Structured and Correlated Multi-Armed Bandits: Algorithms, Theory and Applications

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Samarth Gupta

B.Tech and M.Tech in Electrical Engineering, IIT Bombay

Carnegie Mellon University
Pittsburgh, PA

May 2022

Acknowledgements

It brings me great joy that this section is much longer than what I thought it would be when I started my PhD. Over the last 5 years, I have had the fortune of meeting and working with some of the brightest, friendliest and most kind people, who have had a significant impact on both my thesis and my life at CMU.

First, I would like to thank Gauri Joshi and Osman Yağan for being my amazing PhD advisors. Five years ago, when I started my PhD, we knew next to nothing about Multi-Armed Bandits and it fills me with a lot of pride that today we have been able to write several research papers as a team. None of this would have been possible if it were not for their guidance on the projects, optimism about the outcome and their constant care about my well-being. The journey was far from being a straightforward one, and there were way too many roadblocks. In all these times, they put more faith in me than I ever had in myself. I am very thankful to Gauri and Osman for instilling such confidence in my research, writing and presentations. Their ever-present support allowed me to pursue my research interests with complete freedom, without having to worry about the final outcomes. I have wholeheartedly enjoyed my time at CMU and I am extremely grateful to Gauri and Osman for helping me navigate this beautiful PhD journey.

Next, I would like to thank my academic collaborators and labmates. I would like to express my appreciation to my committee members, Sanjay Shakkottai and Carlee Joe-Wong. Thanks for taking part in my PhD dissertation defense committee and providing me with valuable comments and suggestions. The discussions we had during the thesis prospectus helped in shaping my final thesis. It was due to these discussions that I was able to kick start a collaborative project with Jinhang Zuo. Thanks Jinhang for collaborating with me during the final year of my PhD. I would like to sincerely thank Shreyas Chaudhari for conducting discussion and experiments on two of the main projects of my PhD. I would also like to thank Yae Jee Cho for a fun collaborative project and teaching me a thing or two about federated learning. I am also grateful to have had friendships within my two lab groups. Thanks to Rashad and Yong for being super friendly seniors and helping me settle in at CMU. Thanks to Delphi, Can, Isabel, Yingrui, John, Mansi and Vaibhav for the research discussions and chit-chat in group meetings. Thanks to Ankur, Jianyu, Yae Jee, Tuhin, Pranay, Jiin, Divyansh, Rudy and Ahmet for meeting up regularly in weekly group meetings and discussing interesting research and playing fun math(y) games together. I would also like to thank my undergrad advisor, Sharayu Moharir, for mentoring me during my visits to India. Thanks to the Siebel Energy Institute, the Carnegie Bosch Institute, the Manufacturing Futures Initiative, the CyLab IoT Initiative, NSF (for grants CCF-1840860 and CCF-2007834), CyLab presidential fellowship and David H. Barakat and LaVerne Owen-Barakat CIT Dean's fellowship for funding my research.

Whatever I am today is because of the unconditional love and support of my family. I am thankful to all my family members for their well wishes and for always being with me. My parents, Anita Gupta and Siddharth Gupta, have always given me the complete freedom to pursue whatever I want, while always having my back. I can never thank them enough for the way they have raised me. It has always been a comforting feeling to know that I could talk to my brother, Parth, (and his wife) Shagun about any problems that I may be facing. Thanks for all your love.

People you spend the most time with shape who you are, and I am very fortunate to have myself surrounded by the most kind, caring and inspiring friends. I would like to express my gratitude towards these friends who have remained a constant presence in my life all throughout my PhD and have had numerous positive effects on my personality. First off, I would like to thank Ankur Mallick for being my roommate and being the first point of contact for any good/bad news. Thanks for listening to my countless cribs and conducting unbounded conversations on wide ranging topics such as coded computing, US politics, educational system and mental health. Thanks Ankur for the kind friendship, expanding my social network and sharing the PhD journey together. Next, I would like to thank Abhilasha Jain for maintaining such a close friendship, despite being so far way during the bulk of my PhD. I highly appreciate the fact that I could talk to Abhilasha about pretty much anything on my mind. Thanks for helping me clear my headspace, imparting wisdom, helping me live a balanced life and for making me laugh in the toughest of times. Next, I would like to thank the warm and comforting presence of Mansi Sood in my life. Thanks for checking in with me and sending in thoughtful presents from time to time. I have thoroughly enjoyed pulling Mansi's leg for sending Pittsburgh photos in the middle of a work day.

I have been fortunate to have a lot of friends in CIC, where I worked for almost all of my PhD (barring the 1 year COVID19 lockdown). These people added a lot of joy to my typical weekday by taking a lot of banter filled breaks. I would like to thank Akshay Gadre for showing up to my desk with his chai tea latte consistently for 5 years. I appreciate the random thoughts discussed during tea time with FNU Vaibhav. Thanks Tuhinangshu for always saying 'yes' to a coffee break. Thanks Rajat for joining me on lunch breaks and empathizing with me on the series of paper rejections. While I made a lot of friends in Pittsburgh, there have been friendships which have lasted since high school/undergrad. I am grateful to my childhood friend, Shubham Anand, for supporting me with his words of encouragment ever since grade 10 and for always welcoming me in NYC whenever I needed NYC the most. Thanks to Nishant Gurunath for helping me out during my tough years of both undergraduate and graduate school. Thanks to Rishabh Rai, Parantap Singh, Akhil Shetty, Kush Motwani, Ayush Baid, Aasheesh Tandon, Shubham Nema and Niranjan Thakurdesai for keeping in touch during my grad school and the covid-19 pandemic.

I feel grateful that I was also able to pursue my non-academic interests as well during all these years and it would not have been possible without the amazing company that I have had. I would like to thank Sanghamitra and Sandeep for hosting numerous potlucks and board game nights right from the first year. Thanks Rajshekhar for playing Tennis with me in the odd years of my PhD life. I am unaware of the reason behind this pattern. Thanks Abhilasha and Akhil for filling that tennis buddy gap in 2nd and 4th year respectively. Thanks to Sandeep, Rajat, Aniruddh, Nishant, Kushal, Shiv, Yae Jee, Vaibhav, Tuhinangshu, Divyansh and Jiin for playing squash with me at different points of time. Thanks to Vaibhav for trying out skateboarding and long distance running. Thanks to Mansi and Charvi for lending me their bikes and to Ankush Das for showing me all the biking routes during the pandemic. Thanks to Vaibhav and Tuhinangshu for continuing to bike with me. Thanks to Charvi, Shreyash, Manisha, Revathy, Rohit, Devdeep, Pratik and Alankar for joining in on the social hangouts. I am also grateful to Nishant and the countless anonymous players around the globe with whom I have played FIFA to maintain sanity during the pandemic. Thanks to all the friends who have attended some form of celebrations at 2355 Eldridge street and to my neighbors for never making a noise complaint.

Lastly, I would like to thank some people who managed to inspire me without having ever met me. Thanks to Rafael Nadal, for being my childhood idol and for showing me the value of grit and perseverance. Thanks to Novak Djokovic for teaching me a lesson or two on mental strength and to Roger Federer for showing me how to be gracious. I am thankful to Premier League for providing consistent entertainment throughout my PhD and to Jurgen Klopp (and Liverpool FC) for changing me from a doubter to a believer.

Abstract

Multi-Armed bandit (MAB) framework is a widely used sequential decision making framework in which a decision-maker needs to select one of the available K actions in each round, with the objective of maximizing their long-term reward. This framework has been used in practice for several applications including web advertising, medical testing by viewing the use of different ads/treatments as the arms in the MAB problem. The user's response corresponding to these different actions generates a reward for the decision-maker. Under the classical MAB framework, it is implicitly assumed that the rewards corresponding to different actions are independent of each other. But, this may not be the case in practice as rewards corresponding to different actions (i.e., ads/drugs) are likely to be correlated. Motivated by this, we study the structured and correlated MAB problem in this thesis.

First, we study the structured MAB problem where the mean rewards corresponding to different actions are a known function of a hidden parameter θ^*, thereby imposing a structured on the mean rewards of different actions. We study this problem in the most general form by imposing no restriction on the form of mean reward functions and as a result subsume the setting of several previously studied structured bandit frameworks where the mean reward function is assumed to be of a specific form. While mean rewards of different actions may be related to one another in the structured bandit setup, the reward realizations may not necessarily be correlated.

Motivated by this, we propose a novel correlated MAB framework which explicitly captures the correlation in reward across different actions. For both these frameworks, we design novel algorithms that allow us to extend any classical bandit algorithm to the structured and correlated bandit settings. Through rigorous analysis, we show that our proposed algorithms sample certain sub-optimal actions, termed as non-competitive actions, only $O(1)$ times as opposed to the typical $O(\log T)$ samples required by classical algorithms such as Upper Confidence Bound (UCB), Thompson sampling. These significant theoretical performance gains are reflected in our experiments performed on real-world recommendation system datasets such as Movielens, Goodreads. For our proposed correlated bandit framework, we also design best-arm identification algorithms where the task is to identify the best action in as few samples as possible. We demonstrate the achieved performance gains theoretically through sample complexity analysis and empirically through experiments on recommendation system datasets.

To further demonstrate the utility of our proposed correlated bandit framework, we show how the framework can be employed to solve online resource allocation problems, which frequently arise in tasks such as power allocation in wireless systems, financial optimization and multi-server scheduling. This is done by extending our correlated bandit framework and algorithms to the setting of online resource allocation. The performance gains are demonstrated theoretically through regret analysis and empirically through synthetic experiments on the task of online power allocation in wireless systems, job scheduling in multi-server systems and channel assignment under the ALOHA protocol.

Contents

List of Tables

List of Figures

Chapter 1

Overview

Classical Multi-armed Bandit. The *multi-armed bandit* (MAB) problem falls under the class of sequential decision making problems. In the classical multi-armed bandit problem, there are K arms, with each arm having an *unknown* reward distribution. At each round t, we need to decide an arm $k_t \in \mathcal{K}$ and we receive a random reward R_{k_t} drawn from the reward distribution of arm k_t. The goal in the classical multi-armed bandit is to maximize the *long-term* cumulative reward. In order to maximize cumulative reward, it is important to balance the *exploration-exploitation* trade-off, i.e., pulling each arm enough number of times to identify the one with the highest mean reward, while trying to make sure that the arm with the highest mean reward is played as many times as possible. This problem has been well studied starting with the work of Lai and Robbins [1] that proposed the upper confidence bound (UCB) arm-selection algorithm and studied its fundamental limits in terms of bounds on *regret*, which is defined as the difference between the cumulative reward of a genie policy that knows that best-arm apriori and the cumulative reward attained by a bandit algorithm. Subsequently, several other algorithms including Thompson Sampling (TS) [2] and KL-UCB [3], have been proposed for this setting. The generality of the classical multi-armed bandit model allows it to be useful in numerous applications. For example, MAB algorithms are useful in medical diagnosis [4], where the arms correspond to the different treatment mechanisms/drugs and are widely used for the problem of ad optimization [5] by viewing different version of ads as the arms in the MAB problem. The MAB framework is also useful in system testing [6], scheduling in computing systems [7, 8, 9], and web optimization [10, 5].

To see the utility, consider an example where a company needs to run a display advertising campaign for one of their products, and their creative team have designed several different versions that can be displayed (See Figure 1.1. It is expected that the user engagement (in terms of click probability and time spent looking at the ad) depends the version of the ad that is displayed. In order to maximize the total user

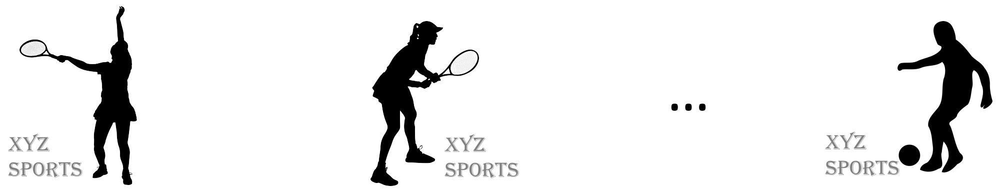


Figure 1.1: By viewing different designs as arms in the bandit problem, the problem of displaying ad designs on a web platform can be viewed as a multi-armed bandit problem. At each round, one of the ads is displayed to the user and reward is observed in terms of the user engagement (i.e., clicks or the time spent viewing the design).

engagement over the course of the ad campaign, multi-armed bandit algorithms can be used; different versions of the ad correspond to the *arms* and the reward from selecting an arm is given by the clicks or time spent looking at the ad version corresponding to that arm. The classical MAB problem implicitly assumes that rewards corresponding different arms are independent of each other. But, this may not be the case in many practical applications as the rewards corresponding to different ads/drugs are likely to be correlated. Motivated by this, we study structured and correlated multi-armed bandit problems in this thesis.

Structured Multi-Armed Bandit. We first study a fundamental variant of classical multi-armed bandits called the *structured multi-armed bandit problem*, where mean rewards of the arms are functions of a *hidden* parameter θ. That is, the expected reward $\mathbb{E}\left[R_k|\theta\right] = \mu_k(\theta)$ of arm k is a *known* function of the parameter θ that lies in a (*known*) set Θ. However, the true value of θ, denoted as θ^*, is unknown. The dependence of mean rewards on the common parameter introduces a *structure* in the MAB problem. For example, rewards observed from an arm may provide partial information about the mean rewards of other arms, making it possible to significantly lower the resulting cumulative regret as compared to the classical MAB setting. To see its connection with classical multi-armed bandits and previously studied variants let's revisit the ad optimization example presented in Figure 1.1.

