s,\tau_{\mathcal{B}^*_{i_{\max}}}(t)=r$. The inequality holds because the event on the LHS indicates that at least one of the three events on the RHS happens. To upper bound the first term, let $Z_t=\sum_{j\in\mathcal{B}^*_{i_{\min}}}\mathbb{I}\{\pi_t=j\}X_j(t)$, where $X_j(t)$ is the random reward from arm j at time t. Let $\nu_t=\sum_{j\in\mathcal{B}^*_{i_{\min}}}\mathbb{I}\{\pi_t=j\}\mu_j$. Note that if $\pi_t\notin\mathcal{B}^*_{i_{\min}}$, $Z_t=\nu_t=0$. Consider the first term on the RHS of (A.5):

$$\mathbb{P}\left(\frac{\sum_{\tau=1}^t(Z_\tau-\nu_\tau)}{s}\geq\sqrt{\frac{8\log t}{s}},\tau_{\mathcal{B}^*_{i_{\min}}}(t)=s\right)\leq\mathbb{P}\left(\mathbb{I}\{\tau_{\mathcal{B}^*_{i_{\min}}}(t)=s\}\cdot e^{\lambda\sum_{\tau=1}^t(Z_\tau-\nu_\tau)}\geq e^{\lambda\sqrt{8s\log t}}\right),\tag{A.6}$$

Using the Markov inequality, we have

$$\mathbb{P}\left(\mathbb{I}\{\tau_{\mathcal{B}^*_{i_{\min}}}(t)=s\}\cdot e^{\lambda\sum_{\tau=1}^t(Z_\tau-\nu_\tau)}\geq e^{\lambda\sqrt{8s\log t}}\right)\leq e^{-\lambda\sqrt{8s\log t}}\cdot\mathbb{E}\left[\mathbb{I}\{\tau_{\mathcal{B}^*_{i_{\min}}}(t)=s\}e^{\lambda\sum_{\tau=1}^t(Z_\tau-\nu_\tau)}\right].\tag{A.7}$$

Let $\mathcal{F}_t=\sigma(Z_1,...,Z_t)$ be a filtration on the observation history, $Y_t=\mathbb{I}\{\pi_t\in\mathcal{B}^*_{i_{\min}}\}$; clearly $Y_t\in\mathcal{F}_{t-1}$. Let $S_t=\sum_{\tau=1}^t Y_\tau$, $G_t=e^{\lambda\sum_{\tau=1}^t(Z_\tau-\nu_\tau)}$ (note that $G_0=1$ and $S_0=0$). We show that $\left\{G_t/e^{\frac{1}{2}\lambda^2S_t}\right\}_t$ is a submartingale. Consider

$$\mathbb{E}\left[\frac{G_t}{e^{\frac{1}{2}\lambda^2S_t}}\middle|\mathcal{F}_{t-1},Y_t=1\right]=\frac{G_{t-1}\mathbb{E}\left[e^{\lambda(Z_t-\nu_t)}\middle|\mathcal{F}_{t-1},Y_t=1\right]}{e^{\frac{1}{2}\lambda^2(S_{t-1}+1)}}\leq\frac{G_{t-1}}{e^{\frac{1}{2}\lambda^2(S_{t-1}+1)}}e^{\frac{1}{2}\lambda^2}=\frac{G_{t-1}}{e^{\frac{1}{2}\lambda^2S_{t-1}}},\tag{A.8}$$

and

$$\mathbb{E}\left[\frac{G_t}{e^{\frac{1}{2}\lambda^2S_t}}\middle|\mathcal{F}_{t-1},Y_t=0\right]=\frac{G_{t-1}\mathbb{E}\left[e^{\lambda(Z_t-\nu_t)}\middle|\mathcal{F}_{t-1},Y_t=0\right]}{e^{\frac{1}{2}\lambda^2S_{t-1}}}=\frac{G_{t-1}}{e^{\frac{1}{2}\lambda^2S_{t-1}}}.\tag{A.9}$$

Note that the inequality in (A.8) holds because given $\mathcal{F}_{t-1}$, π_t is fixed and thus $Z_t = X_{\pi_t}(t)$ which is a sub-Gaussian random variable. Equation (A.9) holds because given $Y_t = 0$, $Z_t = \nu_t = 0$. Therefore, $\left\{G_t/e^{\frac{1}{2}\lambda^2 S_t}\right\}_t$ is a submartingale and

$$\mathbb{E}\left[\frac{G_t}{e^{\frac{1}{2}\lambda^2 S_t}}\right] \leq \mathbb{E}\left[\frac{G_0}{e^{\frac{1}{2}\lambda^2 S_0}}\right] = 1. \tag{A.10}$$

Moreover, we have

$$\mathbb{E}\left[\mathbb{I}\{S_t = s\}\frac{G_t}{e^{\frac{1}{2}\lambda^2 S_t}}\right] \leq \mathbb{E}\left[\frac{G_t}{e^{\frac{1}{2}\lambda^2 S_t}}\right] \leq 1, \tag{A.11}$$

and thus

$$\mathbb{E}\left[\mathbb{I}\{S_t = s\}G_t\right] \leq e^{\frac{1}{2}\lambda^2 s}. \tag{A.12}$$

Applying this to (A.7) and choosing $\lambda = \frac{\sqrt{8s\log t}}{s}$, we have

$$\mathbb{P}\left(\mathbb{I}\{\tau_{\mathcal{B}^*_{i_{\min}}}(t) = s\} \cdot e^{\lambda\sum_{\tau=1}^{t}(Z_\tau - \nu_\tau)} \geq e^{\lambda\sqrt{8s\log t}}\right) \leq e^{\frac{1}{2}\lambda^2 s - \lambda\sqrt{8s\log t}} = e^{-4\log t} = t^{-4}. \tag{A.13}$$

Similarly, the second term can also be upper bounded by t^{-4}. For the third term, let

$$\ell \geq \frac{32\log T}{(\min_{j\in\mathcal{B}^*_{i_{\max}}}\mu_j - \max_{j\in\mathcal{B}^*_{i_{\min}}}\mu_j)^2}. \tag{A.14}$$

Then, since $s \geq \ell$, $t \leq T$, we have

$$\frac{\sum_{j\in\mathcal{B}^*_{i_{\max}}} n_j\mu_j}{r} - \frac{\sum_{j\in\mathcal{B}^*_{i_{\min}}} n_j\mu_j}{s} - 2c_{t,s} \geq \min_{j\in\mathcal{B}^*_{i_{\max}}}\mu_j - \max_{j\in\mathcal{B}^*_{i_{\min}}}\mu_j - \sqrt{\frac{32\log t}{s}} \geq 0. \tag{A.15}$$

Therefore, if we choose $\ell = \lceil\frac{32\log T}{(\min_{j\in\mathcal{B}^*_{i_{\max}}}\mu_j - \max_{j\in\mathcal{B}^*_{i_{\min}}}\mu_j)^2}\rceil$, we get

$$\begin{aligned}
\mathbb{E}[\tau_{\mathcal{B}^*_{i_{\min}}}(T)] &\leq \ell + \sum_{t=|\mathcal{B}^*|}^{T-1}\sum_{s=1}^{t}\sum_{r=1}^{t} 2t^{-4} \\
&\leq \frac{32\log T}{(\min_{j\in\mathcal{B}^*_{i_{\max}}}\mu_j - \max_{j\in\mathcal{B}^*_{i_{\min}}}\mu_j)^2} + O(1) \\
&= \frac{32\log T}{(\min_{j\in\mathcal{B}^*_{i_{\min}}}\Delta_j - \max_{j\in\mathcal{B}^*_{i_{\max}}}\Delta_j)^2} + O(1).
\end{aligned} \tag{A.16}$$

Now we upper bound the number of times that arms in $\mathcal{B}^*_{i_{\max}}$ have been played up to time T. For each $i \in \mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}$,

$$\begin{aligned}
\mathbb{E}[\tau_i(T)] &= \mathbb{E}\left[1 + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{I}\{\pi_t = i\}\right] \\
&\leq \ell + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{P}\left(\pi_t = i, \tau_i(t-1) \geq \ell\right) \qquad \text{(A.17)} \\
&\leq \ell + \sum_{t=|\mathcal{B}^*|+1}^{T} \mathbb{P}\left(L_i(t-1) \geq L_{i_{\max}}(t-1), \tau_i(t-1) \geq \ell\right).
\end{aligned}$$

Using an argument similar to that for $\tau_{\mathcal{B}^*_{i_{\min}}}$, we get

$$\mathbb{E}[\tau_i(T)] \leq \frac{32 \log T}{\Delta_i^2} + O(1). \qquad \text{(A.18)}$$

Therefore, we get the upper bound on regret of LSDT-CSI in (2.9) if $\mathcal{G}^*_\epsilon$ is connected but not complete.

A.3 Proof of Theorem 3

The basic structure of the proof follows that in [46] and [15]. For every suboptimal arm i ($\mu_i < \mu_{i_{\max}}$), we construct a new set of reward distributions with parameters $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, \theta_2^{(i)}, ..., \theta_K^{(i)})$ and means $\boldsymbol{\mu}^{(i)} = (\mu_1^{(i)}, \mu_2^{(i)}, ..., \mu_K^{(i)})$ such that $\mu_i^{(i)} = \max_{j \in \mathcal{V}} \mu_j^{(i)}$. Then we can generate a new graph $\mathcal{G}_\epsilon^{(i)} = (\mathcal{V}^{(i)}, \mathcal{E}^{(i)})$ where $\mathcal{V}^{(i)}$ is the set of new arms, and $(u, v) \in \mathcal{E}^{(i)}$ if and only if $|\mu_u^{(i)} - \mu_v^{(i)}| < \epsilon$.

To establish the relationship between the new problem and the original one, we need to retain the same graph connectivity. Since $\mathcal{B}^*$ is the set of arms that could potentially be optimal given $\mathcal{G}^*_\epsilon$, we could only construct for each $i \in \mathcal{B}^* \setminus \mathcal{A}$ a set of new reward distributions with parameters $\theta^{(i)}$ such that arm i is optimal. Thus, for each $i \in \mathcal{B}^* \setminus \mathcal{A}$, consider $\theta^{(i)}$ with mean rewards $\mu^{(i)}$ satisfying:

1. If $i \in \mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}$: $\mu_i^{(i)} = \mu_{i_{\max}} + \eta$, $\mu_j^{(i)} = \mu_j, \forall j \neq i$.

2. If $i \in \mathcal{B}^*_{i_{\min}}$: $\mu_i^{(i)} = \mu(\theta'_i) + \eta$, $\mu_j^{(i)} = \mu(\theta'_j), \forall j \neq i$, where $\mu(\theta'_i), \mu(\theta'_j)$ are defined as

$$\mu(\theta'_j) = \begin{cases} \mu_j, & \forall j \in \mathcal{B}^*_{i_{\max}}, \\ \mu_{i_{\max}} + \min_{k \in \mathcal{B}^*_{i_{\max}}} \mu_k - \mu_{i_{\min}}, & \forall j \in \mathcal{B}^*_{i_{\min}}, \\ \mu_{i_{\max}} + \min_{k \in \mathcal{B}^*_{i_{\max}}} \mu_k - \mu_j, & \forall j \notin \mathcal{B}^*. \end{cases} \tag{A.19}$$

One can check that in both cases, $\mathcal{G}_\epsilon^{(i)}$ and $\mathcal{G}_\epsilon^*$ have the same connectivity if

$$\eta < \epsilon - \max\{\mu_{i_{\max}} - \min_{j \in \mathcal{N}[i_{\max}]} \mu_j, \max_{j \in \mathcal{N}[i_{\min}]} \mu_j - \mu_{i_{\min}}\}. \tag{A.20}$$

Then we define the log-likelihood ratio between the observations from two sets of arms with distribution parameters $\theta = (\theta_1, ..., \theta_K)$ and $\theta^{(i)} = (\theta_1^{(i)}, ..., \theta_K^{(i)})$ up to time T under any uniformly good policy π as

$$\mathcal{L}^{(i)}(T) = \sum_{j \in \mathcal{V}} \sum_{s=1}^{\tau_j(T)} \log\left(\frac{f(X_{j,s}; \theta_j)}{f(X_{j,s}; \theta_j^{(i)})}\right), \tag{A.21}$$

where $\tau_j(T)$ is the number of times arm j has been played by policy π up to time T and $X_{j,s}$ is the reward obtained when arm j is played for the s-th time. We show that it is unlikely to have

$$\sum_{j \in \mathcal{V}} \tau_j(T) I(\theta_j \| \theta_j^{(i)}) \leq (1 - \gamma) \log T, \tag{A.22}$$

under two separate cases: $\mathcal{L}^{(i)}(T) \leq (1 - \delta) \log T$ and $\mathcal{L}^{(i)}(T) > (1 - \delta) \log T$ where $\delta, \gamma > 0$ are determined later.

1. If $\mathcal{L}^{(i)}(T) \leq (1 - \delta) \log T$: by the uniform goodness of policy π, we have

$$\begin{aligned} & \mathbb{P}_{\boldsymbol{\theta}^{(i)}} \left\{ \sum_{j \in \mathcal{V}} \tau_j(T) I(\theta_j \| \theta_j^{(i)}) \leq (1 - \gamma) \log T \right\} \\ \leq \quad & \mathbb{P}_{\boldsymbol{\theta}^{(i)}} \left\{ \tau_i(T) I(\theta_i \| \theta_i^{(i)}) \leq (1 - \gamma) \log T \right\} \end{aligned}$$

$$
\begin{aligned}
&= \mathbb{P}_{\boldsymbol{\theta}^{(i)}}\left\{T - \tau_i(T) \geq T - \frac{(1-\gamma)\log T}{I(\theta_i \| \theta_i^{(i)})}\right\} \qquad \text{(A.23)}\\
&\leq \frac{\mathbb{E}_{\boldsymbol{\theta}^{(i)}}[T - \tau_i(T)]}{T - \frac{(1-\gamma)\log T}{I(\theta_i \| \theta_i^{(i)})}} = o(T^{\alpha-1}),
\end{aligned}
$$

for all $\alpha > 0$ as $T \to \infty$.

We let

$$
H = \left\{ \sum_{j \in \mathcal{V}} \tau_j(T) I(\theta_j \| \theta_j^{(i)}) \leq (1-\gamma)\log T, \mathcal{L}^{(i)}(T) \leq (1-\delta)\log T \right\}. \qquad \text{(A.24)}
$$

By a change of measure from $\mathbb{P}_{\boldsymbol{\theta}^{(i)}}$ to $\mathbb{P}_{\boldsymbol{\theta}}$, we have

$$
\mathbb{P}_{\boldsymbol{\theta}}\{H\} \leq \int_H \mathrm{d}P_{\boldsymbol{\theta}} = \int_H \exp\left(\mathcal{L}^{(i)}(T)\right) \mathrm{d}P_{\boldsymbol{\theta}^{(i)}} \leq T^{1-\delta} o(T^{\alpha-1}) = o(1), \qquad \text{(A.25)}
$$

for all $\delta > 0$ as $T \to \infty$ if we choose $\alpha < \delta$.

2. If $\mathcal{L}^{(i)}(T) > (1-\delta)\log T$: by the strong law of large numbers, as $t \to \infty$, we have

$$
\frac{1}{t}\sum_{s=1}^{t} \log\left(\frac{f(X_{j,s};\theta_j)}{f(X_{j,s};\theta_j^{(i)})}\right) \to I(\theta_j \| \theta_j^{(i)}) \text{ almost surely.} \qquad \text{(A.26)}
$$

Rewrite $\mathcal{L}^{(i)}(T)$ as

$$
\mathcal{L}^{(i)}(T) = \sum_{j \in \mathcal{V}} \tau_j(T) \frac{1}{\tau_j(T)} \sum_{s=1}^{\tau_j(T)} \log\left(\frac{f(X_{j,s};\theta_j)}{f(X_{j,s};\theta_j^{(i)})}\right) \qquad \text{(A.27)}
$$

and then

$$
\begin{aligned}
&\mathbb{P}_{\boldsymbol{\theta}}\left\{ \sum_{j \in \mathcal{V}} \tau_j(T) I(\theta_j \| \theta_j^{(i)}) \leq (1-\gamma)\log T, L^{(i)}(T) > (1-\delta)\log T \right\}\\
= \;& \mathbb{P}_{\boldsymbol{\theta}}\left\{ \sum_{j \in \mathcal{K}} \tau_j(T) I(\theta_j \| \theta_j^{(i)}) \leq (1-\gamma)\log T, \right.\\
&\left. \sum_{j \in \mathcal{K}} \tau_j(T) \frac{1}{\tau_j(T)} \sum_{s=1}^{\tau_j(T)} \log\left(\frac{f(X_{j,s};\theta_j)}{f(X_{j,s};\theta_j^{(i)})}\right) > (1-\delta)\log T \right\} \qquad \text{(A.28)}\\
= \;& o(1),
\end{aligned}
$$

as $T \to \infty$ if we choose $\gamma > \delta$.

Now we have proved that for all $i \in \mathcal{B}^* \setminus \mathcal{A}$, we have

$$\sum_{j \in \mathcal{V}} \frac{\mathbb{E}[\tau_j(T)]}{\log T} I(\theta_j \| \theta_j^{(i)}) \geq 1. \tag{A.29}$$

To be specific,

1. If $i \in \mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}$, let $\eta \to 0$, we have

$$\mathbb{E}[\tau_i(T)] \geq \frac{\log T}{I(\theta_i \| \theta_{i_{\max}})}, \tag{A.30}$$

2. If $i \in \mathcal{B}^*_{i_{\min}}$, let $\eta \to 0$, we have

$$\sum_{j \notin \mathcal{B}^*_{i_{\max}}} \mathbb{E}[\tau_j(T)] I(\theta_j \| \theta'_j) \geq \log T. \tag{A.31}$$

Therefore, the optimal constant in front of $\log T$ is the solution to the linear program $\mathcal{P}_1$:

$$\begin{aligned} \mathcal{P}_1 : C_1 \quad = \quad & \min_{\{\tau_i\}_{i \in \mathcal{V}}} \sum_{i \in \mathcal{V}} \Delta_i \tau_i, \\ s.t. \quad & \sum_{j \notin \mathcal{B}^*_{i_{\max}}} \tau_j I(\theta_j \| \theta'_j) \geq 1, \\ & \tau_i \geq \frac{1}{I(\theta_i \| \theta_{i_{\max}})}, \quad \forall i \in \mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}, \\ & \tau_i \geq 0, \qquad\qquad \forall i \in \mathcal{V}. \end{aligned} \tag{A.32}$$

where θ'_j is the parameter of the density function $f(x_j; \theta'_j)$ whose mean value $\mu(\theta'_j)$ is defined in (A.19).

In light of the LP $\mathcal{P}_1$, each sub-optimal arm in $\mathcal{B}^*_{i_{\max}}$ has to be played $\Omega(\log T)$ times to be distinguished from the optimal one. Moreover, the total number of times that arms in $\mathcal{V} \setminus \mathcal{B}^*_{i_{\max}}$ are played should be at least $\Omega(\log T)$. Thus if we consider the regret order in terms of the number of arms and the time length, we

can conclude that for fixed Δ_i, $I(\theta_i||\theta_i')$ and $I(\theta_i||\theta_{i_{\max}})$, the regret for any uniformly good policy is of order

$$\Omega\Big((1 + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) \log T\Big),$$

as $T \to \infty$, which matches the upper bound on regret of LSDT-CSI. Therefore, LSDT-CSI is order optimal.

A.4 Proof of NP-Completeness of CONSISTENT-NAE-3SAT

The problem is clearly in NP since a given truth assignment can be verified in polynomial time. To show the NP-completeness, we first show that 1-CONSISTENT-NAE-3SAT is NP-complete. Note that 1-CONSISTENT-NAE-3SAT asks if there exists a truth assignment such that every clause has exactly 1 true literal given true instance of NAE-3SAT.

It is clear that 1-CONSISTENT-NAE-3SAT is in NP. We give a reduction from 1-IN-3SAT, a known NP-complete problem [58], as follows: given an instance of 1-IN-3SAT, for every clause $C_i = (x_{i,1}, x_{i,2}, x_{i,3})$, we construct three clauses in the corresponding 1-CONSISTENT-NAE-3SAT instance with two additional variables a_i, b_i:

$$C_{i,1} = (x_{i,1}, x_{i,2}, a_i),\ C_{i,2} = (x_{i,2}, x_{i,3}, b_i),\ C_{i,3} = (a_i, b_i, x_{i,2}).$$

This is clearly a polynomial time reduction.

We first show that the 1-CONSISTENT-NAE-3SAT instance we constructed is not-all-equal (NAE) satisfiable, i.e., there exists a truth assignment such that every clause is satisfied and contains at most 2 true literals. For any arbitrary truth assignment of $(x_1, ..., x_n)$, we can choose $(a_1, b_1, ..., a_m, b_m)$ according to Table

A.1. One can check that every clause is satisfied with at most 2 true literals. Therefore, the 1-CONSISTENT-NAE-3SAT instance is NAE satisfiable.

$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	a_i	b_i
0	0	0	1	1
0	0	1	1	0
0	1	0	0	0
1	0	0	0	1
0	1	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Table A.1: Truth table for NAE-3SAT.

Now we assume that the original 1-IN-3SAT instance is satisfied by an assignment of $(x_1, ..., x_n)$ with three cases:

1. only $x_{i,1}$ is true: let $a_i = 0, b_i = 1$;
2. only $x_{i,2}$ is true: let $a_i = 0, b_i = 0$;
3. only $x_{i,3}$ is true: let $a_i = 1, b_i = 0$.

It is clear that the 1-CONSISTENT-NAE-3SAT is satisfied by the assignment of $(x_1, ..., x_n, a_1, b_1, ..., a_m, b_m)$.

On the other hand, assume that the 1-CONSISTENT-NAE-3SAT instance is satisfied by an assignment of $(x_1, ..., x_n, a_1, b_1, ..., a_m, b_m)$. Consider clause $C_{i,1} = (x_{i,1}, x_{i,2}, a_i)$:

1. only $x_{i,1}$ is true: $x_{i,2} = a_i = 0$. It is clear that $b_i = 1$ since $C_{i,3}$ is satisfied. Thus, $x_{i,3} = 0$ since $C_{i,2}$ is satisfied with only one true literal (b_i). Therefore, we have $x_{i,1} = 1, x_{i,2} = 0, x_{i,3} = 0$;

2. only $x_{i,2}$ is true: since $C_{i,1}, C_{i,2}, C_{i,3}$ are all satisfied with only one true literal in each clause, we have $x_{i,1} = x_{i,3} = a_i = b_i = 0$;
3. only a_i is true: $x_{i,1} = x_{i,2} = 0$. since $C_{i,2}, C_{i,3}$ are satisfied with only one true literal in each clause, we have $b_i = x_{i,2} = 0$ and $x_{i,3} = 1$.

Therefore, every clause $C_i = (x_{i,1}, x_{i,2}, x_{i,3})$ in the original 1-IN-3SAT instance is satisfied with only one true literal.

In summary, we have shown that the 1-IN-3SAT instance is satisfiable if and only if the corresponding 1-CONSISTENT-NAE-3SAT instance is satisfiable, which indicates the NP-completeness of 1-CONSISTENT-NAE-3SAT.

Finally, we show that CONSISTENT-NAE-3SAT (clearly in NP) is NP-complete via a reduction from 1-CONSISTENT-NAE-3SAT. Given an instance of 1-CONSISTENT-NAE-3SAT with n variables $(x_1, ..., x_n)$ and m clauses $C_1, ..., C_m$, we add a new clause $C_0 = (x_1, \bar{x}_1, 0)$ and get an instance of CONSISTENT-NAE-3SAT with n variables and $m + 1$ clauses. This is clearly a polynomial reduction and there must exist a NAE satisfiable assignment for the new instance. Now we assume that the original 1-CONSISTENT-NAE-3SAT instance has a satisfiable assignment $(x_1, ..., x_n)$, it follows immediately that the CONSISTENT-NAE-3SAT is also satisfied by the same assignment. On the other hand, we assume that CONSISTENT-NAE-3SAT is satisfied by a truth assignment $(x_1, ..., x_n)$. Since C_0 is satisfied with exactly 1 true literal, so are the other clauses. Thus $(x_1, ..., x_n)$ is a satisfiable assignment for the 1-CONSISTENT-NAE-3SAT instance. Hence, we have shown that the 1-CONSISTENT-NAE-3SAT instance is satisfiable if and only if the corresponding CONSISTENT-NAE-3SAT instance is satisfiable.

In conclusion, CONSISTENT-NAE-3SAT is NP-complete.

A.5 Proof of Theorem 4

It is clear that LEFTANCHOR is in NP since given a graph, one can verify if it is a UIG and if a specific node is a left anchor in polynomial time. Now we show the NP-completeness of LEFTANCHOR through a reduction from CONSISTENT-NAE-3SAT. The reduction is similar to the one used in proving the NP-completeness of the UIG Sandwich Problem in [37].