<u>Personalized recommendations using Contextual and Structured bandit.</u> Although the ad-selection problem can be solved by standard MAB algorithms, there are several specialized MAB variants that are designed to give better performance. For instance, the *contextual* bandit problem [11, 12] has been studied to provide *personalized* displays of the ads to the users. Here, before making a choice at each time step (i.e., deciding which version to show to a user), we observe the *context* associated with that user (e.g., age/occupation/income features). Contextual bandit algorithms learn the mappings from the context θ to the most favored version of ad $k^*(\theta)$ in an online manner and thus are useful for personalized recommendations. Under the structured bandit framework, the context θ (age/ income/ occupational features) is *hidden* but the mean rewards for different versions of ad (arms) as a function of hidden context

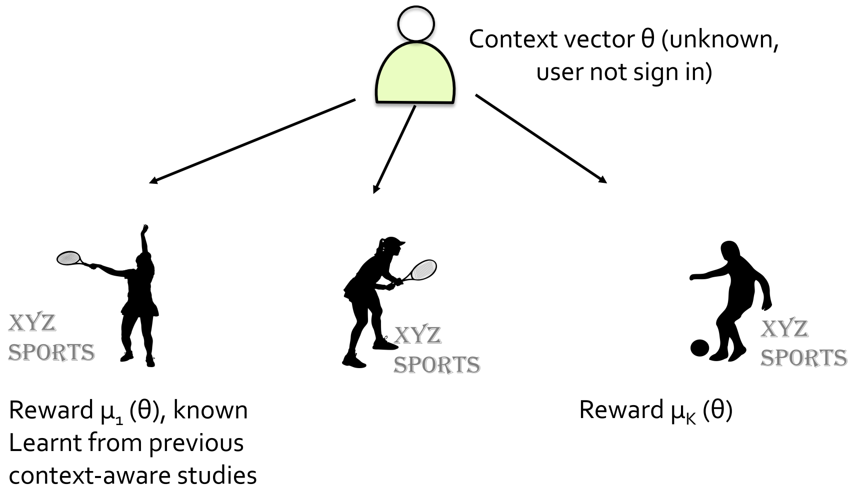


Figure 1.2: Ad-selection application of the structured bandit framework studied in this paper. The context θ (for example, the age/location/income of the user) is unknown because the user is not signed in.

θ are known. Therefore, the structured bandit model proves useful for personalized recommendation in which the context of the user is unknown, but the reward mappings $\mu_k(\theta)$ are known through surveyed data. (See Figure 1.2.)

While the structured bandit setting was studied previously by several researchers [13, 14, 15, 16, 17, 18], we make several new contributions to the field of structured bandits. The details of these can be seen in Chapter 2.

1. We study structured bandits in a general setting without making any assumptions on the nature of mean reward functions $\mu_k(\theta)$, which allows our work to subsume several previously studied settings such as the global and regional multi-armed bandits [17, 16]

2. We propose a novel algorithmic approach, which extends any classical bandit algorithm to the structured bandit setting. Previously, only extensions of UCB were known to the structured bandit setting in [13]. As algorithms such as TS, KL-UCB perform much better than UCB empirically, extending them to the structured bandits setting offers significant performance gains.

3. The proposed algorithms are easy to implement and require only the knowledge of $\mu_k(\theta)$ and not the full conditional reward distribution $\Pr(R_k|\theta)$, unlike other works [14, 15]. We show the use of our proposed structured bandit algorithms to provide personalized recommendation for users without viewing their hidden context θ. We do so by performing experiments on the Movielens dataset [19].

4. Our proposed algorithms work even when only upper and lower bounds on $\mu_k(\theta)$ are known. This is a key advantage, as the mean reward functions are often learned empirically and it is unreasonable

to obtain accurate estimate of $\mu_k(\theta)$ from the data. To the best of our knowledge, our proposed algorithms are the only known algorithms in the setting where only upper and lower bounds on $\mu_k(\theta)$.

Novel Correlated Multi-Armed Bandit model. While mean rewards of different arms are related to one another in structured bandits, they are not necessarily correlated. Motivated by this, in Chapter 3, we study a novel variant of the classical multi-armed bandit problem in which rewards corresponding to different arms are correlated to each other, i.e., the conditional reward distribution satisfies $f_{R_\ell|R_k}(r_\ell|r_k) \neq f_{R_\ell}(r_\ell)$, whence $\mathbb{E}\left[R_\ell|R_k\right] \neq \mathbb{E}\left[R_\ell\right]$. Such correlations can only be learned upon obtaining samples from different arms simultaneously, i.e., by pulling multiple arms at a time. As that is not allowed in the classical Multi-Armed Bandit formulation, we assume the knowledge of such correlations in the form of prior knowledge that might be obtained through domain expertise or from controlled surveys. One way of capturing correlations is through the knowledge of the joint reward distribution. However, if the complete joint reward distribution is known, then the best-arm is known trivially. Instead, in our work, we only assume restrictive information about correlations in the form of *pseudo-rewards* that constitute an upper bound on conditional expected rewards. This makes our model more general and suitable for practical applications.

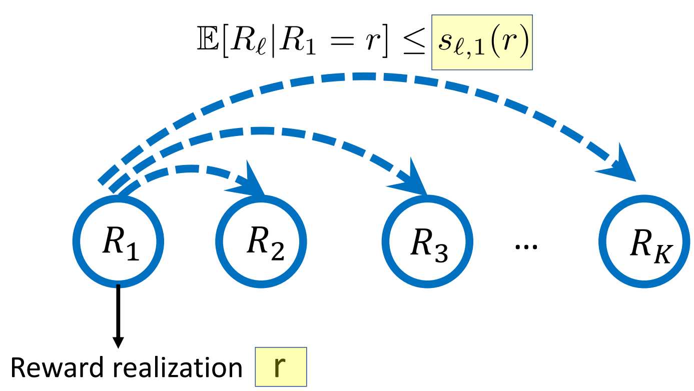


Figure 1.3: Upon observing a reward r from an arm k, pseudo-rewards $s_{\ell,k}(r)$, give us an upper bound on the conditional expectation of the reward from arm ℓ given that we observed reward r from arm k. These pseudo-rewards model the correlation in rewards corresponding to different arms.

Fig. 1.3 presents an illustration of our model, where the pseudo-rewards, denoted by $s_{\ell,k}(r)$, provide an upper bound on the reward that we could have received from arm ℓ given that pulling arm k led to a reward of r; i.e.,

$$\mathbb{E}[R_\ell|R_k = r] \leq s_{\ell,k}(r). \tag{1.1}$$

Note that pseudo-rewards are only upper bounds on the conditional expected reward and can be arbitrarily loose. In the extreme case, when no prior knowledge is available, these pseudo-rewards could be replaced by maximum possible reward and the formulation and our proposed algorithm reduce to the classical Multi-Armed Bandit framework. To view the utility of our proposed correlated multi-armed bandit model and how it contrasts with structured and contextual bandit models, let's revisit the ad recommendation example in Figure 1.1.

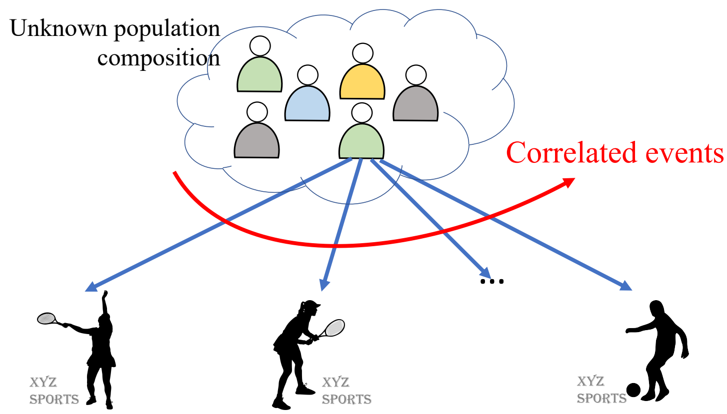


Figure 1.4: The ratings of a user corresponding to different versions of the same ad are likely to be correlated. For example, if a person likes first version, there is a good chance that they will also like the 2nd one as it also related to tennis. However, the population composition is unknown, i.e., the fraction of people liking the first/second or the last version is unknown.

Global Recommendations using Correlated-Reward Bandits. In many practical settings, the reward we get from different arms at any given step are likely to be correlated. In the ad-selection example given in Figure 1.4, a user reacting positively (by clicking, ordering, etc.) to the first version of the ad with a girl playing tennis might also be more likely to click the second version as it is also related to tennis; of course one can construct examples where there is negative correlation between click events to different ads. The model we study in this paper explicitly captures these correlations through the knowledge of pseudo-rewards $s_{\ell,k}(r)$ (See Figure 1.3). Similar to the classical MAB setting, the goal here is to display versions of the ad to maximize user engagement. In addition, unlike contextual bandits, we do not observe the context (age/occupational/income) features of the user and do not focus on providing personalized recommendation. Instead our goal is to provide global recommendations to a population whose demographics is unknown.

Regret minimization and Best-Arm Identification in Correlated MAB. In Chapter 3, we show that the knowledge of such bounds in the form of pseudo-rewards $s_{\ell,k}(r)$, even when they are not all tight, can lead to significant improvement in the cumulative reward obtained by reducing the amount of *exploration* compared to classical MAB algorithms. Our proposed MAB model and algorithm can be applied in all real-world applications of the classical Multi-Armed bandit problem, where it is possible to know pseudo-rewards from domain knowledge or through surveyed data. We also design best-arm identification algorithms in the correlated MAB setting in Chapter 4, where the goal is to find the best-arm (i.e., the arm with the highest mean reward) in as few samples as possible with confidence $1 - \delta$. Our results show that the partial knowledge of existing correlations can lead to significant reduction in the sample complexity of the designed algorithms. These theoretical results are validated through experiments for the task of providing a global recommendation to a community. We do so by studying the task of recommending best movie in the Movielens dataset [19] and the task of identifying best book for a community through the Goodreads dataset [20].

Correlated bandits for online resource allocation. To further demonstrate the utility of our proposed correlated multi-armed bandit framework, we show how the framework can be employed to solve online resource allocation problems in Chapter 5, which frequently arise in tasks such as power allocation in wireless systems, financial optimization and multi-server scheduling. In the case of financial optimization, the company needs to decide the investment of its limited financial budget across different products with the goal of maximizing its overall revenue. In such problems, the task is to distribute a limited *budget* (i.e., money, power, etc.) among available *entities* (i.e., product teams, channel etc.) with the objective of maximizing the *reward* attained (i.e., revenue, throughput, etc.). These budget allocation problem can be framed as the following optimization,

$$\underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad \sum_{k=1}^{K} f_k(a_k) \quad \text{s.t.} \quad \sum_{k=1}^{K} a_k \leq Q, \quad a_k \in \mathcal{A}, \tag{1.2}$$

with S being the budget allocation vector $(a_1, a_2, \ldots a_K)$ and Q representing the total available budget. The function $f_k(a_k)$ represents the reward attained from entity k upon allocating a budget of a_k to entity k. Moreover, in these problems, the reward obtained upon allocating a budget of a_k to entity k may be random and may depend on the underlying randomness associated with entity k. For instance, the revenue of the product may depend on the underlying unknown demand/market factors.

$$\begin{aligned} \underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \leq Q, a_k \in \mathcal{A}. \end{aligned} \tag{1.3}$$

In this scenario, the optimization problem can be solved if $\mathbb{E}\left[f_k(a_k, X_k)\right]$ is known for all (a_k, k) pairs, i.e., the mean reward of each entity k is known at all budget allocations a_k for entity k. In view of this, we refer to (1.3) as the *offline budget allocation problem*. In practice, the reward function may be unknown and the X_k's may be unknown parameters in the reward function. For instance, in the financial optimization, the reward obtained for a given budget allocation a_k for product k may depend on underlying market conditions X_k

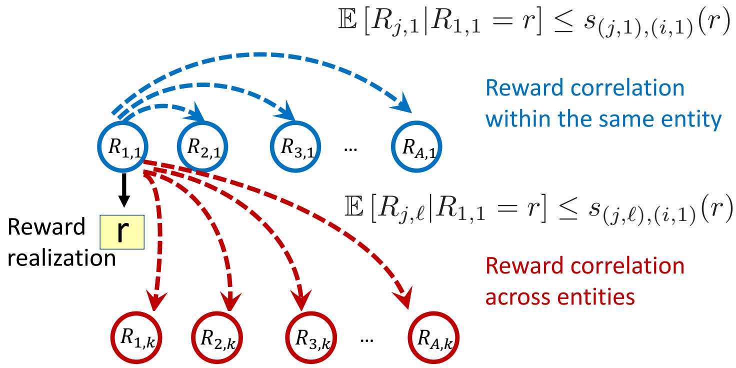


Figure 1.5: Upon observing a reward r from a base arm, pseudo-rewards $s_{(j,\ell),(i,k)}(r)$, give us an upper bound on the conditional expectation of the reward from base arm (j,ℓ) given that we observed reward r from arm (i,k). Reward received for entity k at a given budget i may provide some information on what the reward would have been if budget j were allocated to entity k, leading to correlations within entity. The rewards of different entities may also be correlated.

and one may not know the corresponding reward function f_k. As a result, $\mathbb{E}[f_k(a_k, X_k)]$ remains unknown. Motivated by this, we study the online resource allocation problem, where the goal is to sequentially decide a budget allocation $\boldsymbol{S_t} = (a_{1,t}, a_{2,t} \ldots a_{k,t}, \ldots a_{k,t})$ for each round t, so as to maximize the cumulative reward attained over a total of T rounds. To perform this allocation, there is a need to estimate $\mathbb{E}[f_k(a_k, X_k)]$ for each (a_k, k) pair and subsequently use these estimates to decide a budget allocation $\boldsymbol{S_t}$ that generates the maximum possible reward in round t. First, we view this problem as a *combinatorial* multi-armed bandit problem as done in [21]. Under the combinatorial MAB framework, the goal is to maximize the long term cumulative reward while being able to select multiple arms in each round under certain constraints. In the budget allocation problem, we can view the assignment of budget a_k to entity k as an arm to view the problem as a combinatorial bandit problem. Next, we note that the rewards received under different budget allocations (and correspondingly the arms in combinatorial bandit problem) are correlated as the reward attained for a given budget allocation i to entity k may give some information on what the reward would have been if budget j were allocated to entity ℓ as illustrated in Figure 1.5. Such correlations can be modeled through our pseudo-reward framework and subsequently used to design better budget allocation schemes as presented in Chapter 5. We theoretically show that using these correlations can lead to significant performance improvements and subsequently validate these through synthetic experiments on the task of online power allocation in wireless systems, job scheduling in multi-server systems and channel assignment under the ALOHA protocol.

Finally, we conclude the thesis by exploring potential future directions in Chapter 6. We look at potential application areas where our correlated bandit framework could be applied and discuss challenges for developing new multi-armed bandit frameworks.

Chapter 2

Structured Multi-Armed Bandits

2.1 Introduction

2.1.1 Overview

In this chapter, we study a fundamental variant of classical multi-armed bandits called the *structured multi-armed bandit problem*, where mean rewards of the arms are functions of a *hidden* parameter θ. That is, the expected reward $\mathbb{E}[R_k|\theta] = \mu_k(\theta)$ of arm k is a *known* function of the parameter θ that lies in a (*known*) set Θ. However, the true value of θ, denoted as θ^*, is unknown. The dependence of mean rewards on the common parameter introduces a *structure* in the MAB problem. For example, rewards observed from an arm may provide partial information about the mean rewards of other arms, making it possible to significantly lower the resulting cumulative regret as compared to the classical MAB setting.