Given an instance of CONSISTENT-NAE-3SAT, let $x_1, ..., x_n$ be n variables and $C_1, ..., C_m$ be m clauses where $C_i = (x_{i,1}, x_{i,2}, x_{i,3})$ and $x_{i,j} \in \{x_1, ..., x_n, \bar{x}_1, ..., \bar{x}_n\}$. For every variable x_i, we construct a variable gadget with 5 vertices $(x_i, x_i', p, \bar{x}_i', \bar{x}_i)$ in the LEFTANCHOR instance: add 4 type-S edges $(x_i, x_i'), (x_i', p), (p, \bar{x}_i'), (\bar{x}_i', \bar{x}_i)$ to $\mathcal{E}_1$ and 6 type-D edges (x_i, p), $(x_i, \bar{x}_i')$, $(x_i, \bar{x}_i)$, $(x_i', \bar{x}_i')$, $(x_i', \bar{x}_i)$, $(p, \bar{x}_i)$ to $\mathcal{E}_2$ (see Figure A.1: solid line edges represent type-S edges in $\mathcal{E}_1$, missing edges represent type-D edges in $\mathcal{E}_2$).

Moreover, for every clause $C_i = (x_{i,1}, x_{i,2}, x_{i,3})$, we construct a clause gadget with 6 vertices $(x_{i,1}, x_{i,2}, x_{i,3}, v_{i,1}, v_{i,2}, v_{i,3})$ in the LEFTANCHOR instance: add 3 type-S edges $(x_{i,j}, v_{i,j}), j = 1, 2, 3$ to $\mathcal{E}_1$ and 9 type-D edges $(v_{i,j}, x_{i,k}), j \neq k$ and $(v_{i,j}, v_{i,k}), j \neq k$ to $\mathcal{E}_2$ (see Figure A.2: solid line edges represent type-S edges in $\mathcal{E}_1$, missing edges represent type-D edges in $\mathcal{E}_2$, and dash line edges represent unknown edges in $\overline{\mathcal{E}_1 \cup \mathcal{E}_2}$). Note that every vertex $x_{i,j}$ in the clause gadget belongs to one of the variable gadgets, we don't create additional vertices.

In summary, there are $4n + 3m + 1$ vertices in the LEFTANCHOR instance:

$$\mathcal{V} = \{p\} \cup \{x_i, x_i', \bar{x}_i', \bar{x}_i | i = 1, ..., n\} \cup \{v_{i,1}, v_{i,2}, v_{i,3} | i = 1, ..., m\},$$

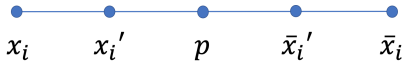

Figure A.1: Variable gadget

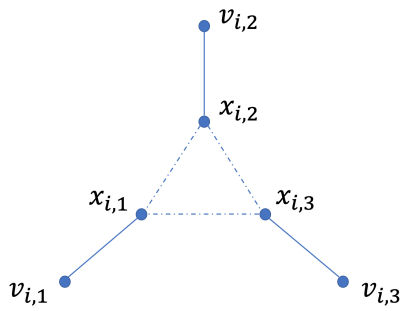

Figure A.2: Clause gadget

$4n + 3m$ type-S edges:

$$\mathcal{E}_1 = \left\{(x_i, x_i'), (x_i', p), (p, \bar{x}_i'), (\bar{x}_i', \bar{x}_i) | i = 1, ..., n\right\} \cup \left\{(x_{i,j}, v_{i,j}) | i = 1, ..., m, j = 1, 2, 3\right\},$$

and $6n + 9m$ type-D edges:

$$\begin{aligned} \mathcal{E}_2 = \quad & \left\{(x_i, p), (x_i, \bar{x}_i'), (x_i, \bar{x}_i), (x_i', \bar{x}_i'), (x_i', \bar{x}_i), (p, \bar{x}_i) \middle| i = 1, ..., n\right\} \\ & \cup \left\{(v_{i,j}, x_{i,k}) \middle| i = 1, ..., m, j \neq k\right\} \cup \left\{(v_{i,j}, v_{i,k}) \middle| i = 1, ..., m, j \neq k\right\}. \end{aligned} \quad \text{(A.33)}$$

Clearly the construction is done in polynomial time. Moreover, it is shown in [37] that if there exists a truth assignment of $(x_1, ..., x_n)$ such that every clause C_i is satisfied with at most two true literals, there exists a UIG $\mathcal{G}' = (\mathcal{V}, \mathcal{E}_3)$ such that $\mathcal{E}_1 \subseteq \mathcal{E}_3$ and $\mathcal{E}_3 \cap \mathcal{E}_2 = \emptyset$. Now, let x_1 and $\bar{x}_1$ be two nodes that we want to

decide if they can be left anchors. Then we get two corresponding instances of LEFTANCHOR for any given instance of CONSISTENT-NAE-3SAT. We need to show that the instance of CONSISTENT-NAE-3SAT is satisfiable if and only if at least one of the two corresponding instances of LEFTANCHOR is satisfiable, i.e., at least one of the two nodes x_1 and $\bar{x}_1$ can be a left anchor of a UIG $\mathcal{G}'' = (\mathcal{V}, \mathcal{E}_4)$ where $\mathcal{E}_1 \subseteq \mathcal{E}_4$ and $\mathcal{E}_4 \cap \mathcal{E}_2 = \emptyset$.

We first assume that the CONSISTENT-NAE-3SAT instance is satisfied by a truth assignment of $(x_1, ..., x_n)$. Suppose every clause has only 1 true literal and with out loss of generality, we assume $x_1 = 1$. We show that x_1 can be a left anchor of a UIG satisfying the constraints. We assign a unit length interval for every vertex in $\mathcal{V}$ as follows (see Figure A.3): we let $I(p) = P$. For $i = 1, ..., n$, if $x_i = 1$, we let $I(x_i) = A_i$, $I(x_i') = L$, $I(\bar{x}_i') = R$ and $I(\bar{x}_i) = B_i$; if $x_i = 0$, we let $I(x_i) = B_i$, $I(x_i') = R$, $I(\bar{x}_i') = L$ and $I(\bar{x}_i) = A_i$. In other words, we put all the true (or false) literals to the left (or right) "staircases" and assign x_i' and $\bar{x}_i'$ accordingly. For every clause $C_i, i = 1, ..., m$, let $x_{i,j}$ be the true literal in C_i, then we let $I(v_{i,j}) = I(x_{i,j})$. For the other two false literals x_{i,k_1}, x_{i,k_2}, the two corresponding intervals both have non-overlapping tails. Therefore, we can assign $I(v_{i,k_1})$ and $I(v_{i,k_2})$ extending from the respective tails. For example in Figure A.3: consider a clause $(x_1, \bar{x}_2, \bar{x}_3)$, A_1, B_2, B_3 are assigned to the three literals and V_1, V_2, V_3 are assigned to the associated vertices $v_{i,1}, v_{i,2}, v_{i,3}$. One can easily check that the induced UIG from the interval assignment of vertices in $\mathcal{V}$ satisfies all the edge constraints and x_1 is a left anchor. Similarly, we can show that if $x_1 = 0$, then $\bar{x}_1$ can be a left anchor of a UIG in the LEFTANCHOR instance. Now suppose every clause has 2 true literals, we can use similar proof structure to show that if $x_1 = 0$, then x_1 can be a left anchor of a UIG satisfying the edges constraints; otherwise $\bar{x}_1$ can be a left anchor.

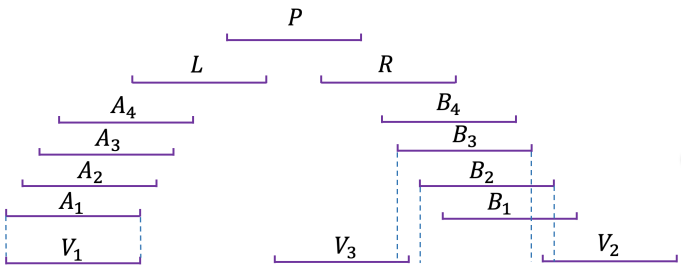

Figure A.3: Unit interval realization of a satisfiable instance of CONSISTENT-NAE-3SAT.

On the other hand, we assume that at least one of the two nodes x_1 and $\bar{x}_1$ can be a left anchor of a UIG $\mathcal{G}'' = (\mathcal{V}, \mathcal{E}_4)$ satisfying the edge constraints and we need to show that the original instance of CONSISTENT-NAE-3SAT is satisfiable. Without loss of generality, we assume that x_1 can be a left anchor. Let $I(v)$ be the unit interval assigned to vertex v in the UIM of the UIG $\mathcal{G}''$. By changing scale and shifting, we can assume that every interval has length 1 and $I(p) = [0, 1]$. Consider for every variable gadget $i = 1, ..., n$, It is not difficult to see that $I(x_i)$ contains either -1 or 2. We assign truth values to the variables as follows: let $x_i = 1$ if $I(x_i)$ contains -1 and $x_i = 0$ if $I(x_i)$ contains 2. Now we show that every clause is satisfied with only 1 true literal OR every clause is satisfied with only 2 true literals by the truth assignment.

Consider every clause gadget: we show that there is exactly one edge among $(x_{i,j}, x_{i,k})$, $j \neq k$ that belongs to $\mathcal{E}_4$:

1. if all three edges belong to $\mathcal{E}_4$, then $v_{i,1}, v_{i,2}, v_{i,3}$ form an asteroidal triple[1], which is forbidden in a UIG [48];

[1]An asteroidal triple in a graph is a triple of mutually non-adjacent nodes i, j, k such that between any two of them, there exists a path avoiding the neighborhood of the third.

2. if exactly two edges belong to $\mathcal{E}_4$, e.g., $(x_{i,1}, x_{i,2})$ and $(x_{i,1}, x_{i,3})$, then $x_{i,1}, x_{i,2}, x_{i,3}$ and $v_{i,1}$ form a claw ($K_{1,3}$) which is also forbidden in a UIG [48];

3. $I(x_{i,j})$ contains either -1 or 2. Hence, there always exist two intervals containing the same point, thus intersecting. Therefore, at least one edge belongs to $\mathcal{E}_4$.

Since there is exactly one edge among $(x_{i,j}, x_{i,k})$, $j \neq k$ that belongs to $\mathcal{E}_4$, it follows that every clause has only 1 or 2 true literals. Furthermore, since x_1 is a left anchor of $\mathcal{G}''$, within every clause gadget containing x_1, the truth assignment of x_1 should be different from the other two variables: consider a clause gadget containing x_1, assume that $x_{i,k}$ has the same truth assignment as x_1 and x_1 is connected to v_1, then we have $(x_1, x_{i,k}), (x_1, v_1) \in \mathcal{E}_4$, but $(v_1, x_{i,k}) \notin \mathcal{E}_4$. This contradicts the assumption that x_1 is a left anchor since $(v_1, x_{i,k})$ should also belong to $\mathcal{E}_4$ if x_1 is a left anchor and $(x_1, x_{i,k}), (x_1, v_1) \in \mathcal{E}_4$.

We first consider if $x_1 = 1$, then we claim that every clause has exactly one true literal. We prove by contradiction: assume that there exists a clause $C_i = (x_{i,1}, x_{i,2}, x_{i,3})$ with two true literals, e.g., $x_{i,1}, x_{i,2}$. By the assignment of truth values, it is clear that $I(x_{i,1}) = [l_1, r_1]$ and $I(x_{i,2}) = [l_2, r_2]$ contains -1. Without loss of generality, we assume that $l_1 < l_2$. Then we consider $I(v_{i,1}) = [l_v, r_v]$: since $(v_{i,1}, x_{i,1}) \in \mathcal{E}_4$ and $(v_{i,1}, x_{i,2}) \notin \mathcal{E}_4$, we have $l_1 \leq r_v < l_2$. Then it is not difficult to see that $I(v_{i,1})$ doesn't contain -1 and l_v is smaller than the left coordinate of $I(x_1)$, i.e., x_1 is not a left anchor. Contradiction! Therefore, we have shown that every clause has exactly one true literal. On the other hand, if $x_1 = 0$, it can be shown similarly that every clause has exactly two true literals. In summary, we have shown that the original instance of CONSISTENT-NAE-3SAT is satisfiable if at least one of the two nodes x_1 and $\bar{x}_1$ can be a left anchor of the corresponding

UIG. This completes the entire reduction and we conclude that LEFTANCHOR is NP-complete.