Structured bandit models arise in many applications and have been studied by several authors with motivating applications including dynamic pricing (described in [17]), cellular coverage optimization (by [22]), drug dosage optimization (discussed in [16]) and system diagnosis; see Section 2.1.2 for an illustrative application of the structured MAB framework. In this chapter, we consider a *general* version of the structured MAB framework that subsumes and generalizes several previously considered settings. More importantly, we propose a novel and unified approach that would allow extending any current or future MAB algorithm (UCB, TS, KL-UCB, etc.) to the structured setting; see Sections 2.1.3 and 2.2.2 for our main contributions and a comparison of our work with related literature.

2.1.2 An Illustrative Example

For illustration purposes, consider the example of movie recommendation, where a company would like to decide which movie(s) to recommend to each user with the goal of maximizing user engagement (e.g.,

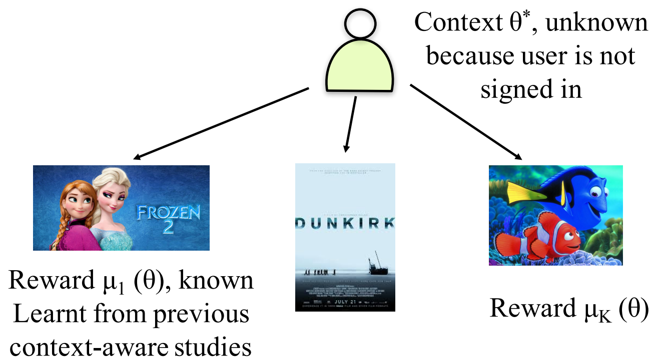


Figure 2.1: Movie recommendation application of the structured bandit framework studied in this chapter. The context θ (for example, the age of the user) is unknown because the user is not signed in. But if a user gives a high rating the first movie (Frozen) one could infer that the age θ is small, which in turn implies that the user will give a high rating to the third movie (Finding Nemo).

in terms of click probability and time spent watching, etc.). In order to achieve this, the company needs to identify the most appealing movie for the user in an online manner and this is where multi-armed bandit algorithms can be helpful. However, classical MAB algorithms are typically based on the (implicit) assumption that rewards from different arms (i.e., different movies in this context) are independent of each other. This assumption is unlikely to hold in reality since the user choices corresponding to different movies are likely to be related to each other; e.g., the engagement corresponding to different movies may depend on the age/occupation/income/taste of the user.

To address this, *contextual* bandits [23, 24] have been proposed and studied widely for personalized recommendations. There, it is assumed that before making a choice (of which movie to recommend), a *context* feature of the user is observed; the context can include personal information of the user including age, occupation, income, or previous browsing information. Contextual bandit algorithms aim to learn the mapping from context information to the most appealing arm, and can prove useful in applications involving personalized recommendations (or, advertising). However in several use cases, observing contextual features leads to privacy concerns. In addition, the contextual features may not be visible for *new* users or users who are signed in anonymously to protect their identity.

The structured bandit setting considered in this chapter (and by many others [17, 22, 16]) can be viewed as the same problem setting with contextual bandits with the following difference. Unlike contextual bandits, the context of the users are *hidden* in the structured setting, but in return it is assumed that the mappings from the contexts to *mean* rewards of arms are known a priori. It is anticipated that the mean reward mappings can be learned from paid surveys in which users participate with their consent. The proposed structured bandit framework's goal is to use this information to provide the best recommendation to an anonymous user whose context vector θ is unknown; e.g., see Figure 2.1. *Thus, our problem formulation* *is complementary to contextual bandits; in contextual bandits the context θ is known while the reward mappings $\mu_k(\theta)$ are unknown, whereas in our setting θ is unknown and the mean rewards $\mu_k(\theta)$ are known.* A detailed problem formulation discussing the assumptions and extensions of this set-up is given in Section 2.2.

2.1.3 Main Contributions and Organization

We summarize the key contributions of the chapter below. The upcoming sections will develop each of these in detail.

1. **General Setting Subsuming Previous Works**: Structured bandits have been studied in the past [17, 16, 18, 25, 26, 27] but with certain restrictions (e.g., being linear, invertible, etc.) on the mean reward mappings $\mu_k(\theta)$. We consider a general setting that puts no restrictions on the mean reward mappings. In fact, our setting subsumes recently studied models such as Global Bandits [17], Regional Bandits [16] and structured bandits with linear rewards [18]; see Section 2.2.2 for a detailed comparison with previous works.

   There are a couple of recent works [13, 14] that do consider a general structured bandit setting similar to our work – see Section 2.2.2 for details. Our approach differs from these in its flexibility to extend any classical bandit algorithm (UCB, Thompson sampling, etc.) to the structured bandit setting. In particular, using Thompson sampling [28, 29] as the underlying bandit algorithm yields a robust and empirically superior way (see Section 2.4.4) to minimize superfluous exploration. The UCB-S algorithm proposed in [13] extends the UCB algorithm to structured setting. However, the approach presented in [13] can not be extended to Thompson sampling or other classical bandit algorithms; in fact, this point was highlighted in [13] as an open question for future work. In [14], there are several assumptions in the model that are not imposed here, including the assumption that the conditional reward distributions are known and reward mappings are continuous. In addition, the main focus of [14] is the parameter regime where regret scales logarithmically with time T, while our approach demonstrates the possibility of achieving *bounded* regret.

2. **Extending any classical bandit algorithm to the structured bandit setting**: We propose a novel and unified approach that would allow extending any classical or future MAB algorithm (that is developed for the non-structured setting) to the structured bandit framework given in Figure 2.2. Put differently, we propose a *class* of structured bandit algorithms referred to as ALGORITHM-C, where "ALGORITHM" can be any classical bandit algorithm including UCB, TS, KL-UCB, etc. A detailed description of the resulting algorithms, e.g., UCB-C, TS-C, etc., are given in Section 2.3 with their steps illustrated in Figure 2.3.

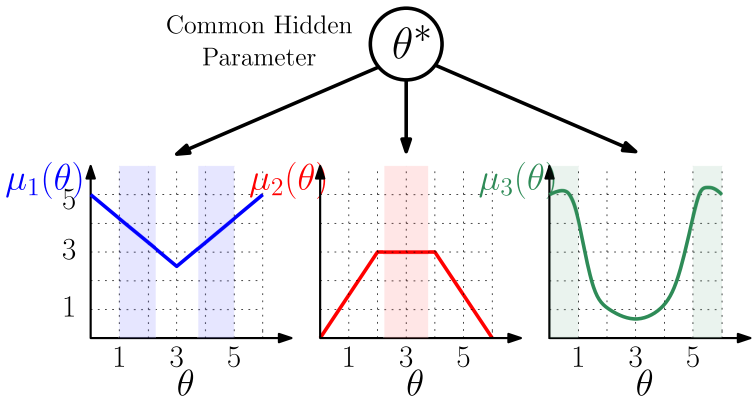


Figure 2.2: Structured bandit setup: mean rewards of different arms share a common hidden parameter. This example illustrates a 3-armed bandit problem with shaded regions indicating the values of θ for which the particular arm is optimal.

3. **Unified regret analysis**: A key benefit of our algorithms is that they pull a subset of the arms (referred to as the *non-competitive* arms) only $O(1)$ times. Intuitively, an arm is non-competitive if it can be identified as sub-optimal with high probability using only the samples from the optimal arm k^*. This is in contrast to classical MAB algorithms where all sub-optimal arms are pulled $O(\log T)$ times, where T is the total number of rounds. This is shown by analyzing the expected regret $\mathbb{E}[Reg(T)]$, which is the difference between the expected cumulative reward obtained by using the proposed algorithm and the expected cumulative reward of a genie policy that always pulls the optimal arm k^*. In particular, we provide rigorous regret analysis for UCB-C as summarized in the theorem below, and describe how our proof technique can be extended to other classical MAB algorithms.

Theorem 1 (Expected Regret Scaling)**.** *The expected regret of the UCB-C algorithm has the following scaling with respect to the number of rounds T:*

$$\mathbb{E}[Reg(T)] \leq (C(\theta^*) - 1) \cdot O(\log T) + (K - C(\theta^*))O(1) \tag{2.1}$$

where $C(\theta^*)$ is the number of competitive arms (including the optimal arm k^*) and θ^* is the true value of the hidden parameter. The remaining $K - C(\theta^*)$ arms are called non-competitive. An arm is said to be non-competitive if there exists an $\epsilon > 0$ such that $\mu_k(\theta) < \mu_{k^*}(\theta)$ for all $\theta \in \Theta^{*(\epsilon)}$, where $\Theta^{*(\epsilon)} = \{\theta \in \Theta : |\mu_{k^*}(\theta) - \mu_{k^*}(\theta^*)| \leq \epsilon\}$ (more details in Section 2.4).

The exact regret upper bound with all the constants follows from Theorem 2 and Theorem 3 in Section 2.4. Recall that for the standard MAB setting [1], the regret upper bound is $(K-1)O(\log T)$, where K is the total number of arms. Theorem 1 reveals that with our algorithms only $C(\theta^*) - 1$ out of the $K - 1$ sub-optimal arms are pulled $O(\log T)$ times. The other arms are pulled only $O(1)$ times.

4. **Reduction in the effective number of arms.** For any given set of reward functions $\mu_k(\theta)$, the number $C(\theta^*)$ of competitive arms depends on the *unknown* parameter θ^*; see Figure 2.4 in Section 2.4 for an illustration of this fact. We show that $C(\theta^*)$ can be much smaller than K in many practical cases. This is because, the reward functions (particularly that corresponding to the *optimal* arm) can provide enough information about the hidden θ^*, which in turn can help infer the sub-optimality of several other arms. More specifically, this happens when the reward functions are not flat around θ^*, that is, the pre-image set of $\{\theta \in \Theta : \mu_k(\theta) = \mu_k(\theta^*)\}$ is small. In the special case where the optimal arm k^* is *invertible* or has a unique maximum at $\mu_{k^*}(\theta^*)$, $C(\theta^*) = 1$ and our algorithms can achieve O(1) regret.

5. **Evaluation on real-world datasets**: In Section 2.4, we present extensive simulations comparing the regret of the proposed algorithm with previous methods such as GLM-UCB [25] and UCB-S [13]. We also present simulation results for the case where the hidden parameter θ is a *vector*. In Section 2.6, we perform experiments on the MovieLens dataset to demonstrate the applicability of the UCB-C and TS-C algorithms. Our experimental results show that both UCB-C and TS-C lead to significant improvement over the performance of existing bandit strategies. In particular, TS-C is shown to consistently outperform all other algorithms across a wide range of settings.

2.2 Problem Formulation

2.2.1 Structured bandits setup

Consider a multi-armed bandit setting with the set of arms $\mathcal{K} = \{1, 2, \ldots, K\}$. At each round t, the player pulls arm $k_t \in \mathcal{K}$ and observes a reward r_{k_t}. The reward r_{k_t} is a random variable with mean $\mu_{k_t}(\theta) = \mathbb{E}\left[r_{k_t}|\theta, k_t\right]$, where θ is a *fixed, but unknown parameter* which lies in a known set Θ; see Figure 2.2.

We denote the (unknown) true value of θ by θ^*. There are no restrictions on the set Θ. Although we focus on scalar θ in this chapter for brevity, the proposed algorithms and regret analysis can be generalized to the case where we have a hidden parameter *vector* $\vec{\theta} = [\theta_1, \theta_2, \ldots \theta_m]$. In Section 2.4, we present simulation results for the case of a parameter vector \`. The mean reward functions $\mu_k(\theta) = \mathbb{E}[R_k|\theta]$ for $k \in \mathcal{K}$ can be arbitrary functions of θ with no linearity or continuity constraints imposed. While $\mu_k(\theta)$ are known to the player, the conditional distribution of rewards, i.e., $p(r_k|\theta)$ is not known.

We assume that the rewards R_k are sub-Gaussian with variance proxy σ^2, i.e., $\mathbb{E}\left[\exp\left(s\left(R_k - \mathbb{E}\left[R_k\right]\right)\right)\right] \leq \exp\left(\frac{\sigma^2 s^2}{2}\right)$ $\forall s \in \mathbb{R}$, and σ is known to the player. Both assumptions are common in the multi-armed bandit literature [30, 31, 17, 13]. In particular, the sub-Gaussianity of rewards enables us to apply Hoeffding's inequality, which is essential for the analysis of regret (defined below).

The objective of the player is to select arm k_t in round t so as to maximize her cumulative reward $\sum_{t=1}^{T} r_{k_t}$ after T rounds. If the player had known the hidden θ^*, then she would always pull arm $k^* = \arg\max_{k \in \mathcal{K}} \mu_k(\theta^*)$ that yields the highest mean reward at $\theta = \theta^*$. We refer to k^* as the *optimal* arm. Maximizing the cumulative reward is equivalent to minimizing the *cumulative regret*, which is defined as

$$Reg(T) \triangleq \sum_{t=1}^{T} \left(\mu_{k^*}(\theta^*) - \mu_{k_t}(\theta^*)\right) = \sum_{k \neq k^*} n_k(T)\Delta_k,$$

where $n_k(T)$ is the number of times arm k is pulled in T slots and $\Delta_k \triangleq \mu_{k^*}(\theta^*) - \mu_k(\theta^*)$ is the *sub-optimality gap* of arm k. Minimizing the cumulative regret is in turn equivalent to minimizing $n_k(T)$, the number of times each sub-optimal arm $k \neq k^*$ is pulled.

Remark 1 (Connection to classical Multi-armed Bandits)**.** *The classical multi-armed bandit setting, which does not explicitly consider a* structure *among the mean rewards of different arms, is a special case of the proposed structured bandit framework. It corresponds to having a hidden parameter vector $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$ and the mean reward of each arm being $\mu_k = \theta_k$. In fact, our proposed algorithm described in Section 2.3 reduces to standard UCB or Thompson sampling ([1, 32]) in this special case.*

The proposed structured bandit subsumes several previously considered models where the rewards are assumed to be linear [18, 26], invertible and Hölder continuous [17, 16], etc. See Section 2.2.2 for a detailed comparison with these works.