A.6 Proof of Theorem 5

We first show that with probability at least $1 - \frac{1}{K^2}$, every arm $i \notin \mathcal{B}^*$ is eliminated by the offline elimination step of LSDT-PSI. Consider any $i \notin \mathcal{B}^*$. Note that under Assumption 3, there exists $j, k \in [m]$ s.t. $\forall u \in \mathcal{B}_j^*, v \in \mathcal{B}_k^*$,

$$(u, i) \in \mathcal{E}_\epsilon^*,\ (v, i) \in \mathcal{E}_\epsilon^*,\ (u, v) \in \overline{\mathcal{E}_\epsilon^*}. \tag{A.34}$$

Let $N = \min\{|\mathcal{B}_j^*|, |\mathcal{B}_k^*|\}$. According to Assumption 4, we have $N \geq \kappa \log K$. We select $\{u_1, u_2, ..., u_N\}$ from $\mathcal{B}_j^*$ and $\{v_1, v_2, ..., v_N\}$ from $\mathcal{B}_k^*$, then for $n = 1, ..., N$, define

$$E_{n,1} = \left\{(u_n, i) \in \mathcal{E}_\epsilon^S\right\}, \tag{A.35}$$

$$E_{n,2} = \left\{(v_n, i) \in \mathcal{E}_\epsilon^S\right\}, \tag{A.36}$$

$$E_{n,3} = \left\{(u_n, v_n) \in \mathcal{E}_\epsilon^D\right\}. \tag{A.37}$$

According to Assumption 5, $\{E_{n,\ell}\}_{n=1,...,N,\ell=1,2,3}$ are independent and $\mathbb{P}(E_{n,1}) = \mathbb{P}(E_{n,2}) = p_S$, $\mathbb{P}(E_{n,3}) = p_D$. Therefore, according to the offline elimination step of LSDT-PSI, the probability that arm i is not eliminated is upper bounded as follows:

$$\mathbb{P}(i \text{ is not eliminated from } \mathcal{B}_0) \leq \prod_{n=1}^{N}\left(1 - \prod_{\ell=1}^{3}\mathbb{P}(E_{n,\ell})\right) = (1 - p_S^2 p_D)^N. \tag{A.38}$$

Since $N \geq \kappa \log K$ and according to Assumption 5, $p_S^2 p_D \geq 1 - e^{-2/\kappa}$, we have

$$(1 - p_S^2 p_D)^N \leq (e^{-2/\kappa})^{\kappa \log K} \leq \frac{1}{K^2}. \tag{A.39}$$

Moreover, we can show that as $K \to \infty$:

$$\begin{aligned}\mathbb{E}_{\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D}\left[|\mathcal{B}_0|\right] &= \sum_{i=1}^{K} \mathbb{P}(i \text{ is not eliminated from } \mathcal{B}_0) \\ &= |\mathcal{B}^*| + \sum_{i \notin \mathcal{B}^*} \frac{1}{K^2} \leq |\mathcal{B}^*| + o(1). \end{aligned} \tag{A.40}$$

A.7 Proof of Theorem 6

The basic structure of the proof follows that in [12] and [15]. Define $\mathcal{Q} = \{i \in \mathcal{V}' : \Delta_i > 4\epsilon\}$ (note that $\mathcal{V}' = \mathcal{B}_0$). For each $i \in \mathcal{Q}$, let

$$m_i = \min\left\{m \geq 0 : 2^{-m} < \frac{\sqrt{2\lambda}(\Delta_i - 3\epsilon)}{4}\right\}. \tag{A.41}$$

One can easily verify that

$$\min\left\{\frac{1}{2}, \frac{\sqrt{2\lambda}(\Delta_i - 3\epsilon)}{8}\right\} \leq 2^{-m_i} < \frac{\sqrt{2\lambda}(\Delta_i - 3\epsilon)}{4}, \tag{A.42}$$

and

$$\max_{i \in \mathcal{Q}} m_i \leq \max\left\{1, \left\lceil \log_2\left(\frac{8}{\sqrt{2\lambda}\epsilon}\right)\right\rceil\right\}. \tag{A.43}$$

We first consider suboptimal arms in $\mathcal{Q}$ and analyze regret in the following cases:

(a) Some suboptimal arm $i \in \mathcal{Q}$ is not eliminated in round m_i (or before) with an optimal arm $i_{\max} \in \mathcal{B}_{m_i}$.

Consider $i \in \mathcal{Q}$, note that if

$$\frac{\sum_{j \in \mathcal{N}'[i]} \bar{x}_j(m)\tau_j(m)}{\sum_{j \in \mathcal{N}'[i]} \tau_j(m)} \leq \frac{\sum_{j \in \mathcal{N}'[i]} \mu_j \tau_j(m)}{\sum_{j \in \mathcal{N}'[i]} \tau_j(m)} + \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2\sum_{j \in \mathcal{N}'[i]} \tau_j(m)}}, \tag{A.44}$$

and

$$\frac{\sum_{j \in \mathcal{N}'[i_{\max}]} \bar{x}_j(m)\tau_j(m)}{\sum_{j \in \mathcal{N}'[i_{\max}]} \tau_j(m)} \geq \frac{\sum_{j \in \mathcal{N}'[i_{\max}]} \mu_j \tau_j(m)}{\sum_{j \in \mathcal{N}'[i_{\max}]} \tau_j(m)} - \sqrt{\frac{\log(T\tilde{\Delta}_m^2)}{2\sum_{j \in \mathcal{N}'[i_{\max}]} \tau_j(m)}}, \tag{A.45}$$

hold for $m = m_i$, then under the assumption that $i_{\max}, i \in \mathcal{B}_{m_i}$, we have

$$\sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\sum_{j\in\mathcal{N}'[i]}\tau_j(m_i)}} \leq \sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\lambda\sum_{j\in\mathcal{N}'[i]}z_j\log(T\tilde{\Delta}_{m_i}^2)/\tilde{\Delta}_{m_i}^2}} \leq \frac{\tilde{\Delta}_{m_i}}{\sqrt{2\lambda}} < \frac{\Delta_i - 3\epsilon}{4}, \tag{A.46}$$

$$\sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\sum_{j\in\mathcal{N}'[i_{\max}]}\tau_j(m_i)}} \leq \sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\lambda\sum_{j\in\mathcal{N}'[i_{\max}]}z_j\log(T\tilde{\Delta}_{m_i}^2)/\tilde{\Delta}_{m_i}^2}} \leq \frac{\tilde{\Delta}_{m_i}}{\sqrt{2\lambda}} < \frac{\Delta_i - 3\epsilon}{4}. \tag{A.47}$$

Thus,

$$\begin{aligned}
& \frac{\sum_{j\in\mathcal{N}'[i]}\bar{x}_j(m_i)\tau_j(m_i)}{\sum_{j\in\mathcal{N}'[i]}\tau_j(m_i)} + \sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\sum_{j\in\mathcal{N}'[i]}\tau_j(m_i)}} + \epsilon \\
\leq\ & \frac{\sum_{j\in\mathcal{N}'[i]}\mu_j\tau_j(m_i)}{\sum_{j\in\mathcal{N}'[i]}\tau_j(m_i)} + \frac{\Delta_i - 3\epsilon}{2} + \epsilon \\
\leq\ & \mu_i + 2\epsilon + \frac{\Delta_i - 3\epsilon}{2} \\
=\ & \mu_{i_{\max}} - \epsilon - \frac{\Delta_i - 3\epsilon}{2} \\
\leq\ & \frac{\sum_{j\in\mathcal{N}'[i_{\max}]}\mu_j\tau_j(m_i)}{\sum_{j\in\mathcal{N}'[i_{\max}]}\tau_j(m_i)} - 2\sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\sum_{j\in\mathcal{N}'[i_{\max}]}\tau_j(m_i)}} \\
\leq\ & \frac{\sum_{j\in\mathcal{N}'[i_{\max}]}\bar{x}_j(m_i)\tau_j(m_i)}{\sum_{j\in\mathcal{N}'[i_{\max}]}\tau_j(m_i)} - \sqrt{\frac{\log(T\tilde{\Delta}_{m_i}^2)}{2\sum_{j\in\mathcal{N}'[i_{\max}]}\tau_j(m_i)}}.
\end{aligned} \tag{A.48}$$

Therefore, arm i will be eliminated in round m_i. Using Hoeffding's inequality, we know that for every $m = 0, 1, 2, ...,$

$$\mathbb{P}\{(A.44) \text{ doesn't hold}\} \leq \frac{1}{T\tilde{\Delta}_m^2}, \tag{A.49}$$

$$\mathbb{P}\{(A.45) \text{ doesn't hold}\} \leq \frac{1}{T\tilde{\Delta}_m^2}. \tag{A.50}$$

As a consequence, the probability that a suboptimal arm i is not eliminated in round m_i (or before) by an optimal arm is bounded by $2/(T\tilde{\Delta}_{m_i}^2)$ and thus, the regret contributed by case (a) is upper bounded by

$$R_a(T) \leq \sum_{i\in Q}\frac{2\Delta_i}{\tilde{\Delta}_{m_i}^2} = O(|\mathcal{V}'|). \tag{A.51}$$

(b) The last remaining optimal arm $i_{\max}$ is eliminated by some suboptimal arm i in some round $m^* < m_f$.

Note that if (A.44) and (A.45) hold at $m = m^*$, then

$$\begin{aligned} & \frac{\sum_{j\in\mathcal{N}'[i_{\max}]} \bar{x}_j(m^*)\tau_j(m^*)}{\sum_{j\in\mathcal{N}'[i_{\max}]} \tau_j(m^*)} + \sqrt{\frac{\log(T\tilde{\Delta}^2_{m^*})}{2\sum_{j\in\mathcal{N}'[i_{\max}]} \tau_j(m^*)}} + \epsilon \\ \geq & \frac{\sum_{j\in\mathcal{N}'[i_{\max}]} \mu_j\tau_j(m^*)}{\sum_{j\in\mathcal{N}'[i_{\max}]} \tau_j(m^*)} + \epsilon \\ \geq & \frac{\sum_{j\in\mathcal{N}'[i]} \mu_j\tau_j(m^*)}{\sum_{j\in\mathcal{N}'[i]} \tau_j(m^*)} \\ \geq & \frac{\sum_{j\in\mathcal{N}'[i]} \bar{x}_j(m^*)\tau_j(m^*)}{\sum_{j\in\mathcal{N}'[i]} \tau_j(m^*)} - \sqrt{\frac{\log(T\tilde{\Delta}^2_{m^*})}{2\sum_{j\in\mathcal{N}'[i]} \tau_j(m^*)}}. \end{aligned} \tag{A.52}$$

Therefore, the optimal arm $i_{\max}$ will not be eliminated in round m^*. Consequently, by (A.49) and (A.50) the probability that $i_{\max}$ is eliminated by a suboptimal arm i in round m^* is upper bounded by $2/(T\tilde{\Delta}^2_{m^*})$. Thus the regret contributed by case (b) is upper bounded by

$$\begin{aligned} R_b(T) &\leq \sum_{m^*=0}^{m_f} \sum_{i\in\mathcal{V}'\setminus\mathcal{A}} \frac{2}{T\tilde{\Delta}^2_{m^*}} \max_{j\in\mathcal{V}'\setminus\mathcal{A}} \Delta_j T \\ &\leq \sum_{i\in\mathcal{V}'\setminus\mathcal{A}} \sum_{m^*=0}^{m_f} \frac{2}{2^{-2m^*}} \\ &= \sum_{i\in\mathcal{V}'\setminus\mathcal{A}} \frac{2(2^{2m_f+2}-1)}{3} \\ &\leq \sum_{i\in\mathcal{V}'\setminus\mathcal{A}} \frac{2(16\cdot(\frac{8}{\sqrt{2\lambda}\epsilon})^2-1)}{3} = O(|\mathcal{V}'|). \end{aligned} \tag{A.53}$$

(c) Each arm $i \in Q$ is eliminated in (or before) round m_i. Note that arm i will be played until the last arm in $\mathcal{N}'[i]$ is eliminated or the last round $m_f \leq \lceil \log_2(8/\sqrt{2\lambda}\epsilon)\rceil$. Thus,

$$R_c(T) \leq \sum_{i\in Q} \Delta_i z_i \frac{\lambda \log(T\tilde{\Delta}^2_{m_i'})}{\tilde{\Delta}^2_{m_i'}}, \tag{A.54}$$

where

$$m_i' \leq \min\left\{\max_{j\in\mathcal{N}'[i]} m_j, \left\lceil \log_2\left(\frac{8}{\sqrt{2\lambda}\epsilon}\right)\right\rceil\right\}. \tag{A.55}$$

Therefore, the regret contributed by arms in Q is upper bounded by

$$R_Q(T) \leq \sum_{i\in Q} \Delta_i z_i \frac{32\log(T\hat{\Delta}_i^2)}{\hat{\Delta}_i^2} + O(|\mathcal{V}'|), \tag{A.56}$$

where

$$\hat{\Delta}_i = \max\{\min_{j\in\mathcal{N}'[i]} \Delta_j - 3\epsilon, \epsilon\}. \tag{A.57}$$

Moreover, for each arm $j \in \mathcal{V}' \setminus (Q \cup \mathcal{A})$, if j is eliminated before m_f, then the number of times that arm j has been played up to time T is upper bounded by

$$\mathbb{E}[\tau_j(T)] \leq \frac{32 z_j \log(T\epsilon^2)}{\epsilon^2}. \tag{A.58}$$

Otherwise, j will only be played when $L_j(t) > L_{i_{\max}}(t)$ if $i_{\max}$ is not eliminated. Since we have already shown in case (b) that the regret caused by the fact that $i_{\max}$ is eliminated before m_f is upper bounded by $O(|\mathcal{V}'|)$, we assume that $i_{\max}$ is not eliminated after m_f rounds. Using an argument similar to that in the proof of Theorem 2, we have

$$\mathbb{E}[\tau_j(T)] \leq \frac{8\log T}{\Delta_i^2}. \tag{A.59}$$

Note that the constant before $\log T$ becomes $8/\Delta_i^2$ instead of $32/\Delta_i^2$ because the reward distributions are assumed to be 1/2 sub-Gaussian. Thus, the total regret of LSDT-PSI is upper bounded by

$$R(T) \leq \sum_{j\in\mathcal{V}'\setminus(Q\cup\mathcal{A})} \Delta_j \max\left\{\frac{8\log T}{\Delta_j^2}, \frac{32 z_j \log(T\epsilon^2)}{\epsilon^2}\right\} + \sum_{i\in Q} \Delta_i z_i \frac{32\log(T\hat{\Delta}_i^2)}{\hat{\Delta}_i^2} + O(|\mathcal{V}'|). \tag{A.60}$$

A.8 Proof of Corollary 1

According to Theorem 6, for every realization of the partially revealed UIG $\mathcal{G}_\epsilon = (\mathcal{V}, \mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D)$, the expected regret of LSDT-PSI is upper bounded by

$$O\Big((|\mathcal{V} \setminus (\mathcal{Q} \cup \mathcal{A})| + \sum_{i \in \mathcal{Q}} z_i) \log T\Big), \tag{A.61}$$

where $\mathcal{Q} = \{i \in \mathcal{V}' : \Delta_i > 4\epsilon\}$.