2.2.2 Related Work

Since we do not make any assumptions on the mean reward functions $\mu_1(\theta), \mu_2(\theta), \ldots, \mu_K(\theta)$, our model subsumes several previously studied frameworks [17, 16, 18]. The similarities and differences between our model and existing works are discussed below.

Structured bandits with linear rewards [18]. In [18], the authors consider a similar model with a common hidden parameter $\theta \in \mathbb{R}$, but the mean reward functions, $\mu_k(\theta)$ are linear in θ. Under this assumption, they design a greedy policy that achieves bounded regret. Our formulation does not make linearity assumptions on the reward functions. In the special case when $\mu_k(\theta)$ are linear, our proposed algorithm also achieves bounded regret.

Global and regional bandits. The papers [17, 16] generalize this to invertible and Hölder-continuous reward functions. Instead of scalar θ, [16] considers M common unknown parameters, that is, $\theta = (\theta_1, \theta_2, \ldots, \theta_M)$. Under these assumptions, [17, 16] demonstrate that it is possible to achieve bounded regret through a greedy policy. In contrast, our work makes no invertibility or continuity assumptions on the reward functions $\mu_k(\theta)$. In the special case when $\mu_k(\theta)$ are invertible, our proposed algorithm also achieves bounded regret.

Finite-armed generalized linear bandits. In the finite-armed linear bandit setting [26], the reward function of arm x_k is $\vec{\theta}^\intercal x_k$, which is subsumed by our formulation. For the case when $\mu_k(\theta) = g(\vec{\theta}^\intercal x_k)$, our setting becomes the generalized linear bandit setting [25], for some known function g. Here, θ is the shared unknown parameter. Due to the particular form of the mean reward functions, linear bandit algorithms construct confidence ellipsoid for θ^* to make decisions. This approach cannot be easily extended to non-linear settings. Although designed for the more general non-linear setting, our algorithms demonstrate comparable regret to the GLM-UCB [25], which is designed for linear bandits.

Minimal exploration in structured bandits [14] The problem formulation in [14] is very similar to this chapter. However, [14] assumes knowledge of the conditional reward distribution $p(R_k|\theta)$ in addition to knowing the mean reward functions $\mu_k(\theta)$. It also assumes that the mappings $\theta \to \mu_k(\theta)$ are continuous. As noted before, none of these assumptions are imposed in this chapter. Another major difference of [14] with our work is that they focus on obtaining asymptotically optimal results for the regimes where regret scales as $\log(T)$. When all arms are *non-competitive* (the case where our algorithms lead to $\mathrm{O}(1)$ regret), the solution to the optimization problem described in [14, Theorem 1] becomes 0, causing the algorithm to get stuck in the exploitation phase. Put differently, the algorithm proposed in [14] is not applicable to cases where $C(\theta^*) = 1$. An important contribution of [14] is that it provides a lower bound on the regret of structured bandit algorithms. In fact, the lower bound presented in this chapter is directly based on the lower bound in [14].

Finite-armed structured bandits [13]. The work closest to ours is [13]. They consider the same model that we consider and propose the UCB-S algorithm, which is a UCB-style algorithm for this setting. Our approach allows us to extend our UCB-style algorithm to other classical bandit algorithms such as Thompson sampling. In Section 2.4 and Section 2.6, we extensively compare our proposed algorithms (both qualitatively and empirically) with the UCB-S algorithm proposed in [13]. As observed in the simulations, UCB-S is susceptible to small changes in the mean reward functions and θ^*, whereas the UCB-C algorithm that we propose here is seen to be much more robust to such variations.

Connection to information-directed sampling. Works such as [15, 33] consider a similar structured setting but assume that the conditional reward distributions $p(R_k|\theta)$ and the prior $p(\theta)$ are known, whereas we only consider that the *mean* reward functions $\mu_k(\theta) = \mathbb{E}\left[R_k|\theta\right]$ are known. The proposed algorithms are based on Thompson sampling from the posterior distribution of θ. Firstly, this approach will require a good prior over θ, and secondly, updating the posterior can be computationally expensive since it requires computing integrals over possibly high-dimensional spaces. The focus of [15] is on *worst-case* regret bounds (which are typically $O(\sqrt{T})$), where the minimum gap between two arms can scale as $O(\log T/T)$, while [33] gives gap-dependent regret bounds in regimes where the regret scales as $O(\log T)$. In addition to providing gap-dependent regret bounds, we also identify regimes where it is possible to achieve $O(1)$ regret.

Best-arm identification. In several applications such as as hyper-parameter optimization [34] and crowd-sourced ranking [35, 36, 37], the objective is to maximize the probability of identifying the arm with the highest expected reward within a given time budget of T slots instead of maximizing the cumulative reward; that is, the focus is on exploration rather than exploitation. Best-arm identification started to be studied fairly recently [38, 39, 40]. A variant of the fixed-time budget setting is the fixed-confidence setting [31, 41, 42, 43, 44], where the aim is to minimize the number of slots required to reach a δ-error in identifying the best arm. Very few best-arm identification works consider structured rewards [45, 46, 47, 48], and they mostly assume *linear* rewards. The algorithm design and analysis tools are quite different in the best-armed bandit identification problem as compared to regret minimization. Thus, extending our approach to best-arm identification would be a non-trivial future research direction.

2.3 Proposed Algorithm: Algorithm-C

For the problem formulation described in Section 2.2, we propose the following three-step algorithm called ALGORITHM-Competitive, or, in short, ALGORITHM-C. Figure 2.3 illustrates the algorithm steps for the mean reward functions shown in Figure 2.2. Step 3 can employ any classical multi-armed bandit algorithm such as UCB or Thompson Sampling (TS), which we denote by ALGORITHM. Thus, we give a unified approach to translate any classical bandit algorithm to the structured bandit setting. The formal description of ALGORITHM-C with UCB and TS as final steps is given in Algorithm 1 and Algorithm 2, respectively.

At each round $t+1$, the algorithm performs the following steps:

Step 1: Constructing a confidence set, $\hat{\Theta}_t$. From the samples observed till round t, we define the confidence set as follows:

$$\hat{\Theta}_t = \left\{ \theta : \forall k \in \mathcal{K}, \quad |\mu_k(\theta) - \hat{\mu}_k(t)| < \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}} \right\}.$$

Here, $\hat{\mu}_k(t)$ is the empirical mean of rewards obtained from the $n_k(t)$ pulls of arm k. For each arm k, we construct a confidence set of θ such that the true mean $\mu_k(\theta)$ is within an interval of size $\sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}$ from $\hat{\mu}_k(t)$. This is illustrated by the error bars along the y-axis in Figure 2.3(a), with the corresponding confidence sets shown in grey for each arm. Taking the intersection of these K confidence sets gives us $\hat{\Theta}_t$, wherein θ lies with high probability, as shown in Figure 2.3(b).

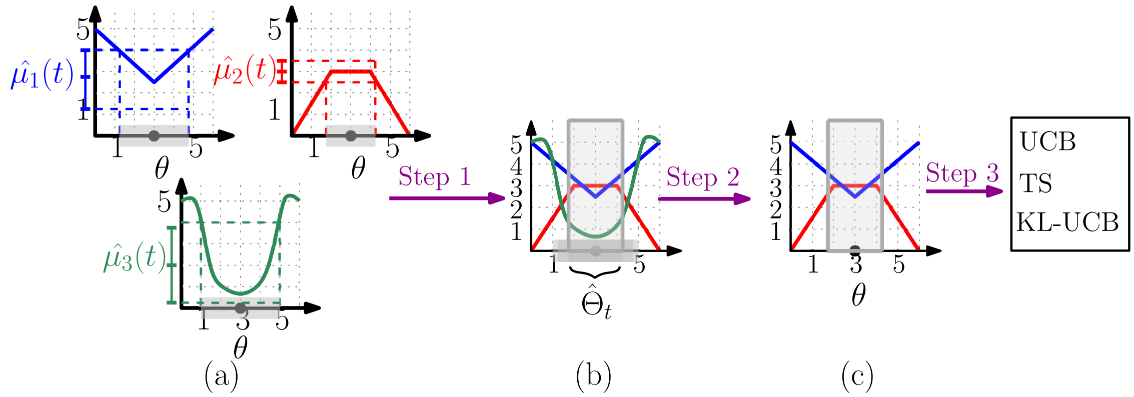


Figure 2.3: Illustration of the steps of the proposed algorithm. In step 1, for each arm k we find the set of θ such that $|\mu_k(\theta) - \hat{\mu}_k(t)| < \sqrt{2\alpha\sigma^2 \log t / n_k(t)}$ (shaded in gray in part (a)). The intersection of these sets gives the confidence set $\hat{\Theta}_t$ shown in part (b). Next, we observe that the mean reward $\mu_3(\theta)$ of Arm 3 (shown in green) cannot be optimal if the unknown parameter θ^* lies in set $\hat{\Theta}_t$. Thus, it is declared as $\hat{\Theta}_t$-non-competitive and not considered in Step 3. In step 3, we pull one of the $\hat{\Theta}_t$-competitive arms (shown in (c)) using a classical bandit algorithm such as UCB, Thompson Sampling, KL-UCB, etc.

Step 2: Finding the set $\mathcal{C}_t$ of $\hat{\Theta}_t$-Competitive arms. We let $\mathcal{C}_t$ denote the set of $\hat{\Theta}_t$-Competitive arms at round t, defined as follows.

Definition 1 ($\hat{\Theta}_t$-Competitive arm)**.** *An arm k is said to be $\hat{\Theta}_t$-Competitive if its mean reward is the highest among all arms for some $\theta \in \hat{\Theta}_t$; i.e., $\exists \theta \in \hat{\Theta}_t$ such that $\mu_k(\theta) = \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$.*

Definition 2 ($\hat{\Theta}_t$-Non-competitive arm)**.** *An arm k is said to be $\hat{\Theta}_t$-Non-competitive if it is* not *$\hat{\Theta}_t$-Competitive; i.e., if $\mu_k(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$ for all $\theta \in \hat{\Theta}_t$.*

If an arm is $\hat{\Theta}_t$-Non-competitive, then it cannot be optimal if the true parameter lies inside the confidence set $\hat{\Theta}_t$. These $\hat{\Theta}_t$-Non-competitive arms are not considered in Step 3 of the algorithm for round $t+1$. However, these arms can be $\hat{\Theta}_t$-Competitive in subsequent rounds; see also Remark 2. For example, in Figure 2.3(b), the mean reward of Arm 3 (shown in green) is strictly lower than the two other arms for all $\theta \in \hat{\Theta}_t$. Hence, this arm is declared as $\hat{\Theta}_t$-Non-competitive and only Arms 1 and 2 are included in the competitive set $\mathcal{C}_t$. In the rare case when $\hat{\Theta}_t$ is empty, we set $\mathcal{C}_t = \{1, \ldots, K\}$ and go directly to step 3 below.

Step 3: Pull an arm from the set $\mathcal{C}_t$ using a classical bandit algorithm. At round $t+1$, we choose one of the $\hat{\Theta}_t$-Competitive arms using any classical bandit Algorithm (for e.g., UCB, Thompson sampling, KL-UCB, or any algorithm to be developed for the classical bandit framework which does not explicitly model a structure connecting the rewards of different arms). Formal descriptions for UCB-C and TS-C, i.e., the structured bandit versions on UCB [1, 32] and Thompson Sampling [28] algorithms, are presented in Algorithm 1 and Algorithm 2, respectively. The ability to employ any bandit algorithm in its last step is an

Algorithm 1 UCB-Competitive (UCB-C)

1: **Input:** Reward Functions $\{\mu_1, \mu_2 \dots \mu_K\}$
2: **Initialize:** $n_k = 0$ for all $k \in \{1, 2, \dots K\}$
3: **for** each round $t+1$ **do**
4: **Confidence set construction:**

$$\hat{\Theta}_t = \left\{\theta : \forall k \in \mathcal{K}, \mid \mu_k(\theta) - \hat{\phi}_k(t) \mid < \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right\}.$$

If $\hat{\Theta}_t$ is an empty set, then define $\mathcal{C}_t = \{1, \dots, K\}$ and go to step 6

5: **Define competitive set $\mathcal{C}_t$:**

$$\mathcal{C}_t = \left\{k : \mu_k(\theta) = \max_{\ell \in \mathcal{K}} \mu_\ell(\theta) \quad \text{for some } \theta \in \hat{\Theta}_t\right\}.$$

6: **UCB among competitive arms**

$$k_{t+1} = \arg\max_{k \in \mathcal{C}_t} \left(\hat{\mu}_k(t) + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right).$$

7: Update empirical mean $\hat{\mu}_k(t+1)$ and $n_k(t+1)$ for arm k_{t+1}.
8: **end for**

Algorithm 2 Competitive Thompson Samp. (TS-C)

1: Steps 1 to 5 as in Algorithm 1
2: **Apply Thompson sampling on $\mathcal{C}_t$:**
for $k \in \mathcal{C}_t$ **do**
Sample $S_{k,t} \sim \mathcal{N}\left(\hat{\mu}_k(t), \frac{\beta\sigma^2}{n_k(t)}\right)$.
end for
$k_{t+1} = \arg\max_{k \in \mathcal{C}_t} S_{k,t}$
3: Update empirical mean, $\hat{\mu}_k$ and n_k for arm k_{t+1}.

important advantage of our algorithm. In particular, Thompson sampling has attracted a lot of attention [28, 49, 2, 50] due to its superior empirical performance. Extending it to the structured bandit setting results in significant regret improvement over previously proposed structured bandit algorithms.

Remark 2 (Connection to successive elimination algorithms for best-arm identification)**.** *Note that the empirically competitive set is updated at every round t. Thus, an arm that is empirically non-competitive at some round τ can be empirically competitive in subsequent rounds. Hence, the proposed algorithm is different from successive elimination methods used for best-arm identification [38, 39, 31, 40]. Unlike successive elimination methods, the proposed algorithm does not permanently eliminate empirically non-competitive arms but allows them to become competitive again in subsequent rounds.*

Remark 3 (Comparison with UCB-S proposed in [13])**.** *The paper [13] proposes an algorithm called UCB-S for the same structured bandit framework considered in this work. UCB-S constructs the confidence set $\hat{\Theta}_t$ in the* *same way as Step 1 described above. It then pulls the arm* $k = \arg\max_{k \in \mathcal{K}} \sup_{\theta \in \hat{\Theta}_t} \mu_k(\theta)$. *Taking the supremum of* $\mu_k(\theta)$ *over* θ *makes UCB-S sensitive to small changes in* $\mu_k(\theta)$ *and to the confidence set* $\hat{\Theta}_t$. *Our approach of identifying competitive arms is more robust, as observed in Section 2.4 and Section 2.6. Moreover, the flexibility of using Thompson Sampling in Step 3 results in a significant reduction in regret over UCB-S. As noted in [13], the approach used to design UCB-S cannot be directly generalized to Thompson Sampling and other bandit algorithms.*

Remark 4 (Computational complexity of ALGORITHM-C)**.** *The computational complexity of* ALGORITHM-C *depends on the construction of* $\hat{\Theta}_t$ *and identifying* $\hat{\Theta}_t$*-competitive arms. The algorithm is easy to implement in cases where the set* Θ *is small or in situations where the pre-image of mean reward functions* $\mu_k(\theta)$ *can be easily computed. For our simulations and experiments, we discretize the set* Θ *wherever* Θ *is uncountable.*

2.4 Regret Analysis and Insights

In this section, we evaluate the performance of the UCB-C algorithm through a finite-time analysis of the expected cumulative regret defined as

$$\mathbb{E}\left[Reg(T)\right] = \sum_{k=1}^{K} \mathbb{E}\left[n_k(T)\right] \Delta_k, \tag{2.2}$$

where $\Delta_k = \mu_{k^*}(\theta^*) - \mu_k(\theta^*)$ and $n_k(T)$ is the number of times arm k is pulled in a total of T time steps. To analyze the expected regret, we need to determine $\mathbb{E}\left[n_k(T)\right]$ for each sub-optimal arm $k \neq k^*$. We derive $\mathbb{E}\left[n_k(T)\right]$ separately for competitive and non-competitive arms. Our proof presents a novel technique to show that each non-competitive arm is pulled only $\mathrm{O}(1)$ times; i.e., our algorithms stop pulling non-competitive arms after some finite time. To establish the fact that competitive arms are pulled $\mathrm{O}(\log T)$ times each, we prove that the proposed algorithm effectively reduces a K-armed bandit problem to a $C(\theta^*)$-armed bandit problem, allowing us to extend the regret analysis of the underlying classical multi-armed bandit algorithm (UCB, Thompson Sampling, etc.)

2.4.1 Competitive and Non-competitive Arms

In Section 2.3, we defined the notion of competitiveness of arms with respect to the confidence set $\hat{\Theta}_t$ at a fixed round t. For our regret analysis, we need asymptotic notions of competitiveness of arms, which are given below.

Definition 3 (Non-competitive and Competitive Arms)**.** *For any* $\epsilon > 0$*, let*

$$\Theta^{*(\epsilon)} = \{\theta : |\mu_{k^*}(\theta^*) - \mu_{k^*}(\theta)| < \epsilon\}.$$

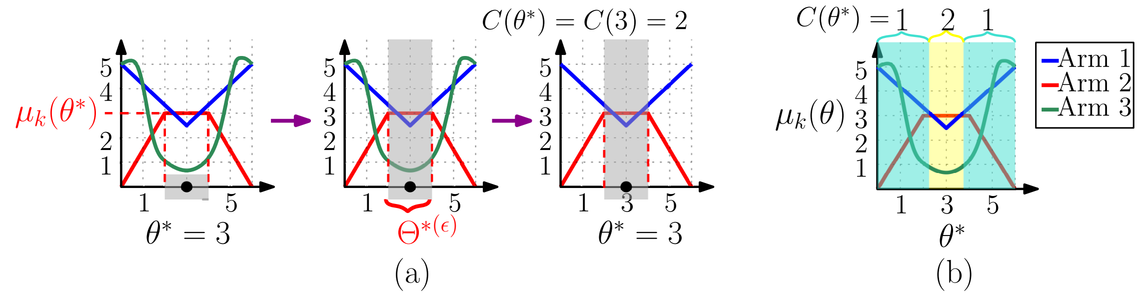


Figure 2.4: (a) Illustration of how the number of competitive arms $C(\theta^*)$ depends on the value of θ^* and the mean reward functions, when $\theta^* = 3$. To identify the competitive arms, we first find the set $\Theta^*(\epsilon) = \{\theta : |\mu_{k^*}(\theta^*) - \mu_{k^*}(\theta)| < \epsilon\}$ for small $\epsilon > 0$. Since Arm 3 (shown in green) is sub-optimal for all $\theta \in \Theta^*(\epsilon)$ it is non-competitive. As a result, $C(\theta^*) = C(3) = 2$. (b) The number of competitive arms depend on the value of θ^*. The grey region illustrates range of θ^* where $C(\theta^*) = 1$ and the yellow region indicates the range of values for which $C(\theta^*) = 2$.

An arm k is said to be non-competitive if there exists an $\epsilon > 0$ such that k is not the optimal arm for any $\theta \in \Theta^{*(\epsilon)}$; i.e., if $\mu_k(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$ for all $\theta \in \Theta^{*(\epsilon)}$. Otherwise, the arm is said to be competitive; i.e., if for all $\epsilon > 0$, $\exists \theta \in \Theta^{*(\epsilon)}$ such that $\mu_k(\theta) = \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$. The number of competitive arms is denoted by $C(\theta^*)$.

Since the optimal arm k^* is competitive by definition, we have

$$1 \leq C(\theta^*) \leq K.$$

We can think of $\Theta^{*(\epsilon)}$ as a confidence set for θ obtained from the samples of the best arm k^*. To intuitively understand the meaning of non-competitiveness, recall that the observed rewards $\hat{\mu}_k(t)$ from the arms help infer that θ^* lies in the confidence set $\hat{\Theta}_t$ with high probability. The observed reward $\hat{\mu}_{k^*}(t)$ of arm k^* will dominate the construction of the confidence set $\hat{\Theta}_t$ because a good multi-armed bandit strategy pulls the optimal arm $O(t)$ times, while other arms are pulled at most $O(\log t)$ times. Thus, for any $\epsilon > 0$, we expect the confidence set $\hat{\Theta}_t$ to converge to $\Theta^{*(\epsilon)}$ as the number $n_{k^*}(t)$ of pulls for the optimal arm gets larger. As a result, if a sub-optimal arm k is non-competitive as per the definition above, i.e., $\mu_k(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$ for all $\theta \in \Theta^{*(\epsilon)}$, then the proposed algorithm will identify k as $\hat{\Theta}_t$-non-competitive (and thus not pull it) with increasing probability at every round t. In fact, our regret analysis shows that the likelihood of a non-competitive arm being pulled at time t decays as $t^{-1-\gamma}$ for some $\gamma > 0$, leading to such arms being pulled only finitely many times.

We note that the number of competitive $C(\theta^*)$ arms is a function of the unknown parameter θ^* and the mean reward functions $\mu_k(\theta)$. Figure 2.4(a) illustrates how $C(\theta^*)$ is determined for the set of reward functions in Figure 2.2 and when $\theta^* = 3$. If $\theta^* = 3$, arm 2 (shown in red) is optimal. The corresponding confidence set $\Theta^{*(\epsilon)} = [2 - \frac{2\epsilon}{3}, 4 + \frac{2\epsilon}{3}]$ is a slightly expanded version of the range of θ corresponding to the flat part of the reward function around θ^*. Arm 3 (shown in green) has sub-optimal mean reward $\mu_3(\theta)$

for all $\theta \in \Theta^{*(\epsilon)}$, and thus it is non-competitive. On the other hand, Arm 1 (shown in blue) is competitive. Therefore, the number of competitive arms $C(\theta^*) = 2$ when $\theta^* = 3$. Figure 2.4(b) shows how $C(\theta^*)$ changes with the value of θ^*. When θ^* is outside of $[2, 4]$, i.e., the the flat portion of Arm 2, $\Theta^{*(\epsilon)}$ is a much smaller set and it is possible to show that both Arms 1 and 3 are non-competitive. Therefore, the number of competitive arms $C(\theta^*) = 1$ when θ^* is outside $[2, 4]$.

2.4.2 Upper Bounds on Regret

Definition 4 (Degree of Non-competitiveness, ϵ_k)**.** *The degree of non-competitiveness ϵ_k of a non-competitive arm k is the largest ϵ for which $\mu_k(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell(\theta)$ for all $\theta \in \Theta^{*(\epsilon)}$, where $\Theta^{*(\epsilon)} = \{\theta : |\mu_{k^*}(\theta^*) - \mu_{k^*}(\theta)| < \epsilon\}$. In other words, ϵ_k is the largest ϵ for which arm k is $\Theta^{*(\epsilon)}$-non-competitive.*

Our first result shows that the expected pulls for non-competitive arms are bounded with respect to time T. Arms with a larger degree of non-competitiveness ϵ_k are pulled fewer times.

Theorem 2 (Expected pulls of each of the $K - C(\theta^*)$ non-competitive Arms)**.** *If arm k is non-competitive, then the number of times it is pulled by UCB-C is upper bounded as*

$$\mathbb{E}[n_k(T)] \leq Kt_0 + \sum_{t=1}^{T} 2Kt^{1-\alpha} + K^3 \sum_{t=Kt_0}^{T} 6\left(\frac{t}{K}\right)^{2-\alpha} \tag{2.3}$$

$$= O(1) \quad \textit{for } \alpha > 3.$$

Here,

$$t_0 = \inf\left\{\tau \geq 2 : \Delta_{min}, \epsilon_k \geq 4\sqrt{\frac{K\alpha\sigma^2 \log \tau}{\tau}}\right\}; \quad \Delta_{min} = \min_{k \in \mathcal{K}} \Delta_k.$$

The $O(1)$ constant depends on the degree of competitiveness ϵ_k through t_0. If ϵ_k is large, it means that t_0 is small and hence $\mathbb{E}[n_k(T)]$ is bounded above by a small constant. The second and third terms in (2.3) sum up to a constant for $\alpha > 3, \beta > 1$.

The next result shows that expected pulls for any competitive arm is $O(\log T)$. This result holds for any sub-optimal arm, but for non-competitive arms we have a stronger upper bound (of $O(1)$) as given in Theorem 2. Regret analysis of UCB-C is presented in Section 2.9.4. In Section 2.9.3, we present a unified technique to prove results for any other Algorithm-C, going beyond UCB-C. We present the regret analysis of TS-C (with Beta prior and $K = 2$) in Section 2.9.5.

Theorem 3 (Expected pulls for each of the $C(\theta^*) - 1$ competitive sub-optimal arms)**.** *The expected number of times a competitive sub-optimal arm is pulled by UCB-C Algorithm is upper bounded as*

$$\mathbb{E}\left[n_k(T)\right] \leq 8\alpha\sigma^2 \frac{\log T}{\Delta_k^2} + \frac{2\alpha}{\alpha - 2} + \sum_{t=1}^{T} 2Kt^{1-\alpha}$$
$$= \mathrm{O}(\log T) \quad \textit{for } \alpha > 2,$$

Plugging the results of Theorem 2 and Theorem 3 in (2.2) yields the bound on the expected regret in Theorem 1. Note that in this work we consider a finite-armed setting where the number of arms K is a fixed constant that does *not* scale with T – we focus on understanding how the cumulative regret scales with T while K remains constant.

2.4.3 Proof Sketch

We now present the proof sketch for Theorem 2. The detail proof is given in the full proofs. For UCB-C, the proof can be divided into three steps presented below. The analysis is unique to our work and allows us to prove that the UCB-C algorithm pulls the non-competitive arms only $\mathrm{O}(1)$ times. The key strength of our approach is that the analysis can be easily extended to any ALGORITHM-C.

i) The probability of arm k^* being $\hat{\Theta}_t$-Non-Competitive is small. Observe that $\theta^* \in \hat{\Theta}_t$ implies that k^* is $\hat{\Theta}_t$-competitive. Let $E_1(t)$ denote the event that the optimal arm k^* is $\hat{\Theta}_t$-non-competitive at round t. As we obtain more and more samples, the probability of θ^* lying outside $\hat{\Theta}_t$ decreases with t. Using this, we show that

$$\Pr(E_1(t)) \leq 2Kt^{1-\alpha}. \tag{2.4}$$

This enable us to bound the expected number of pulls of a competitive arm as follows.

$$\mathbb{E}\left[n_k(t)\right] \leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=0}^{T-1} \Pr(I_k(t) > I_{k^*}(t), k_{t+1} = k). \tag{2.5}$$

In view of (2.4), the first term in (2.5) sums up to a constant for $\alpha > 2$. The term $I_k(t)$ represents the UCB Index if the last step in the algorithm is UCB, i.e., $I_k(t) = \hat{\mu}_k(t) + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}$. The analysis of second term is exactly same as that for the UCB algorithm [32]. Due to this, the upper bound on expected number of pulls of competitive arms using UCB-C has the same pre-log constants as UCB.

ii) The probability of a non-competitive arm being pulled jointly with the event that $n_{k^*}(t) > t/K$ is small. Consider the joint event that a non-competitive arm with parameter ϵ_k is pulled at round $t+1$ and the number of pulls of optimal arm till round t is at least t/K. In Lemma 4, we show that this event is unlikely. Intuitively, this is because when arm k^* is pulled sufficiently many times, the confidence interval of mean of optimal arm is unlikely to contain any value outside $[\mu_{k^*}(\theta^*) - \epsilon_k, \mu_{k^*}(\theta^*) + \epsilon_k]$. Due to this, with high probability, arm k is eliminated for round $t+1$ in step 2 of the algorithm itself. This leads to the result in Lemma 4,

$$\Pr\left(k_{t+1} = k, n_{k^*}(t) > \frac{t}{K}\right) \leq 2t^{1-\alpha} \quad \forall t > t_0 \tag{2.6}$$

iii) The probability that a sub-optimal arm is pulled more than $\frac{t}{K}$ times till round t is small. We show that

$$\Pr\left(n_k(t) > \frac{t}{K}\right) \leq 6K^2 \left(\frac{t}{K}\right)^{2-\alpha} \quad \forall t > Kt_0. \tag{2.7}$$

This result is specific to the last step used in Algorithm-C. To show (2.7) we first derive an intermediate result for UCB-C which states that

$$\Pr(k_{t+1} = k, n_k(t) \geq s) \leq (2K+4)t^{1-\alpha} \quad \text{for } s \geq \frac{t}{2K}.$$

Intuitively, if we have large number of samples of arm k, its UCB index is likely to be close to μ_k, which is unlikely to be larger than the UCB index of optimal arm k^* (which is around μ_{k^*} if n_{k^*} is also large, or even higher if n_{k^*} is *small* due to the exploration term added in UCB index).