Let $C_{\text{PSI}} = |\mathcal{V} \setminus (\mathcal{Q} \cup \mathcal{A})| + \sum_{i \in \mathcal{Q}} z_i$, we need to show that

$$\mathbb{E}_{\mathcal{E}_\epsilon^S, \mathcal{E}_\epsilon^D}[C_{\text{PSI}}] \leq \alpha(1 + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|), \tag{A.62}$$

where α is a constant independent of T and the size of the action space. We simplify the notation of expectation in (A.62) to $\mathbb{E}[C_{\text{PSI}}]$. Note that

$$\mathbb{E}[C_{\text{PSI}}] = \mathbb{E}[C_{\text{PSI}}|F]\mathbb{P}(F) + \mathbb{E}[C_{\text{PSI}}|\bar{F}]\mathbb{P}(\bar{F}) \tag{A.63}$$

where $F = \{\text{every } i \notin \mathcal{B}^* \text{ is eliminated from } \mathcal{B}_0\}$.

From Theorem 5, the probability that every arm $i \notin \mathcal{B}^*$ is not eliminated is upper bounded by $1/K^2$, therefore, we have

$$\mathbb{P}(\bar{F}) \leq \sum_{i \notin \mathcal{B}^*} \frac{1}{K^2} \leq \frac{1}{K}. \tag{A.64}$$

It is clear that $\mathbb{E}[C_{\text{PSI}}|\bar{F}] \leq K$ and $\mathbb{P}(F) \leq 1$, therefore it suffices to show that

$$\mathbb{E}[C_{\text{PSI}}|F] \leq \alpha(1 + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|) - 1. \tag{A.65}$$

Notice that given F, every arm out of $\mathcal{B}^*$ is eliminated. Besides, we assumed that $\mathcal{B}^*_{i_{\min}} \subseteq \mathcal{Q}$. Thus,

$$\mathbb{E}[C_{\text{PSI}}|F] = \mathbb{E}\Big[\sum_{i \in \mathcal{B}^*_{i_{\min}}} z_i \Big| F\Big] + |\mathcal{B}^*_{i_{\max}} \setminus \mathcal{A}|. \tag{A.66}$$

Moreover, it is clear that no matter what realization of the revealed UIG is, every arm $i \in \mathcal{B}^*_{i_{\min}}$ will not be eliminated. The fact that every arm $i \notin \mathcal{B}^*$ is eliminated only affects the probabilistic assumptions on edges with at least one end point not in $\mathcal{B}^*$. Therefore, we can claim that conditioned on F, every type-S edge between arms in $\mathcal{B}^*_{i_{\min}}$ is still observed independently with probability p_S and hence $\mathbb{E}[\sum_{i\in\mathcal{B}^*_{i_{\min}}} z_i|F]$ is equal to the expectation of the optimal value C_L of the following linear program:

$$\begin{aligned} C_L = \quad & \min_{z_1,\dots,z_L} \sum_{i=1}^{L} z_i, \\ s.t. \quad & z_i + \sum_{j\neq i} z_j \mathbb{I}\{(i,j)\in\mathcal{E}\} \geq 1, \forall i, \\ & z_i \geq 0, \forall i. \end{aligned} \tag{A.67}$$

where $L = |\mathcal{B}^*_{i_{\min}}|$ and $\forall i, j \in [L]$, $(i,j) \in \mathcal{E}$ happens independently with probability $p = p_S$. We show that $\mathbb{E}[C_L] \leq c_p$ where c_p is a constant only related to p_S. We consider a solution $z_i^* = \frac{2}{pL}, \forall i \in [L]$. We first show that $\{z_i^*\}$ is in the feasible region with probability at least $1 - 1/L$. Define $A = \{\{z_i^*\} \text{ is feasible}\}$, then

$$\mathbb{P}(\bar{A}) \leq \sum_{i=1}^{L} \mathbb{P}\Big(z_i^* + \sum_{j\neq i} z_j^* \mathbb{I}\{(i,j)\in\mathcal{E}\} < 1\Big) \tag{A.68}$$

$$= \sum_{i=1}^{L} \mathbb{P}\left(\frac{1}{L-1}\sum_{j\neq i} \mathbb{I}\{(i,j)\in\mathcal{E}\} < \frac{Lp-2}{2(L-1)}\right) \tag{A.69}$$

$$\leq \sum_{i=1}^{L} \mathbb{P}\left(\frac{1}{L-1}\sum_{j\neq i} \mathbb{I}\{(i,j)\in\mathcal{E}\} < \frac{p}{2}\right) \tag{A.70}$$

$$\leq \sum_{i=1}^{L} \mathbb{P}\left(\Big(\frac{1}{L-1}\sum_{j\neq i} \mathbb{I}\{(i,j)\in\mathcal{E}\}\Big) - p < -\frac{p}{2}\right) \tag{A.71}$$

$$\leq \sum_{i=1}^{L} e^{-2(L-1)\frac{p^2}{4}} \tag{A.72}$$

Note that the last inequality is derived through the Hoeffding inequality. Without loss of generality, we assume that $\sqrt{\frac{4\log L}{L-1}} < 1$ (otherwise, $\mathbb{E}[C_L]$ is trivially upper bounded by a constant independent of L). If $p > \sqrt{\frac{4\log L}{L-1}}$, the RHS of

(A.72) is upper bounded by $1/L$. Since it is obvious that $C_L \leq L$, we have

$$\begin{aligned} \mathbb{E}[C_L] &= \mathbb{E}[C_L|A]\mathbb{P}(A) + \mathbb{E}[C_L|\bar{A}]\mathbb{P}(\bar{A}) \\ &\leq \sum_{i=1}^{L} z_i^* + 1 = \frac{2}{p} + 1 = \beta_{p,1}. \end{aligned} \tag{A.73}$$

On the other hand, if $p \leq \sqrt{\frac{4\log L}{L-1}}$ (this is equivalent to that L is smaller than a constant that only depends on p, we denote the constant as $\beta_{p,2}$), we have $\mathbb{E}[C_L] \leq L \leq \beta_{p,2}$. In summary, if we let $c_p = \max(\beta_{p,1}, \beta_{p,2})$, we have that $\mathbb{E}[C_L] \leq c_p$. Finally, we let $\alpha = c_p + 1$ and combining with (A.66), we get the desired result in (A.65).

APPENDIX B

PROOFS OF LEMMAS AND THEOREMS IN CHAPTER 3

B.1 Proof of Lemma 1

Let $q_{\ell,s} = \prod_{\sigma=1}^{s} \exp(\gamma_1 \hat{y}_{\ell,\sigma})$ denote the weight of group ℓ at epoch s where

$$\begin{aligned} \hat{y}_{\ell,s} &= \frac{y_{\ell,s}}{q_{\ell,s}} \mathbb{I}(\ell_s = \ell) \leq \frac{\gamma_1}{L}, \\ q_{\ell,s} &= (1-\gamma_1)\frac{g_{\ell,s}}{\sum_{\ell=1}^{L} g_{\ell,s}} + \frac{\gamma_1}{L} \geq \frac{\gamma_1}{L}. \end{aligned} \tag{B.1}$$

Let $G_s = \sum_{\ell=1}^{L} g_{\ell,s}$. We have

$$\begin{aligned} \frac{G_{s+1}}{G_s} &= \sum_{\ell=1}^{L} \frac{g_{\ell,s} e^{\frac{\gamma_1 \hat{y}_{\ell,s}}{L}}}{G_s} = \sum_{\ell=1}^{L} \frac{q_{\ell,s} - \frac{\gamma_1}{L}}{1-\gamma_1} e^{\frac{\gamma_1 \hat{y}_{\ell,s}}{L}} \\ &\leq \sum_{\ell=1}^{L} \frac{q_{\ell,s} - \frac{\gamma_1}{L}}{1-\gamma_1} \left(1 + \frac{\gamma_1 \hat{y}_{\ell,s}}{L} + \left(\frac{\gamma_1 \hat{y}_{\ell,s}}{L}\right)^2\right) \\ &\leq 1 + \frac{\gamma_1/L}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s} \hat{y}_{\ell,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s} \hat{y}_{\ell,s}^2. \end{aligned} \tag{B.2}$$

The second inequality holds due to the facts that $e^x \leq 1 + x + x^2, \forall x \in [0,1]$ and $\frac{\gamma_1 \hat{y}_{\ell,s}}{L} \in [0,1]$. Notice that

$$\begin{aligned} \sum_{\ell=1}^{L} q_{\ell,s} \hat{y}_{\ell,s} &= y_{\ell_s,s}, \\ \sum_{\ell=1}^{L} q_{\ell,s} \hat{y}_{\ell,s}^2 &= q_{\ell_s,s} \frac{y_{\ell_s,s}}{q_{\ell_s,s}} \leq \hat{y}_{\ell_s,s} = \sum_{\ell=1}^{L} \hat{y}_{\ell,s}. \end{aligned} \tag{B.3}$$

Taking logarithms on both sides of (B.2) and summing over s gives

$$\ln \frac{G_{S+1}}{G_1} \leq \frac{\gamma_1/L}{1-\gamma_1} \sum_{s=1}^{S} y_{\ell_s,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{s=1}^{S} \sum_{\ell=1}^{L} \hat{y}_{\ell,s}. \tag{B.4}$$

Meanwhile, for every ℓ,

$$\begin{aligned} \ln \frac{G_{S+1}}{G_1} &\geq \ln \frac{g_{\ell,S+1}}{G_1} = \ln \frac{g_{\ell,1} e^{\frac{\gamma_1}{L} \sum_{s=1}^{S} \hat{y}_{\ell,s}}}{G_1} \\ &= \frac{\gamma_1}{L} \sum_{s=1}^{S} \hat{y}_{\ell,s} - \ln L. \end{aligned} \tag{B.5}$$

Therefore, we have

$$\sum_{s=1}^{S} y_{\ell_s,s} \geq (1-\gamma_1)\sum_{s=1}^{S}\hat{y}_{\ell,s} - \frac{L\ln L}{\gamma_1} - \frac{\gamma_1}{L}\sum_{s=1}^{S}\sum_{\ell=1}^{L}\hat{y}_{\ell,s}. \tag{B.6}$$

We take expectation on both sides of (B.6) over the randomness of $y_{\ell,s}$ for all ℓ and s (more specifically, the randomness of the arm-level EXP3 algorithm run on the ℓ-the group within the r-th epoch), conditioned on the sequence of selected arm groups $(\ell_1, ..., \ell_s)$ and past observations $\{y_{\ell_\sigma,\sigma}\}_{\sigma=1}^{s}$. Note that for every fixed sequence of reward assignment, $y_{\ell,s}$ is independent across ℓ and s. Moreover, $y_{\ell,s}$ is independent of the past history of group selection, i.e., $(\ell_1, ..., \ell_s)$. Therefore, we can obtain

$$\sum_{s=1}^{S} x_{\ell_s,s} \geq (1-\gamma_1)\sum_{s=1}^{S}\frac{x_{\ell,s}}{q_{\ell,s}}\mathbb{I}\{\ell_s=\ell\} - \frac{L\ln L}{\gamma_1} - \frac{\gamma_1}{L}\sum_{s=1}^{S}\sum_{\ell=1}^{L}\frac{x_{\ell,s}}{q_{\ell,s}}\mathbb{I}\{\ell_s=\ell\}. \tag{B.7}$$