The analysis of steps ii) and iii) are unique to our work and help us obtain the $O(1)$ regret for non-competitive arms. Using these results, we can write the expected number of pulls for a non-competitive arm as

$$\begin{aligned}\mathbb{E}\left[n_k(t)\right] \leq Kt_0 &+ \sum_{t=Kt_0}^{T-1} \Pr\left(k_{t+1} = k, n_{k^*}(t) = \max_{k \in \mathcal{K}} n_k(t)\right) \\ &+ \sum_{t=Kt_0}^{T-1} \sum_{k \in \mathcal{K}, k \neq k^*} \Pr(n_k(t) = \max_{k \in \mathcal{K}} n_k(t)).\end{aligned} \tag{2.8}$$

The second term in (2.8) is bounded through step ii) (viz. (2.6)) and the third term in (2.8) is bounded for each sub-optimal arm through step iii) (viz. (2.7)). Together, steps ii) and iii) imply that the expected number of pulls for a non-competitive arm is bounded.

2.4.4 Discussion on Regret Bounds

Reduction in the effective number of arms. The classical multi-armed bandit algorithms, which are agnostic to the structure of the problem, pull each of the $(K-1)$ sub-optimal arms $O(\log T)$ times. In contrast, our UCB-C algorithm pulls only a *subset* of the sub-optimal arms $O(\log T)$ times, with the rest (i.e., non-competitive arms) being pulled only $O(1)$ times. More precisely, our algorithms pull each of the $C(\theta^*) - 1 \leq K - 1$ arms that are competitive but sub-optimal $O(\log T)$ times. It is important to note that the upper bound on the pulls of these competitive arms by UCB-C has the same pre-log constants

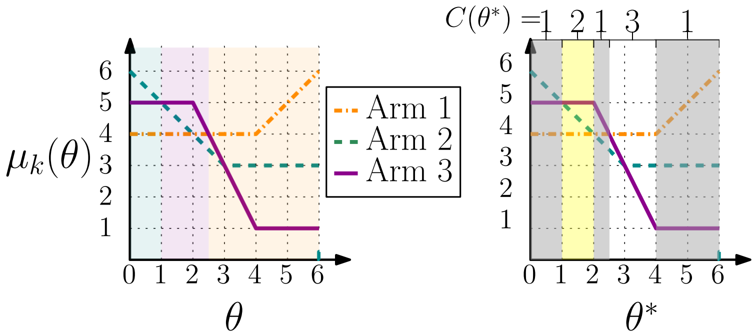


Figure 2.5: (left) Arm 2 is optimal for $\theta^* \in [0, 1]$, Arm 3 is optimal for $\theta^* \in [1, 2.5]$ and Arm 1 is optimal for $\theta^* \in [2.5, 6]$, (right) the number of competitive arms for different ranges of θ shaded in grey ($C(\theta) = 1$), yellow ($C(\theta) = 2$) and white ($C(\theta) = 3$).

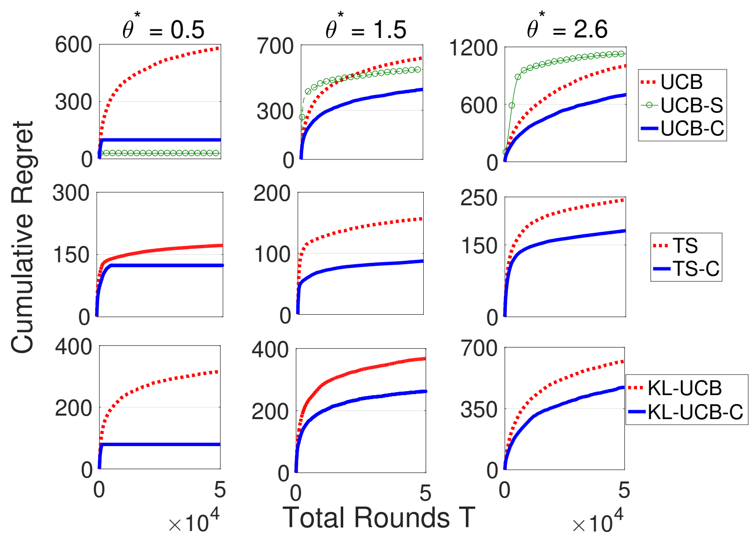


Figure 2.6: Cumulative regret of Algorithm-C vs. Algorithm (UCB in row 1, TS in row 2 and KL-UCB in row 3) for the setting in Figure 2.5. The number of competitive arms is $C(\theta^*) = 1$ in the first column, $C(\theta^*) = 2$ in second column and $C(\theta^*) = 3$ in third column. Unlike UCB-S which only extends UCB, our approach generalizes any classical bandit algorithm such as UCB, TS, and KL-UCB to the structured bandit setting.

with that of the UCB, as shown in Theorem 1. Consequently, the ability of UCB-C to reduce the pulls of non-competitive arms from $O(\log T)$ to $O(1)$ results directly in it achieving a smaller cumulative regret than its non-structured counterpart.

The number of competitive arms, i.e., $C(\theta^*)$, depends on the functions $\mu_1(\theta), \ldots, \mu_K(\theta)$ *as well as* the hidden parameter θ^*. Depending on θ^*, it is possible to have $C(\theta^*) = 1$, or $C(\theta^*) = K$, or any number in between. When $C(\theta^*) = 1$, all sub-optimal arms are non-competitive due to which our proposed algorithms achieve $O(1)$ regret. What makes our algorithms appealing is the fact that even though they do not explicitly try to predict the set (or, the number) of competitive arms, they *stop* pulling any non-competitive arm after finitely many steps.

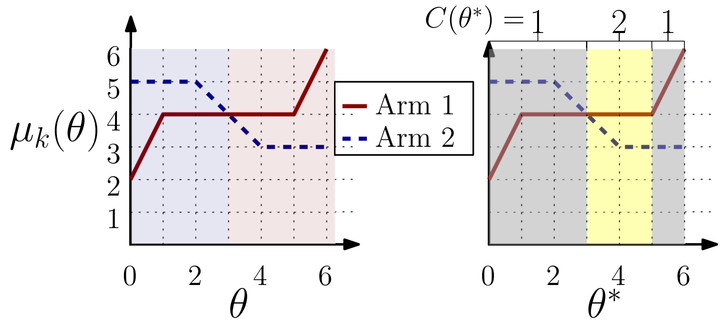


Figure 2.7: Arm 2 is optimal for $\theta^* \in [0,3]$ and Arm 1 is optimal for $\theta^* \in [3,5]$. For $\theta \in [0,3] \cup [5,6]$, $C(\theta) = 1$ and $C(\theta) = 2$ for $\theta \in [3,5]$.

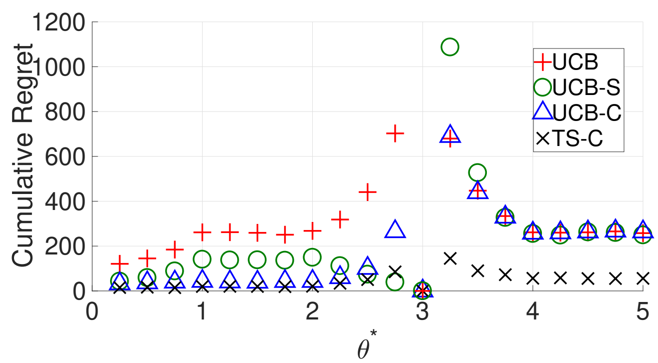


Figure 2.8: Cumulative regret of UCB,UCB-S,UCB-C and TS-C versus θ^* for the example in Figure 2.7 over 50000 runs. UCB-S is sensitive to the value of θ^* and the reward functions as it is seen to achieve a small regret for $\theta^* = 2.75$, but obtains a worse regret than UCB for $\theta^* = 3.25$.

Empirical performance of Algorithm-C. In Figure 2.6 we compare the regret of ALGORITHM-C against the regret of ALGORITHM (UCB/TS/KL-UCB). We plot the cumulative regret attained under ALGORITHM-C vs. ALGORITHM of the example shown in Figure 2.5 for three different values of θ^* : 0.5, 1.5 and 2.6. Refer to Figure 2.5 to see that $C = 1$, 2 and 3 for $\theta^* = 0.5$,1.5 and 2.6, respectively. Due to this, we see that ALGORITHM-C achieves bounded regret for $\theta^* = 0.5$, and reduced regret relative to ALGORITHM for $\theta^* = 1.5$ as only one arm is pulled $O(\log T)$ times. For $\theta^* = 2.6$, even though $C = 3$ (i.e., all arms are competitive), ALGORITHM-C achieves empirically smaller regret than ALGORITHM. We also see the advantage of using TS-C and KL-UCB-C over UCB-C in Figure 2.6 as Thompson Sampling and KL-UCB are known to outperform UCB empirically. For all the simulations, we set $\alpha = 3, \beta = 1$. Rewards are drawn from the distribution $\mathcal{N}(\mu_k(\theta^*), 4)$, i.e, $\sigma = 2$. We average the regret over 100 experiments. For a given experiment, all algorithms use the same reward realizations.

Performance comparison with UCB-S. In the first row of Figure 2.6, we also plot the performance of the UCB-S algorithm proposed in [13], alongside UCB and UCB-C. The UCB-S algorithm constructs the confidence set $\hat{\Theta}_t$ just like UCB-C, and then in the next step selects the arm $k_{t+1} = \arg\max_{k \in \mathcal{K}} \sup_{\theta \in \hat{\Theta}_t} \mu_k(\theta)$.

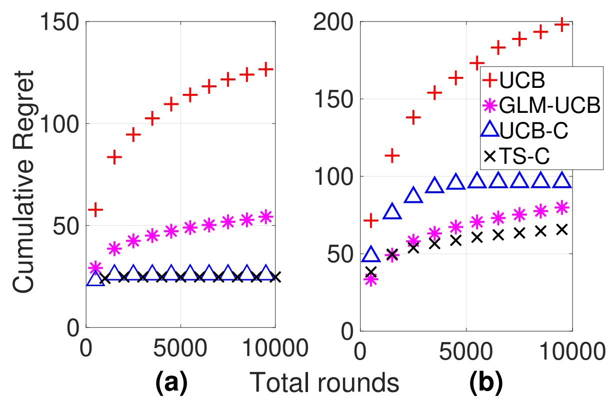


Figure 2.9: Cumulative regret of UCB, GLM-UCB, UCB-C and TS-C in the linear bandit setting, with $x_1 = (2, 1), x_2 = (1, 1.5)$ and $x_3 = (3, -1)$. Mean rewards are $(\vec{\theta}^*)^\intercal x_k$, with $\theta^* = (0.9, 0.9)$ in (a) and $\theta^* = (0.5, 0.5)$ in (b). While UCB-C and TS-C are designed for a much broader class of problems, they show competitive performance relative to GLM-UCB, which is a specialized algorithm for the linear bandit setting.

Informally, it finds the maximum possible mean reward $\mu_k(\theta)$ over $\theta \in \hat{\Theta}$ for each arm k. As a result, UCB-S tends to favor pulling arms that have the largest mean reward for $\theta \in \Theta^{*(\epsilon)}$. This bias renders the performance of UCB-S to depend heavily on θ^*. When $\theta^* = 0.5$, UCB-S has the smallest regret among the three algorithms compared in Figure 2.6, but when $\theta^* = 2.6$ it gives even worse regret than UCB. A similar observation can be made in another simulation setting described below.

Figure 2.8 compares UCB, UCB-S, UCB-C and TS-C for the functions shown in Figure 2.7. We plot the cumulative regret after 50000 rounds for different values of $\theta^* \in [0, 5]$ and observe that TS-C performs the best for most θ^* values. As before, the performance of UCB-S varies significantly with θ^*. In particular, UCB-S has the smallest regret of all when $\theta^* = 2.75$, but achieves worse regret even compared to UCB when $\theta^* = 3.25$. On the other hand, our UCB-C performs better than or at least as good as UCB for all θ^*. While UCB-S also achieves the regret bound of Theorem 1, the ability to employ any ALGORITHM in the last step of ALGORITHM-C is a key advantage over UCB-S, as Thompson Sampling and KL-UCB can have significantly better empirical performance over UCB.

Comparison in linear bandit and multi-dimensional θ settings. As highlighted in Section 2.2, our problem formulation allows θ to be multi-dimensional as well. Figure 2.9 shows the performance of UCB-C and TS-C relative to GLM-UCB in a linear bandit setting. In a linear bandit setting, mean reward of arm k is $\mu_k(\theta^*) = (\theta^*)^\intercal x_k$. Here x_k is a vector associated with arm k, which is known to the player. The parameter θ^* is unknown to the player, and hence it fits in our structured bandit framework. It is important to see that while UCB-C and TS-C are designed for a much broader class of problems, they still show competitive performance relative to specialized algorithms (i.e., GLM-UCB) in the linear bandit setting (Figure 2.9).

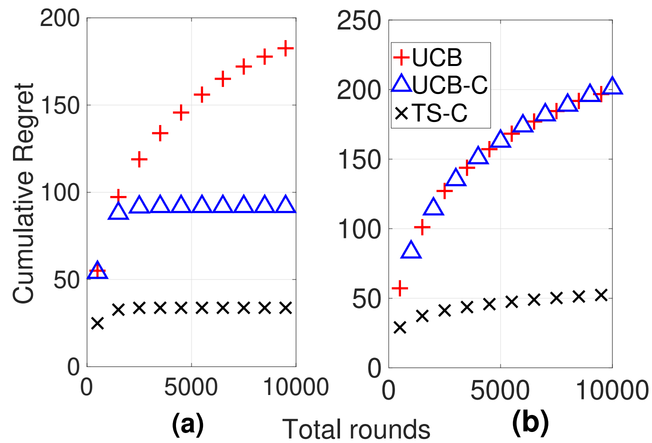


Figure 2.10: Cumulative regret for UCB,UCB-C and TS-C for the case in which $\theta \in [-1,1] \times [-1,1]$. The reward functions are $\mu_1(\vec{\theta}) = \theta_1 + \theta_2$, $\mu_2(\vec{\theta}) = \theta_1 - \theta_2$, and $\mu_3(\vec{\theta}) = \max(|\theta_1|, |\theta_2|)$. The true parameter $\vec{\theta}^*$ is $(0.9, 0.2)$ in (a) and $(-0.2, 0.1)$ in (b). The value of $C(\theta^*)$ is 1,3 in (a) and (b) respectively.

Figure 2.10 shows a setting in which θ is multi-dimensional, but the reward mappings are non-linear and hence the setting is not captured through a linear bandit framework. Our results in Figure 2.10 demonstrate that the UCB-C and TS-C algorithms work in such settings as well while providing significant improvements over UCB in certain cases.

2.4.5 When do we get bounded regret?

When $C(\theta^*) = 1$, all sub-optimal arms are pulled only O(1) times, leading to a bounded regret. Cases with $C(\theta^*) = 1$ can arise quite often in practical settings. For example, when functions are continuous or Θ is countable, this occurs when the optimal arm k^* is *invertible*, or has a unique maximum at $\mu_{k^*}(\theta^*)$, or any case where the set $\Theta^* = \{\theta : \mu_{k^*}(\theta) = \mu_{k^*}(\theta^*)\}$ is a *singleton*. These cases lead to all sub-optimal arms being non-competitive, whence UCB-C achieves bounded (i.e., $O(1)$) regret. There are more general scenarios where bounded regret is possible. To formally present such cases, we utilize a lower bound obtained in [14].

Proposition 1 (Lower bound). *For any uniformly good algorithm [1], and for any $\theta \in \Theta$, we have:*

$$\lim_{T\to\infty} \inf \frac{Reg(T)}{\log T} \geq L(\theta), \text{ where}$$

$$L(\theta) = \begin{cases} 0 & \text{if } \tilde{C}(\theta^*) = 1, \\ > 0 & \text{if } \tilde{C}(\theta^*) > 1. \end{cases}$$

An algorithm π is uniformly good if $Reg^{\pi}(T, \theta) = o(T^a)$ for all $a > 0$ and all $\theta \in \Theta$. Here $\tilde{C}(\theta^*)$ is the number of arms that are Θ^*-Competitive, with Θ^* being the set $\{\theta : \mu_{k^*}(\theta) = \mu_{k^*}(\theta^*)\}$.

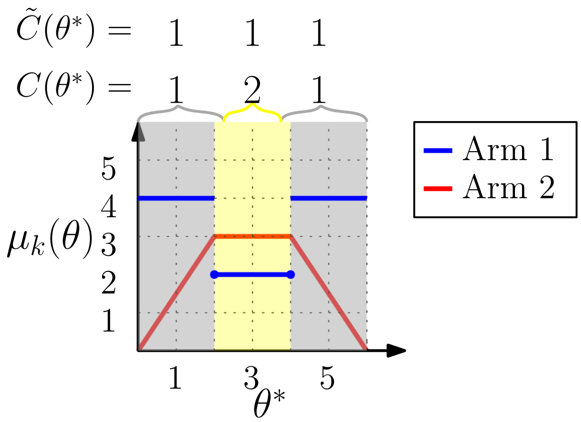


Figure 2.11: For values of $\theta \in [2,4]$ Arm 2 is Θ^*-Non-Competitive but it is still Competitive. As for any set slightly bigger than Θ, i.e., $\Theta^{*(\epsilon)}$, it is $\Theta^{*(\epsilon)}$-Competitive. Hence this is one of the corner case situations where $C(\theta)$ and $\tilde{C}(\theta^*)$ are different.

This suggests that bounded regret is possible only when $\tilde{C}(\theta) = 1$ and logarithmic regret is unavoidable in all other cases. The proof of this proposition follows from a bound derived in [14] and it is given in the full proofs section.

There is a subtle difference between $C(\theta^*)$ and $\tilde{C}(\theta^*)$. This arises in corner case situations when a Θ^*-Non-Competitive arm is competitive. Note that the set $\Theta^* = \{\theta : \mu_{k^*}(\theta^*) = \mu_{k^*}(\theta)\}$ can be interpreted as the confidence set obtained when we pull the optimal arm k^* infinitely many times. In practice, if we sample the optimal arm a *large* number of times, we can only obtain the confidence set $\Theta^{*(\epsilon)} = \{\theta : |\mu_{k^*}(\theta^*) - \mu_{k^*}(\theta)| < \epsilon\}$ for some $\epsilon > 0$. Due to this, there is a difference between $\tilde{C}(\theta^*)$ and $C(\theta^*)$. Consider the case shown in Figure 2.11 with $\theta^* = 3$. For $\theta^* = 3$, Arm 1 is optimal. In this case $\Theta^* = [2,4]$. For all values of $\theta \in \Theta^*$, $\mu_2(\theta) \leq \mu_1(\theta)$ and hence Arm 2 is Θ^*-Non-Competitive. However, for any $\epsilon > 0$, Arm 2 is $\Theta^{*(\epsilon)}$-competitive and hence Competitive. Due to this, we have $\tilde{C}(3) = 1$ and $C(3) = 2$ in this case.

If Θ is a countable set, a Θ^*-Non-Competitive arm is always $\Theta^{*(\epsilon)}$-Non-Competitive, that is, $\tilde{C}(\theta^*) = C(\theta^*)$. This occurs because one can always choose $\epsilon = \min_{\{\theta \in \Theta \setminus \Theta^*\}}\{|\mu_{k^*}(\theta^*) - \mu_{k^*}(\theta)|\}$ so that a Θ^*-Non-Competitive arm is also $\Theta^{*(\epsilon)}$-Non-Competitive. This shows that when Θ is a countable set (which is true for most practical situations where the hidden parameter θ is *discrete*), UCB-C achieves bounded regret *whenever possible*, that is, whenever $\tilde{C}(\theta^*) = 1$. While this property holds true for the case when Θ is a countable set, there can be more general cases where $C(\theta) = \tilde{C}(\theta)$. Our algorithms and regret analysis are valid regardless of Θ being countable or not.

2.5 Additional Exploration of Non-competitive But Informative Arms

The previous discussion shows that the UCB-C and TS-C algorithms enable substantial reductions in the effective number of arms and the expected cumulative regret. A strength of the proposed algorithms that can be a weakness in some cases is that they stop pulling non-competitive arms that are unlikely to be optimal after some finite number of steps. Although an arm may be non-competitive in terms of its reward yield, it can be useful in inferring the hidden parameter θ^*, which in turn may help reduce the regret incurred in subsequent steps. For instance, consider the example shown in Figure 2.12. Here, Arm 3 is sub-optimal for all values of $\theta^* \in [0, 6]$ and is never pulled by UCB-C, but it can help identify whether $\theta^* \geq 3$ or $\theta^* < 3$. Motivated by this, we propose an add-on to Algorithm-C, named as the Informative Algorithm-C (Algorithm 3), that takes the *informativeness* of arms into account and performs additional exploration of the *most informative arm* with a probability that decreases over time.

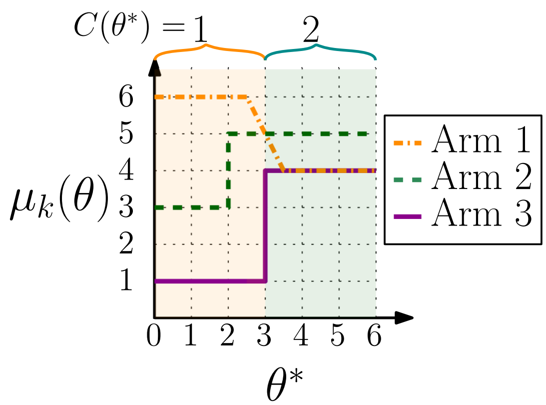


Figure 2.12: In this example, Arm 3 has $\mu_3(\theta^*) = 1$ for $\theta^* < 3$ and $\mu_3(\theta^*) = 4$ for $\theta^* \geq 3$. See that Arm 3 is sub-optimal for all values of θ^*, and hence is non-competitive for all θ^*. However, a few pulls of Arm 3 can still be useful in getting some information on whether $\theta^* \geq 3$ or $\theta^* < 3$.

2.5.1 Informativeness of an Arm

Intuitively, an arm is informative if it helps us to obtain information about the hidden parameter θ^*. At the end of round t, we know a confidence interval $\hat{\Theta}_t$ for the hidden parameter θ^*. We aim to quantify the informativeness of an arm with respect to this confidence set $\hat{\Theta}_t$. For instance, if $\hat{\Theta}_t \in [2, 4]$ in Figure 2.12, we see that the reward function of Arm 3 $\mu_3(\theta)$ has high variance and it suggests that the samples of Arm 3 could be helpful in knowing about θ^*. On the other hand, samples of Arm 2 will not be useful in identifying θ^* if $\hat{\Theta}_t = [2, 4]$. There can be several ways of defining the informativeness $I_k(\hat{\Theta}_t)$ of an arm with respect to set $\hat{\Theta}_t$. We consider the following two metrics in this work.

KL-Divergence. Assuming that θ has a uniform distribution in $\hat{\Theta}_t$, we can define the informativeness $I_k(\hat{\Theta}_t)$ of an arm as the expected KL-Divergence between two samples of arm k, i.e., $I_k(\hat{\Theta}_t) = \mathbb{E}_{\theta_1,\theta_2}\left[D_{KL}(f_{R_k}(R_k|\theta_1), f_{R_k}(R_k|\theta_2))\right]$. Our intuition here is that larger expected KL-divergence for an arm indicates that samples from it have substantially different distributions under different θ^* values, which in turn indicates that those samples will be useful in inferring the true value of θ^*. Assuming that $\Pr(R_k|\theta)$ is a Gaussian distribution with mean $\mu_k(\theta)$ and variance σ^2, then the expected KL-Divergence can be simplified as

$$\begin{aligned}
&\mathbb{E}_{\theta_1,\theta_2}\left[D_{KL}(f_{R_k}(R_k|\theta_1), f_{R_k}(R_k|\theta_2))\right] \\
&= \mathbb{E}_{\theta_1,\theta_2}\left[D_{KL}(\mathcal{N}(\mu_k(\theta_1),\sigma^2), \mathcal{N}(\mu_k(\theta_2),\sigma^2))\right] \\
&= \mathbb{E}_{\theta_1,\theta_2}\left[\frac{1}{2}(\mu_k(\theta_1) - \mu_k(\theta_2))^2\right] \\
&= \int_{\hat{\Theta}_t}\left(\mu_k(\theta) - \int_{\hat{\Theta}_t}\mu_k(\theta)U(\theta)\mathrm{d}\theta\right)^2 U(\theta)\mathrm{d}\theta = V_k(\hat{\Theta}_t),
\end{aligned}$$

where, $V_k(\hat{\Theta}_t)$ is the variance in the mean reward function $\mu_k(\theta)$, calculated when θ is uniformly distributed over the current confidence set $\hat{\Theta}_t$. Observe that the metric $I_k(\hat{\Theta}_t) = V_k(\hat{\Theta}_t)$ is easy to evaluate given the functions $\mu_k(\theta)$ and the confidence set obtained from Step 1.

Entropy. Alternatively, $\mu_k(\theta)$ can be viewed as a derived random variable of θ, where θ is uniformly distributed over the current confidence set $\hat{\Theta}_t$. The informativeness of arm k can then be defined as $I_k(\hat{\Theta}_t) = H(\mu_k(\theta))$. When $\mu_k(\theta)$ is discrete this will be the Shannon entropy $H(\mu_k(\theta)) = \sum_{\theta\in\hat{\Theta}_t} -\Pr(\mu_k(\theta))\log(\Pr(\mu_k(\theta)))$, while for continuous $\mu_k(\theta)$ it will be the *differential entropy* $H(\mu_k(\theta)) = \int_{\hat{\Theta}_t} -f_{\mu_k(\theta)}\log(f_{\mu_k(\theta)})d(\mu_k(\theta))$ where $f_{\mu_k(\theta)}$ is the probability density function of the derived random variable $\mu_k(\theta)$. Observe that differential entropy takes into account the shape as well as the range of $\mu_k(\theta)$. For example, if two reward functions are linear in θ, the one with a higher slope will have higher differential entropy, as we would desire from an informativeness metric. Evaluating the differential entropy in $\mu_k(\theta)$, i.e., , can be computationally challenging.

Other than the two metrics described above, there might be alternative (and potentially more complicated) ways of quantifying the informativeness of an arm. Another candidate would be the *information gain* metric proposed in [15], which defines informativeness in terms of identifying the best arm, rather than inferring θ^*. However, as already mentioned in [15] by the authors, information gain is computationally challenging to implement in practice outside of certain specific class of problems where prior distribution of θ is Beta or Gaussian.

Algorithm 3 Informative UCB-C

1: Steps 1 to 5 as in Algorithm 1
2: **Identify $k_{\hat{\Theta}_t}$, i.e., the most informative arm for set $\hat{\Theta}_t$:**
$k_{\hat{\Theta}_t} = \arg\max_{k \in \mathcal{K}} I_k(\hat{\Theta}_t)$
3: **Play informative arm with probability $\frac{\gamma}{t^d}$, play UCB-C otherwise:**

$$k_{t+1} = \begin{cases} k_{\hat{\Theta}_t} & \text{w.p. } \frac{\gamma}{t^d}, \\ \arg\max_{k \in \mathcal{C}_t} \left(\hat{\mu}_k(t) + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}} \right) & \text{w.p. } \quad 1 - \frac{\gamma}{t^d} \end{cases}$$

4: Update empirical mean, $\hat{\mu}_k$ and n_k for arm k_{t+1}.

2.5.2 Proposed Informative Algorithm-C and its Expected Regret

Given an informativeness metric $I_k(\hat{\Theta}_t)$, we define the most informative arm for the confidence set $\hat{\Theta}_t$ as $k_{\hat{\Theta}_t} = \arg\max_{k \in \mathcal{K}} I_k(\hat{\Theta}_t)$. At round t, Informative Algorithm-C (described in Algorithm 3) picks the most informative arm $k_{\hat{\Theta}_t}$ with probability $\frac{\gamma}{t^d}$ where $d > 1$, and otherwise uses UCB-C or TS-C to pull one of the competitive arms.Here, γ and d are hyperparameters of the Informative UCB-C algorithm. Larger γ or small d results in more exploration during the initial rounds. Setting the probability of pulling the most informative arm as $\frac{\gamma}{t^d}$ ensures that the algorithm pulls the informative arms more frequently at the beginning. This helps shrink $\hat{\Theta}_t$ faster. Setting $d > 1$ ensures that informative but non-competitive arms are only pulled $\sum_{t=1}^{\infty} \frac{\gamma}{t^d} = \mathrm{O}(1)$ times in expectation. Thus, asymptotically the algorithm will behave exactly as the underlying Algorithm-C and the regret of Informative-Algorithm-C is at most an $\mathrm{O}(1)$ constant worse than the Algorithm-C algorithm.