We further take expectation over the randomness of $(\ell_1, ..., \ell_S)$ selected by the group-level EXP3 algorithm. Notice that

$$\mathbb{E}_{\ell_s}\left[\frac{x_{\ell,s}}{q_{\ell,s}}\mathbb{I}\{\ell_s=\ell\}\right] = \frac{x_{\ell,s}}{q_{\ell,s}}q_{\ell,s} + 0\cdot(1-q_{\ell,s}) = x_{\ell,s}. \tag{B.8}$$

Therefore, we have

$$\mathbb{E}_{\text{Group-EXP3}}\left[\sum_{s=1}^{S} x_{\ell_s,s}\right] \geq (1-\gamma_1)\sum_{s=1}^{S} x_{\ell,s} - \frac{L\ln L}{\gamma_1} - \gamma_1 S. \tag{B.9}$$

Since ℓ is chosen arbitrarily, by choosing $\gamma_1 = \sqrt{\frac{L\ln L}{2S}}$, we can conclude that

$$\max_{1\leq\ell\leq L}\sum_{s=1}^{S} x_{\ell,s} - \mathbb{E}_{\text{Group-EXP3}}\left[\sum_{s=1}^{S} x_{\ell_s,s}\right] \leq 2\sqrt{2SL\ln L}. \tag{B.10}$$

B.2 Proof of Lemma 4

Let $g_{\ell,s}$ and $q_{\ell,s}$ denote the weight and the selection probability of group ℓ at epoch s. Let $G_s = \sum_{\ell=1}^{L} g_{\ell,s}$. For every $h^S = (h_1, ..., h_S)$ such that $H(h^S) \leq V$,

consider the V-partition of the time horizon $[1, S]$:

$$[S_1, ..., S_2), [S_2, ..., S_3), ..., [S_V, ...S_{V+1}), \tag{B.11}$$

where $S_1 = 1$ and $S_{V+1} = S + 1$, such that h_s is fixed for $s \in [S_v, S_{v+1}), \forall v = 1, ..., V$. For each segment $[S_v, S_{v+1})$:

$$\begin{aligned}
\frac{G_{s+1}}{G_s} &= \sum_{\ell=1}^{L} \frac{g_{\ell,s+1}}{G_s} = \sum_{\ell=1}^{L} \frac{g_{\ell,s} e^{\gamma_1 \hat{y}_{\ell,s}/L} + \frac{e\alpha G_s}{L}}{G_s} \\
&= \sum_{\ell=1}^{L} \frac{q_{\ell,s} - \frac{\gamma_1}{L}}{1-\gamma_1} e^{\gamma_1 \hat{y}_{\ell,s}/L} + e\alpha \\
&\leq \sum_{\ell=1}^{L} \frac{q_{\ell,s} - \frac{\gamma_1}{L}}{1-\gamma_1} \left(1 + \frac{\gamma_1}{L}\hat{y}_{\ell,s} + \left(\frac{\gamma_1}{L}\right)^2 \hat{y}_{\ell,s}^2\right) + e\alpha \\
&\leq 1 + \frac{\gamma_1/L}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s}\hat{y}_{\ell,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s}\hat{y}_{\ell,s}^2 + e\alpha.
\end{aligned} \tag{B.12}$$

We can further derive that

$$\begin{aligned}
\ln \frac{G_{s+1}}{G_s} &\leq \frac{\gamma_1/L}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s}\hat{y}_{\ell,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{\ell=1}^{L} q_{\ell,s}\hat{y}_{\ell,s}^2 + e\alpha \\
&\leq \frac{\gamma_1/L}{1-\gamma_1} y_{\ell_s,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{\ell=1}^{L} \hat{y}_{\ell,s} + e\alpha.
\end{aligned} \tag{B.13}$$

Summing over $s = S_v, ..., S_{v+1} - 1$, we have

$$\ln \frac{G_{S_{v+1}}}{G_{S_v}} \leq \frac{\gamma_1/L}{1-\gamma_1} \sum_{s=S_v}^{S_{v+1}-1} y_{\ell_s,s} + \frac{(\gamma_1/L)^2}{1-\gamma_1} \sum_{s=S_v}^{S_{v+1}-1} \sum_{\ell=1}^{L} \hat{y}_{\ell,s} + e\alpha(S_{v+1} - S_v). \tag{B.14}$$

By abuse of notation, we let h_v be the action in this segment and then

$$\begin{aligned}
g_{h_v,S_{v+1}} &\geq g_{h_v,S_v+1} \exp\left(\frac{\gamma_1}{L} \sum_{s=S_v+1}^{S_{v+1}-1} \hat{y}_{h_v,s}\right) \\
&\geq \frac{e\alpha}{L} G_{S_v} \exp\left(\frac{\gamma_1}{L} \sum_{s=S_v+1}^{S_{v+1}-1} \hat{y}_{h_v,s}\right) \\
&\geq \frac{\alpha}{L} G_{S_v} \exp\left(\frac{\gamma_1}{L} \sum_{s=S_v}^{S_{v+1}-1} \hat{y}_{h_v,s}\right),
\end{aligned} \tag{B.15}$$

where the last inequality holds since

$$\hat{y}_{h_v,s} \leq 1/q_{h_v,s} \leq L/\gamma_1, \forall s. \tag{B.16}$$

Therefore, we have

$$\ln \frac{G_{S_{v+1}}}{G_{S_v}} \geq \ln\left(\frac{\alpha}{L}\right) + \frac{\gamma_1}{L} \sum_{s=S_v}^{S_{v+1}-1} \hat{y}_{h_v,s}, \tag{B.17}$$

and as a consequence,

$$\sum_{s=S_v}^{S_{v+1}-1} y_{\ell_s,s} \geq (1-\gamma_1) \sum_{s=S_v}^{S_{v+1}-1} \hat{y}_{h_v,s} - \frac{L\ln(L/\alpha)}{\gamma_1} - \frac{\gamma_1}{L} \sum_{s=S_v}^{S_{v+1}-1} \sum_{\ell=1}^{L} \hat{y}_{\ell,s} - \frac{e\alpha L(S_{v+1}-S_v)}{\gamma_1}. \tag{B.18}$$

We sum over all segments v and take expectation on the both side of the inequality, using a similar argument as that used in the proof of Lemma 1, we can obtain that

$$\sum_{s=1}^{S} x_{h_s,s} - \mathbb{E}_{\text{Group-EXP3.S}}\left[\sum_{s=1}^{S} x_{\ell_s,s}\right] \leq \gamma_1 S + \frac{LV\ln(LS)}{\gamma_1} + \gamma_1 S + \frac{eL}{\gamma_1} \tag{B.19}$$

if we choose $\alpha = 1/S$. We further choose $\gamma_1 = \sqrt{\frac{LV\ln(LS)}{S}}$ to obtain the conclusion of Lemma 3 (assuming without loss of generality that $V\ln(LS) \geq e$).

B.3 Proof of Theorem 11

The proof follows the same structure with the one in the proof of Theorem 7. Let $i_{\max}$ be the arm with the greatest cumulative reward. Let $\mathcal{A}_{\ell_{\max}}$ and $\mathcal{B}_{h_{\max}}^{\ell_{\max}}$ be the group and subgroup to which $i_{\max}$ belongs. We decompose the expected weak regret of HLMC with a three-level hierarchy as follows:

$$\begin{aligned}\mathbb{E}_{\text{HLMC}}[R_{\text{W}}(T)] &\leq (C_{\max} - C'_{\max}) + (C'_{\max} - C''_{\max}) + (C''_{\max} - C_{\text{HLMC}}) \\ &= R_1(T) + R_2(T) + R_3(T),\end{aligned} \tag{B.20}$$

where

$$C_{\max} = \sum_{t=1}^{T} r_{i_{\max},t}, \tag{B.21}$$

$$C'_{\max} = \sum_{s=1}^{S_1} \sum_{\tau=1}^{S_2} \mathbb{E}_{\text{Arm-EXP3}(\mathcal{B}_{h_{\max}}^{\ell_{\max}})} \left[\sum_{t \in \mathcal{I}_\tau^s} r_{i_t,t} \right], \tag{B.22}$$

$$C''_{\max} = \sum_{s=1}^{S_1} \mathbb{E}_{\text{Subgroup-EXP3}(\mathcal{A}_{\ell_{\max}})} \left[\sum_{\tau=1}^{S_2} \mathbb{E}_{\text{Arm-EXP3}(\mathcal{B}_{h_\tau}^{\ell_{\max}})} \left[\sum_{t \in \mathcal{I}_\tau^s} r_{i_t,t} \right] \right], \tag{B.23}$$

$$\begin{aligned} C_{\text{HLMC}} &= \mathbb{E}_{\text{HLMC-3L}} \left[\sum_{t=1}^{T} r_{i_t,t}] \right], \\ &= \mathbb{E}_{\text{Group-EXP3}} \left[\sum_{s=1}^{S_1} \mathbb{E}_{\text{Subgroup-EXP3}(\mathcal{A}_{\ell_s})} \left[\sum_{\tau=1}^{S_2} \mathbb{E}_{\text{Arm-EXP3}(\mathcal{B}_{h_\tau}^{\ell_s})} \left[\sum_{t \in \mathcal{I}_\tau^s} r_{i_t,t} \right] \right] \right]. \end{aligned} \tag{B.24}$$

Specifically, $R_1(T)$ corresponds to the arm-level reward loss due to not playing the best arm, assuming that group $\mathcal{A}_{\ell_{\max}}$ and subgroup $\mathcal{B}_{h_{\max}}^{\ell_{\max}}$ are selected at all epochs and subepochs. By applying Lemma 2 at every subepoch, we have

$$R_1(T) \leq 2S_1 S_2 \sqrt{2S_3 N_3 \ln N_3}. \tag{B.25}$$

For $R_1(T)$, which corresponds to the subgroup-level reward loss due to not selecting subgroup $\mathcal{B}_{h_{\max}}^{\ell_{\max}}$ at all subepochs, assuming that group $\mathcal{A}_{\ell_{\max}}$ is selected at all epochs, we apply Lemma 1 at every epoch by defining

$$x_{h,\tau}^{\ell,s} = \mathbb{E}_{\text{Arm-EXP3}(\mathbb{B}_h^\ell)} \left[\frac{1}{S_3} \sum_{t \in \mathcal{I}_\tau^s} r_{i_t,t} \right]. \tag{B.26}$$

Then we obtain that

$$R_2(T) \leq 2S_1 S_3 \sqrt{2S_2 N_2 \ln N_2}. \tag{B.27}$$

Finally, $R_3(T)$ corresponds to the group-level reward loss due to not selecting group $\mathcal{A}_{\ell_{\max}}$ at all epochs. By defining

$$z_{\ell,s} = \mathbb{E}_{\text{Subgroup-EXP3}(\mathbb{A}_\ell)} \left[\frac{1}{S_2} \sum_{\tau=1}^{S_2} x_{h_\tau,\tau}^{\ell,s} \right], \tag{B.28}$$

we can apply Lemma 1 again to obtain that

$$R_3(T) \leq 2S_2S_3\sqrt{2S_1N_1\ln N_1}. \tag{B.29}$$

The upper bound in Theorem 11 is obtained by combining (B.25), (B.27), and (B.29) together and selecting

$$S_i = \left\lceil \frac{T^{1/3}(N_i \ln N_i)^{2/3}}{(\prod_{j\neq i} N_j \ln N_j)^{1/3}} \right\rceil, \forall i = 1, 2, 3.$$

APPENDIX C

ADDITIONAL RESULTS AND PROOFS IN CHAPTER 3

C.1 Implementation of PSLinUCB-Hybrid

Algorithm 11: PSLinUCB-Hybrid

Input: $\alpha > 0, \omega \in \mathbb{N}^+, \delta > 0, k = d \times m$.

Initialization: $\mathbf{A}_0^{pre}, \mathbf{A}_0^{cum} = \mathbf{I}_k$. $\mathbf{b}_0^{pre}, \mathbf{b}_0^{cum} = \mathbf{0}_{k\times 1}$.

for $t = 1, 2, ..., T$ **do**

// Parameter Estimation and Arm Selection

Observe the feature vector x_{u_t} of the current user u_t and the cross-feature $z_{u_t,a}$ for every arm $a \in \mathcal{A}_t$.

$\hat{\beta}^{cum} = (\mathbf{A}_0^{cum})^{-1}\mathbf{b}_0^{cum}$.

for $a \in \mathcal{A}_t$ **do**

if a is new **then**

$\mathbf{A}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{I}_d$, $\mathbf{b}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times 1}$, $\mathbf{B}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times k}$, $SW_a \leftarrow \emptyset$.

$\hat{\theta}_a^{cum} \leftarrow (\mathbf{A}_a^{cum})^{-1}(\mathbf{b}_a^{cum} - \mathrm{B}_a^{cum}\hat{\beta}^{cum})$.

$p_{t,a} \leftarrow x_{u_t}^T\hat{\theta}_a^{cum} + z_{u_t,a}^T\hat{\beta}^{cum} + \alpha\sqrt{s_{t,a}}$.

Play $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$, obtain reward $r_{u_t,a_t}(t)$.

Append $(x_{u_t}, z_{u_t,a_t}, r_{u_t,a_t}(t))$ to the end of SW_{a_t}.

Update $\mathbf{A}_0^{cum}, \mathbf{b}_0^{cum}, \mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}, \mathbf{b}_{a_t}^{cum}$ using (4.6).

Update $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}, \mathbf{b}_{a_t}^{cur}$ in the same way with that in updating $\mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}, \mathbf{b}_{a_t}^{cum}$ (replace *cum* with *cur*).

(continued in the next page.)

Algorithm 12: PSLinUCB-Hybrid (continued)

//Change Detection and Model Update

if $|SW_{a_t}| \geq \omega$ **then**

$\hat{\beta}^{pre} \leftarrow (\mathbf{A}_0^{pre})^{-1}\mathbf{b}_0^{pre}$, $\hat{\theta}_{a_t}^{pre} \leftarrow (\mathbf{A}_{a_t}^{pre})^{-1}(\mathbf{b}_{a_t}^{pre} - \mathbf{B}_{a_t}^{pre}\hat{\beta}^{pre})$.

Let $SW_{a_t} = \{(x_s, z_s, r_s)\}_{s=1}^{\omega}$.

if $|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T \hat{\theta}_{a_t}^{pre} + z_s^T \hat{\beta}^{pre} - r_s)| \geq \delta$ **then**

Update $\mathbf{A}_0^{cum}, \mathbf{b}_0^{cum}, \mathbf{A}_0^{pre}, \mathbf{b}_0^{pre}$ using (4.8).

$\mathbf{A}_0^{pre} \leftarrow \mathbf{A}_0^{cum}, \mathbf{b}_0^{pre} \leftarrow \mathbf{b}_0^{cum}, SW_{a_t} \leftarrow \emptyset$.

$\mathbf{A}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{A}_{a_t}^{cur}$, $\mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{I}_d$.

$\mathbf{B}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{B}_{a_t}^{cur}$, $\mathbf{B}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times k}$.

$\mathbf{b}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{b}_{a_t}^{cur}$, $\mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times 1}$.

else

$(x_1, z_1, r_1) \leftarrow \text{Popleft}(SW_{a_t})$.

Update $\mathbf{A}_0^{pre}, \mathbf{b}_0^{pre}, \mathbf{A}_{a_t}^{pre}, \mathbf{B}_{a_t}^{pre}, \mathbf{b}_{a_t}^{pre}$ according to (4.6) (replace *cum* with *pre* and $(x_{u_t}, z_{u_t,a_t}, r_{u_t,a_t}(t))$ with (x_1, z_1, r_1)).

Update $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}, \mathbf{b}_{a_t}^{cur}$ in the same way with that in updating $\mathbf{A}_{a_t}^{pre}, \mathbf{B}_{a_t}^{pre}, \mathbf{b}_{a_t}^{pre}$ (replace *pre* with *cur* and operation + with −).

C.2 Proof of Theorem 12

Define events $F_i = \{\tau_i \geq \nu_i\}, 1 \leq i \leq M-1$ and $D_i = \{\tau_i < \nu_i + L/2\}, 1 \leq i \leq M-2$, $D_{M-1} = \{\tau_{M-1} \leq T\}$. Then we have

$$\begin{aligned} & \mathbb{E}[R(T)] \leq \mathbb{E}[R(T)\mathbb{I}(F_1)] + T(1 - \mathbb{P}(F_1)) \\ \leq\ & \mathbb{E}[R(\nu_1)\mathbb{I}(F_1)] + \mathbb{E}[R(T) - R(\nu_1)] + K \\ \leq\ & (\delta_0 + \gamma)\nu_1 + 2\alpha\sqrt{2\nu_1 dK\log\frac{\nu_1}{d}} + 2K + \mathbb{E}[R(T) - R(\nu_1)] \end{aligned} \tag{C.1}$$

Note that the first inequality follows from Lemma 6 on bounding the probability of false alarm in the first fist stationary segment $[0, \nu_1]$ provided that b satisfies (C.22) and $c = \sqrt{\frac{2}{\omega}\log(2T)}$. The second inequality follows from Lemma 5 on $[0, \nu_1]$. The next step is to bound $\mathbb{E}[R(T) - R(\nu_1)]$, which satisfies

$$\begin{aligned}
& \mathbb{E}[R(T) - R(\nu_1)] \\
\leq\ & \mathbb{E}[R(T) - R(\nu_1)|F_1 D_1] + T(1 - \mathbb{P}(F_1 D_1)) \\
=\ & \mathbb{E}[R(T) - R(\nu_1)|F_1 D_1] + T(\mathbb{P}(\bar{F}_1 D_1) + \mathbb{P}(F_1 \bar{D}_1) + \mathbb{P}(\bar{F}_1 \bar{D}_1)) \qquad \text{(C.2)} \\
\leq\ & \mathbb{E}[R(T) - R(\tau_1)|F_1 D_1] + \mathbb{E}[R(\tau_1) - R(\nu_1)|F_1 D_1] + 2K \\
\leq\ & \tilde{\mathbb{E}}[R(T - \tau_1)] + \mathbb{E}[\tau_1 - \nu_1|F_1 D_1] + 2K \\
\leq\ & \tilde{\mathbb{E}}[R(T - \tau_1)] + \omega\lceil K/\gamma \rceil + 2K,
\end{aligned}$$

where the second inequality holds due to the following facts

1. $\mathbb{P}(\bar{F}_1 D_1) = \mathbb{P}(\bar{F}_1) \leq KT^{-1}$ according to Lemma 6, provided that b satisfies (C.22) and $c = \sqrt{\frac{2}{\omega}\log(2T)}$;
2. $\mathbb{P}(F_1 \bar{D}_1) = \mathbb{P}(\bar{D}_1) \leq 2T^{-2}$ according to Lemma 7;
3. $\mathbb{P}(\bar{F}_1 \bar{D}_1) = 0$ since $\bar{F}_1$ and $\bar{D}_1$ cannot happen simultaneously.

The third inequality holds due to the fact that the learning process is restarted once a change is detected and $\tilde{\mathbb{E}}$ is the expectation taken over the random process induced by the learning algorithm after the first detected change time.

Finally, if we recursively upper bound $\tilde{\mathbb{E}}[R(T - \nu_1)]$ by the same arguments as above and repeat the process for $M - 1$ times, we have

$$\begin{aligned}
\mathbb{E}[R(T)] \leq\ & (\delta_0 + \gamma)T + \sum_{i=1}^{M} 2\alpha\sqrt{2\nu_i dK\log\frac{\nu_i}{d}} \\
& + 4KM + \omega M\lceil K/\gamma \rceil. \qquad \text{(C.3)}
\end{aligned}$$

Let $\delta_0 = 1/T$, $\gamma = \sqrt{\frac{KM\omega}{T}}$, $\alpha > \sqrt{2d\log\frac{T}{\delta_0}}$, and apply Cauchy-Schwatz inequality to the second term, we can obtain

$$\mathbb{E}[R(T)] \leq \tilde{C}_1\sqrt{TMK\omega} + \tilde{C}_2\sqrt{TMKd^2\log^2 T}. \tag{C.4}$$

C.3 Proof of Lemma 5

Let $\mathbb{I}(\cdot)$ be the indicator function and R_{a_t} be the one-step regret at time t when the algorithm plays arm a_t. The expected cumulative regret can be partitioned as follows:

$$\begin{aligned}
\mathbb{E}[R(T)] &= \mathbb{E}[R(T)\mathbb{I}(\tau_1 \leq T)] + \mathbb{E}[R(T)\mathbb{I}(\tau_1 > T)] \\
&\leq T\cdot\mathbb{P}(\tau_1 \leq T) + \mathbb{E}[R(T)\mathbb{I}(\tau_1 > T)] \\
&\leq T\cdot\mathbb{P}(\tau_1 \leq T) + \sum_{t=1}^{T}\mathbb{E}[R_{a_t}\mathbb{I}(\tau_1 > T, a_t \text{ is random selected})] \\
&\quad + \sum_{t=1}^{T}\mathbb{E}[R_{a_t}\mathbb{I}(\tau_1 > T, a_t \text{ is selected by UCB index})].
\end{aligned} \tag{C.5}$$

According to the algorithm, it is not difficult to see that the second term on the RHS of the above inequality satisfies

$$\sum_{t=1}^{T}\mathbb{E}[R_{a_t}\mathbb{I}(\tau_1 > T, a_t \text{ is random selected})] \leq K\cdot\left\lceil\frac{T\gamma}{K}\right\rceil \leq K + T\gamma. \tag{C.6}$$

For the last term, we have:

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}[R_{a_t}\mathbb{I}(\tau_1 > T, a_t \text{ is selected by UCB index})] \\
\leq\ &\sum_{t=1}^{T}\mathbb{E}[(r_{a_t^*} - r_{a_t})\mathbb{I}(\forall a\in\mathcal{A}, \text{no change detected up to time } t-1, \\
&\qquad\qquad a_t \text{ is selected by UCB index})] \\
=\ &\sum_{t=1}^{T}(x_{u_t}^T\theta_{a_t^*} - x_{u_t}^T\theta_{a_t})\mathbb{I}(\forall a\in\mathcal{A}, \text{no change detected up to time } t-1, \\
&\qquad\qquad a_t \text{ is selected by UCB index}).
\end{aligned} \tag{C.7}$$

Note that if no change has been detected up to time $t-1$, the estimation of $\theta_a, \forall a \in \mathcal{A}$ has not been restarted and thus, $\hat{\theta}_a$ is calculated based on all past observations. Thus, according to the algorithm, the RHS of (C.7) is upper bounded by

$$\begin{aligned}
& \sum_{t=1}^{T} (x_{u_t}^T \theta_{a_t^*} - x_{u_t}^T \theta_{a_t}) \mathbb{I}(\forall a \in \mathcal{A}, \text{no change detected up to time } t-1, \\
& \qquad\qquad a_t \text{ is selected by UCB index}) \\
\leq & \sum_{t=1}^{T} (x_{u_t}^T \hat{\theta}_{a_t^*} + \|\hat{\theta}_{a_t^*} - \theta_{a_t^*}\|_{A_{a_t^*}(t-1)} \cdot \|x_{u_t}\|_{A_{a_t^*}^{-1}(t-1)} - x_{u_t}^T \theta_{a_t}) \mathbb{I}(a_t \text{ is selected by UCB index}) \\
\leq & \sum_{t=1}^{T} x_{u_t}^T \hat{\theta}_{a_t} + \alpha \|x_{u_t}\|_{A_{a_t}^{-1}(t-1)} - x_{u_t}^T \theta_{a_t} \qquad \text{(C.8)} \\
\leq & \sum_{t=1}^{T} 2\alpha \|x_{u_t}\|_{A_{a_t}^{-1}(t-1)},
\end{aligned}$$

where $A_a(t-1) = I + \sum_{\tau=1}^{t-1} \mathbb{I}(a_\tau = a) x_{u_\tau} x_{u_\tau}^T$ and $\|x\|_A = \sqrt{x^T A x}$. The first inequality simply follows from Lemma 2 in [38]. By selecting $\alpha > \|\hat{\theta}_a - \theta_a\|_{A_a(t-1)}, \forall a \in \mathcal{A}$ and t, the second inequality follows from the fact that the UCB index of a_t is greater than a_t^* at time t. The last inequality also holds according to Lemma 2 in [38] and the selection of α. It has been shown in [1] (specifically, Theorem 2) that for an arm a and any constant $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\|\hat{\theta}_a - \theta_a\|_{A_a(t-1)} \leq 1 + \sqrt{d \log\left(\frac{1+t}{\delta}\right)}. \qquad \text{(C.9)}$$

Therefore, if we choose $\delta = \delta_0/K$ and $\alpha > \sqrt{2d \log(KT/\delta_0)}$, then with probability at least $1-\delta_0$, we have $\alpha > \|\hat{\theta}_a - \theta_a\|_{A_a(t-1)}$, $\forall a \in \mathcal{A}$ and t, and consequently, (C.8) holds with probability $1-\delta_0$. Moreover, with probability δ_0 when (C.8) does not hold, the cumulative regret is trivially upper bounded by T.

Furthermore, let $\mathcal{T}_a$ be the set of time steps when arm a is selected up to time

T, the RHS of (C.8) satisfies:

$$\sum_{t=1}^{T} 2\alpha \|x_{u_t}\|_{A_{a_t}^{-1}(t-1)} \tag{C.10}$$

$$= 2\alpha \sum_{a\in\mathcal{A}} \sum_{t\in\mathcal{T}_a} \|x_{u_t}\|_{A_a^{-1}(t-1)} \tag{C.11}$$

$$\leq 2\alpha \sum_{a\in\mathcal{A}} \sqrt{|\mathcal{T}_a| \sum_{t\in\mathcal{T}_a} \|x_{u_t}\|^2_{A_a^{-1}(t-1)}} \tag{C.12}$$

$$\leq 2\alpha \sum_{a\in\mathcal{A}} \sqrt{|\mathcal{T}_a| \cdot 2d \log\left(1 + \frac{|\mathcal{T}_a|}{d}\right)} \tag{C.13}$$

$$\leq 2\alpha \sqrt{2TdK \log\left(\frac{T}{d}\right)}, \tag{C.14}$$

where the first and third inequalities hold by Cauchy-Schwarz inequality and the second inequality hold by Lemma 11 in [1] and Lemma 3 in [38]. In summary, the expected cumulative regret under the stationary environment is upper bounded by

$$\mathbb{E}[R(T)] \leq T \cdot \mathbb{P}(\tau_1 \leq T) + K + T(\gamma + \delta_0) + 2\alpha \sqrt{2TdK \log\left(\frac{T}{d}\right)}. \tag{C.15}$$

C.4 Proof of Lemma 6

Define $\tau_{a,1}$ be the first detection time of arm a. Then $\tau_1 = \min_{a\in\mathcal{A}}\{\tau_{a,1}\}$ and

$$\mathbb{P}(\tau_1 \leq T) \leq \sum_{a\in\mathcal{A}} \mathbb{P}(\tau_{a,1} \leq T). \tag{C.16}$$

Let $\{(x_i, r_{a,i})\}_{i=t-\omega+1,..,t}$ be the last ω observations of arm a before time t and define

$$S_{a,t} = \frac{2}{\omega} \left| \sum_{i=t-\omega/2+1}^{t} x_i^T \tilde{\theta}_a(t-\omega+1, t-\omega/2) - r_{a,i} \right|, \tag{C.17}$$

where $\tilde{\theta}_a(t-\omega+1, t-\omega/2)$ is the estimate of θ_a based on the observations in $\{(x_i, r_{a,i})\}_{i=t-\omega+1}^{t-\omega/2}$. According to the modified PSLinUCB algorithm, we have

$$\tau_{a,1} = \inf\{t \leq \omega : S_{a,t} \geq b + c\}. \tag{C.18}$$

Moreover, we define $\tau_{a,1}^{(j)} = \inf\{t = j + n\omega, n \in \mathbb{Z}^+ : S_{a,t} \geq b + c\}$. Then it is not difficult to see that at each $t_n = j + n\omega, n \in \mathbb{Z}^+$, the observations used for change detection are disjoint and thus, $\tau_{a,1}^{(j)}$ is a random variable with the geometric distribution:

$$\mathbb{P}(\tau_{a,1}^{(j)} = n\omega + j) = p(1-p)^{n-1}, \tag{C.19}$$

where $p = \mathbb{P}(S_{a,\omega} > b + c)$ and thus

$$\mathbb{P}(\tau_{a,1} \leq T) \leq \omega(1 - (1-p)^{T/\omega}). \tag{C.20}$$

To upper bound p, we have

$$\begin{aligned} & \mathbb{P}(S_{a,\omega} > b + c) \\ \leq \; & \mathbb{P}\left(\frac{2}{\omega}\left|\sum_{i=t-\omega/2+1}^{t} x_i^T \tilde{\theta}_a(t-\omega+1, t-\omega/2) - x_i^T \theta_a\right| > b\right) \\ & + \mathbb{P}\left(\frac{2}{\omega}\left|\sum_{i=t-\omega/2+1}^{t} x_i^T \theta_a - r_{a,i}\right| > c\right). \end{aligned} \tag{C.21}$$

For the first term in the RHS of (C.21), if we choose b to satisfy the following condition for any t:

$$b \geq \sqrt{2d\log\left(\frac{\omega}{\delta_1}\right)}\left(\frac{2}{\omega}\sum_{i=t-\omega/2+1}^{t} \|x_i\|_{\tilde{A}_a^{-1}(t-\omega+1,t-\omega/2)}\right), \tag{C.22}$$

where $\tilde{A}_a(t-\omega+1, t-\omega/2) = I + \sum_{i=t-\omega+1}^{t-\omega/2} x_i x_i^T$, then the first term in the RHS of (C.21) is upper bounded by δ_1 according to Lemma 2 in [38] and Theorem 1 in [1]. The second term can be bounded by $2\exp(-\omega c^2)$ according to the Hoeffding's inequality. Let $\delta_1 = 1/(2T^2)$ and $c \geq \sqrt{\frac{2}{\omega}\log(2T)}$, it is not difficult to see that

$$p = \mathbb{P}(S_{a,\omega} > b + c) \leq T^{-2}. \tag{C.23}$$

Since $(1-x)^a > 1 - ax$ for $a > 1$ and $x \in (0,1)$, it can be shown that

$$\mathbb{P}(\tau_1 \leq T) \leq \sum_{a \in \mathcal{A}} \mathbb{P}(\tau_{a,1} \leq T) \leq KT^{-1}. \tag{C.24}$$

C.5 Proof of Lemma 7

Notice that the round-robin exploration in the algorithm guarantees that within $L/2$ time steps, each arm is sampled at least $\omega/2$ times. We upper bound the probability of $\{\tau_1 > \nu_1 + L/2\}$ as follows: consider a be the arm at which the change point occurs. Let t be the time step when a is sampled $\omega/2$ times in the new stationary segment (notice that $t \leq \nu_1 + L/2$). The change at a is not detected only if one of the following events happens:

$$E_1 = \left\{ \frac{2}{\omega} \left| \sum_{i=t-\omega/2+1}^{t} x_i^T \tilde{\theta}_a(t-\omega+1, t-\omega/2) - x_i^T \theta_a^{\text{old}} \right| > b \right\}, \tag{C.25}$$

$$E_2 = \left\{ \frac{2}{\omega} \left| \sum_{i=t-\omega/2+1}^{t} x_i^T \theta_a^{\text{new}} - r_{a,i} \right| > c \right\}, \tag{C.26}$$

$$E_3 = \left\{ \frac{2}{\omega} \left| \sum_{i=t-\omega/2+1}^{t} x_i^T \theta_a^{\text{old}} - x_i^T \theta_a^{\text{new}} \right| < b + c \right\}, \tag{C.27}$$

where θ_a^{new} and θ_a^{old} correspond to the ground-truth preference vectors of arm a after and before the change point. Therefore,

$$\mathbb{P}(\tau_1 > \nu_1 + L/2) \leq \mathbb{P}(E_1) + \mathbb{P}(E_2) + \mathbb{P}(E_3). \tag{C.28}$$

The first two terms has been shown to be upper bounded by $1/T^2$ in the proof of Lemma 6 and the last term equals 0 under the condition that $\Delta \geq b+c$. Therefore, the conclusion in Lemma 7 holds.

BIBLIOGRAPHY

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, 1995.

[3] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[4] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107, 2013.

[5] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

[6] Noga Alon, Nicolo Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

[7] Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, pages 217–226, 2009.

[8] Peter Auer. Using confidence bounds for exploitation-exploration tradeoffs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

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

[10] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

[11] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[12] Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

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

[14] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

[15] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Stochastic bandits with side observations on networks. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, pages 289–300, 2014.

[16] Valeriĭ Vladimirovich Buldygin and IU V Kozachenko. *Metric characterization of random variables and random processes*, volume 188. American Mathematical Soc., 2000.

[17] Yang Cao, Zheng Wen, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 418–427, 2019.

[18] Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 142–151, 2012.

[19] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

[20] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[21] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011.

[22] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Regret minimisation in multi-armed bandits using bounded arm memory. *arXiv preprint arXiv:1901.08387*, 2019.

[23] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: general framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[24] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[25] Richard Combes, Stefan Magureanu, and Alexandre Proutiere. Minimal exploration in structured stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 1763–1771, 2017.

[26] Richard Combes and Alexandre Proutiere. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, pages 521–529, 2014.

[27] Derek G Corneil, Hiryoung Kim, Sridhar Natarajan, Stephan Olariu, and Alan P Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.

[28] T Cover and M Hellman. The two-armed-bandit problem with time-invariant finite memory. *IEEE Transactions on Information Theory*, 16(2):185–195, 1970.

[29] Thomas M Cover. A note on the two-armed bandit problem with finite memory. *Information and Control*, 12(5):371–377, 1968.

[30] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, pages 355–366, 2008.

[31] Benoit Duvocelle, Panayotis Mertikopoulos, Mathias Staudigl, and Dries Vermeulen. Learning in time-varying games. *arXiv preprint arXiv:1809.03066*, 2018.

[32] Peter Frankl and Hiroshi Maehara. Open-interval graphs versus closed-interval graphs. *Discrete Mathematics*, 63(1):97–100, 1987.

[33] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

[34] Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Conference on Learning Theory*, pages 359–376, 2011.

[35] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 174–188. Springer, 2011.

[36] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[37] Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15(3):251–261, 1994.

[38] Xueying Guo, Xiaoxiao Wang, and Xin Liu. AdaLinUCB: Opportunistic learning for contextual bandits. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[39] Manjesh Kumar Hanawal and Venkatesh Saligrama. Efficient detection and localization on graph structured data. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5590–5594. IEEE, 2015.

[40] Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[41] Cédric Hartland, Nicolas Baskiotis, Sylvain Gelly, Michèle Sebag, and Olivier Teytaud. Change point detection and meta-bandits for online learning in dynamic environments. In *Confrence Francophone sur lapprentissage automatique, Cepadues*, page 237250, 2007.

[42] Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michele Sebag. Multi-armed bandit, dynamic environments and meta-bandits, 2006. In *NIPS-2006 Workshop, Online Trading Between Exploration and Exploitation, Whistler, Canada*, 2006.

[43] Lars Holst. On the lengths of the pieces of a stick broken at random. *Journal of Applied Probability*, 17(3):623–634, 1980.

[44] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 681–690, 2008.

[45] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[46] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[47] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems*, 20:1, 2007.

[48] C Lekkeikerker and J Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51(1):45–64, 1962.

[49] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670, 2010.

[50] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 297–306, 2011.

[51] David Liau, Zhao Song, Eric Price, and Ger Yang. Stochastic multi-armed bandits in constant space. In *International Conference on Artificial Intelligence and Statistics*, pages 386–394, 2018.

[52] Keqin Liu and Qing Zhao. Adaptive shortest-path routing under unknown and stochastically varying link states. In *International Symposium on Model-*

ing and Optimization in Mobile, Ad Hoc and Wireless Networks, pages 232–237. IEEE, 2012.

[53] Chi-Jen Lu and Wei-Fu Lu. Making online decisions with bounded memory. In *International Conference on Algorithmic Learning Theory*, pages 249–261. Springer, 2011.

[54] Thodoris Lykouris, Vasilis Syrgkanis, and Éva Tardos. Learning and efficiency in games with dynamic population. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 120–129. SIAM, 2016.

[55] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bound and optimal algorithms. In *Conference on Learning Theory*, pages 975–999, 2014.

[56] Herbert Robbins. A sequential decision problem with a finite memory. *Proceedings of the National Academy of Sciences of the United States of America*, 42(12):920, 1956.

[57] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[58] Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[59] Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems*, pages 1602–1610, 2011.

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

[61] Sattar Vakili, Keqin Liu, and Qing Zhao. Deterministic sequencing of exploration and exploitation for multi-armed bandit problems. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):759–767, 2013.

[62] Sattar Vakili and Qing Zhao. Achieving complete learning in multi-armed bandit problems. In *2013 Asilomar Conference on Signals, Systems and Computers*, pages 1778–1782. IEEE, 2013.

[63] Michal Valko. *Bandits on graphs and structures*. PhD thesis, École normale supérieure de Cachan, 2016.

[64] Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, pages 46–54, 2014.

[65] Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 495–504, 2018.

[66] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, pages 2080–2090, 2019.

[67] Xiao Xu and Qing Zhao. Distributed no-regret learning in multiagent systems: Challenges and recent developments. *IEEE Signal Processing Magazine*, 37(3):84–91, 2020.

[68] H Peyton Young. *Strategic learning and its limits*. OUP Oxford, 2004.

[69] Jia Yuan Yu and Shie Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1177–1184, 2009.

[70] Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *International Conference on Machine Learning*, pages 7683–7692, 2019.