2.5.3 Simulation results

We implement two versions of Informative-Algorithm-C, namely Algorithm-C-KLdiv and Algorithm-C-Entropy, which use the KL-divergence and Entropy metrics respectively to identify the most informative arm $k_{\hat{\Theta}_t}$ at round t. Algorithm-C-KLdiv picks the arm with highest variance in $\hat{\Theta}_t$, i.e., $I_k(\hat{\Theta}_t) = \arg\max_k V_k(\hat{\Theta}_t)$. Algorithm-C-Entropy picks an arm whose mean reward function, $\mu_k(\theta)$, has largest shannon entropy for $\theta \in \hat{\Theta}_t$ (assuming θ to be a uniform random variable in $\hat{\Theta}_t$). As a baseline for assessing the effectiveness of the informativeness metrics, we also implement Algorithm-C-Random which selects $k_{\hat{\Theta}_t}$ by sampling one of the arms uniformly at random from the set of all arms $\mathcal{K}$ at round t.

Figure 2.13 shows the cumulative regret of the aforementioned algorithms for the reward functions shown in Figure 2.12, where the hidden parameter $\theta^* = 3.1$. Among UCB-C, UCB-C-KLdiv, UCB-C-Entropy and UCB-C-Random, UCB-C-KLdiv has the smallest cumulative regret. This is because UCB-C-KLdiv

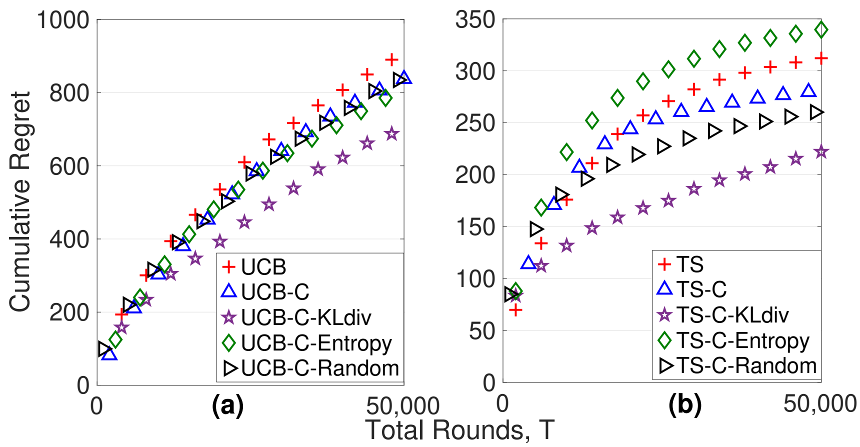


Figure 2.13: Performance comparison of ALGORITHM, ALGORITHM-C and Informative ALGORITHM-C algorithms (with parameter $\gamma = 30$, $d = 1.1$) for the example shown in Figure 2.12 with $\theta^* = 3.1$. UCB-C, TS-C do not pull Arm 3 at all, but UCB-C-KLdiv, TS-C-KLdiv pull it in the initial rounds to determine whether $\theta^* > 3$ or not. As a result, UCB-C-KLdiv and TS-C-KLdiv shrink $\hat{\Theta}_t$ faster initially and have a better empirical performance than UCB-C and TS-C, while retaining similar regret guarantees of UCB-C and TS-C respectively.

identifies Arm 3 as the most informative arm, samples of which are helpful in identifying whether $\theta^* > 3$ or $\theta^* < 3$. Hence, occasional pulls of Arm 3 lead to fast shrinkage of the set $\hat{\Theta}_t$. In contrast to UCB-C-KLdiv, UCB-C-Entropy identifies Arm 2 as the most informative arm for $\hat{\Theta} = [0, 6]$, due to which UCB-C-Entropy samples Arm 2 more often in the initial stages of the algorithm. As the information obtained from Arm 2 is relatively less useful in deciding whether $\theta^* > 3$ or not, we see that UCB-C-Entropy/TS-C-Entropy does not perform as well as UCB-C-KLdiv/TS-C-KLdiv in this scenario. UCB-C-Random picks the most informative arm by selecting an arm uniformly at random from the available set of arms. The additional exploration through random sampling is helpful, but the cumulative regret is larger than UCB-C-KLdiv as UCB-C-Random pulls Arm 3 fewer times relative to UCB-C-KLdiv. For this particular example, cumulative regret of UCB-S was 2500, whereas other UCB style algorithms achieve cumulative regret of 600-800 as shown in the Figure 2.13(a). This is due to the preference of UCB-S to pick Arm 1 in this example. We see similar trends among TS-C, TS-C-KLdiv, TS-C-Entropy and TS-C-Random. The cumulative regret is smaller for Thompson sampling variants as Thompson sampling is known to outperform UCB empirically.

We would like to highlight that the additional exploration by Informative-Algorithm-C is helpful only in cases where non-competitive arms help significantly shrink the confidence set $\hat{\Theta}_t$. For the experimental setup presented in Section 2.6 below, the reward functions are mostly flat as seen in Figure 2.16 and thus, Informative Algorithm-C does not give a significant improvement over the corresponding Algorithm-C. Therefore for clarity of the plots, we do not present experiments results for Informative-C in other settings of this work.

2.6 Experiments with Movielens data

We now show the performance of UCB-C and TS-C on a real-world dataset. We use the Movielens dataset [19] to demonstrate how UCB-C and TS-C can be deployed in practice and demonstrate their superiority over classical UCB and TS. Since movie recommendations is one of many applications of structured bandits, we do not compare with methods such as collaborative filtering that are specific to recommendation systems. Also, we do not compare with contextual bandits since the structured bandit setting has a different goal of making recommendations *without accessing a user's contextual features*.

The Movielens dataset contains a total of 1M ratings made by 6040 users for 3883 movies. There are 106 different user *types* (based on having distinct age and occupation features) and 18 different genres of movies. The users have given ratings to the movies on a scale of 1 to 5. Each movie is associated with one (and in some cases, multiple) genres. For the experiments, of the possibly multiple genres for each movie, we choose one uniformly at random. The set of users that belong to a given type is referred to as a *meta-user*; thus there are 106 different meta-users. These 106 different meta-users correspond to the different values that the hidden parameter θ can take in our setting. For example, one of the meta-users in the data-set represents college students whose age is between 18 and 24, and this corresponds to the case $\theta^* = 25$. We split the dataset into two equal parts, training and test. This split is done at random, while ensuring that the training dataset has samples from all 106 meta-users.

For a particular meta-user whose features are unknown (i.e., the true value of θ is hidden), we need to sequentially choose one of the genres (i.e., one of the arms) and recommend a movie from that genre to the user. In doing so, our goal is to maximize the *total* rating given by this user to the movies we recommended. We use the training dataset (50% of the whole data) to learn the mean reward mappings from meta-users (θ) to different genres (arms); these mappings are shown in Figure 2.16. The learned mappings indicate that the mean-reward mappings of meta-users for different genres are related to one another. For example, on average 56+ year old retired users may like documentaries more than children's movies. In our experiments, these dependencies are learned during the training. In practical settings of recommendations or advertising, these mappings can be learned from pilot surveys in which users participate with their consent.

We test the algorithm for three different meta-users, i.e., for three different values of θ^*. The movie rating samples for these meta-users are obtained from the test dataset, (the remaining 50% of the data). Figure 2.14 shows that UCB-C and TS-C achieve significantly lower regret than UCB, TS as only a few arms are pulled $O(\log T)$ times. This is because only $C(\theta^*) - 1$ of the sub-optimal arms are pulled $O(\log T)$ times by our UCB-C and TS-C algorithms. For our experimental setting, the value of C depends on θ^* (which is unknown to the algorithm). Figure 2.15 shows how $C(\theta^*)$ varies with θ^*, where it is seen that

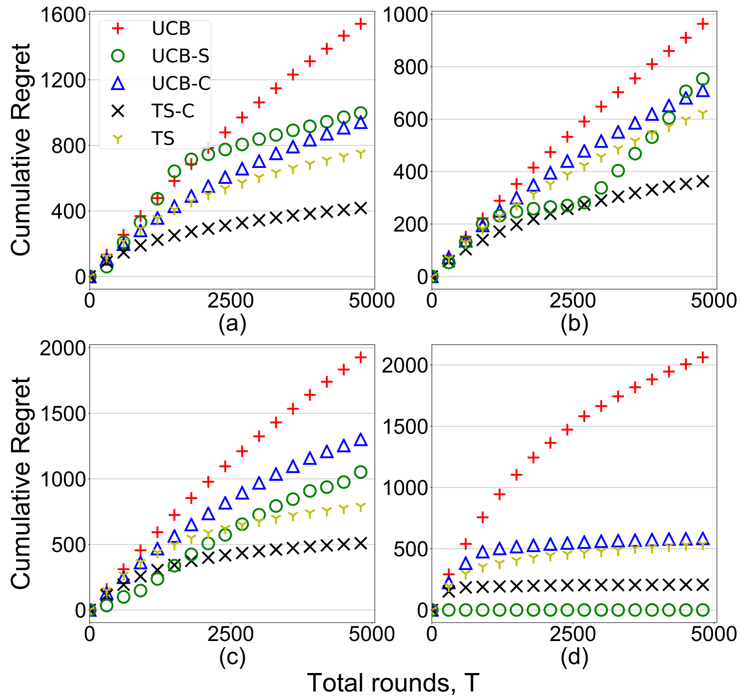


Figure 2.14: Regret plots for UCB, UCB-S, UCB-C, TS and TS-C for (a) $\theta^* = 67$ (35-44 year old grad/college students), (b) $\theta^* = 87$ (45-49 year old clerical/admin), (c) $\theta^* = 25$ (18-24 year old college students) and (d) $\theta^* = 93$ (56+ Sales and Marketing employees). The value of $C(\theta^*)$ is 6, 6, 3 and 1 for (a), (b), (c) and (d) respectively – in all cases $C(\theta^*)$ is much smaller than $K = 18$.

$C(\theta^*)$ is significantly smaller than K for all θ^*. As a result, the performance improvements observed in Figure 2.14 for our UCB-C and TS-C algorithms will apply to other θ^* values as well. There are θ^* values for which UCB-C is better than UCB-S, and vice versa. But, TS-C always outperforms UCB-C and UCB-S in our experiments. We tried Informative UCB-C in this setting as well, but the results were similar to that of UCB-C because the arms in this setting are not too informative.

2.7 Noisy mean reward functions

Under the structured bandit framework studied so far, it is assumed that the mean reward mappings $\mu_k(\theta)$ are known *exactly*. We now aim to study structured bandit formulations where we only know mean reward functions within certain bounds, e.g.,

$$\mu_k^{(l)}(\theta) \leq \mu_k(\theta) \leq \mu_k^{(u)}(\theta),$$

where $\mu_k^{(l)}(\theta), \mu_k^{(u)}(\theta)$ are known. This formulation is important for several practical applications. For example, consider the example of personalized recommendation where the goal is to recommend best

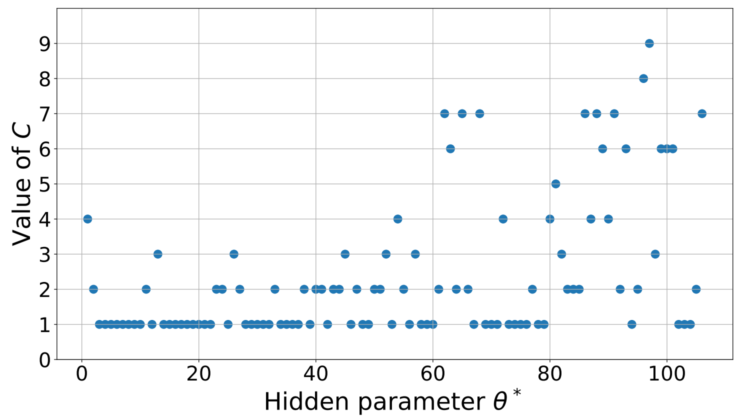


Figure 2.15: The value of $C(\theta^*)$ varies with the unknown hidden parameter θ^* (i.e., the age and occupation of the anonymous user). We see that for all θ^*, $C(\theta^*) < K$. While the total number of arms, $K = 18$, the value of $C(\theta^*)$ ranges between 1 and 9. This suggests that the ALGORITHM-C approach can lead to significant performance improvement for this problem.

movie genre to a user with feature θ^*. The mean reward mappings $\mu_k(\theta)$ are learned by taking average of ratings of movie genre k by users with context θ. The learned mapping may indicate that the mean rating given to children's movies by adults 65+ is 2.4. However, if we were to use personalized recommendation for an adult with age 65+, their mean rating for children's movies may not exactly be 2.4. As a result, the structured bandit model used with *exact* mean reward mappings may not be the best model in this case. The typical structured bandit works [14, 13, 15] work under this assumption that the mean reward mappings are known *exactly*, which makes their use restrictive in such envisioned applications.

2.7.1 Knowledge on upper and lower bounds on mean reward functions

Consider a scenario where only partial information is known about the mean reward functions $\mu_k(\theta)$. We show in this section that our proposed algorithmic approach is flexible enough to accommodate for such scenarios. The flexibility in our framework can be maintained as long as we have some upper and lower bound on mean reward functions. Suppose the mean reward functions are *noisy*, but it is known that $\mu_k^{(l)}(\theta) \leq \mu_k(\theta) \leq \mu_k^{(u)}(\theta)$. In such a case, we can alter the definition of $\hat{\Theta}_t$ as follows,

$$\hat{\Theta}_t = \hat{\Theta}_t^{(u)} \cap \hat{\Theta}_t^{(l)},$$

where $\hat{\Theta}_t^{(u)}$ and $\hat{\Theta}_t^{(l)}$ are defined as follows,

$$\hat{\Theta}_t^{(u)} = \left\{ \theta : \forall k \in \mathcal{K}, \quad \mu_k^{(l)}(\theta) \leq \hat{\mu}_k(t) + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}} \right\},$$

$$\hat{\Theta}_t^{(l)} = \left\{ \theta : \forall k \in \mathcal{K}, \quad \mu_k^{(u)}(\theta) \geq \hat{\mu}_k(t) - \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}} \right\}.$$

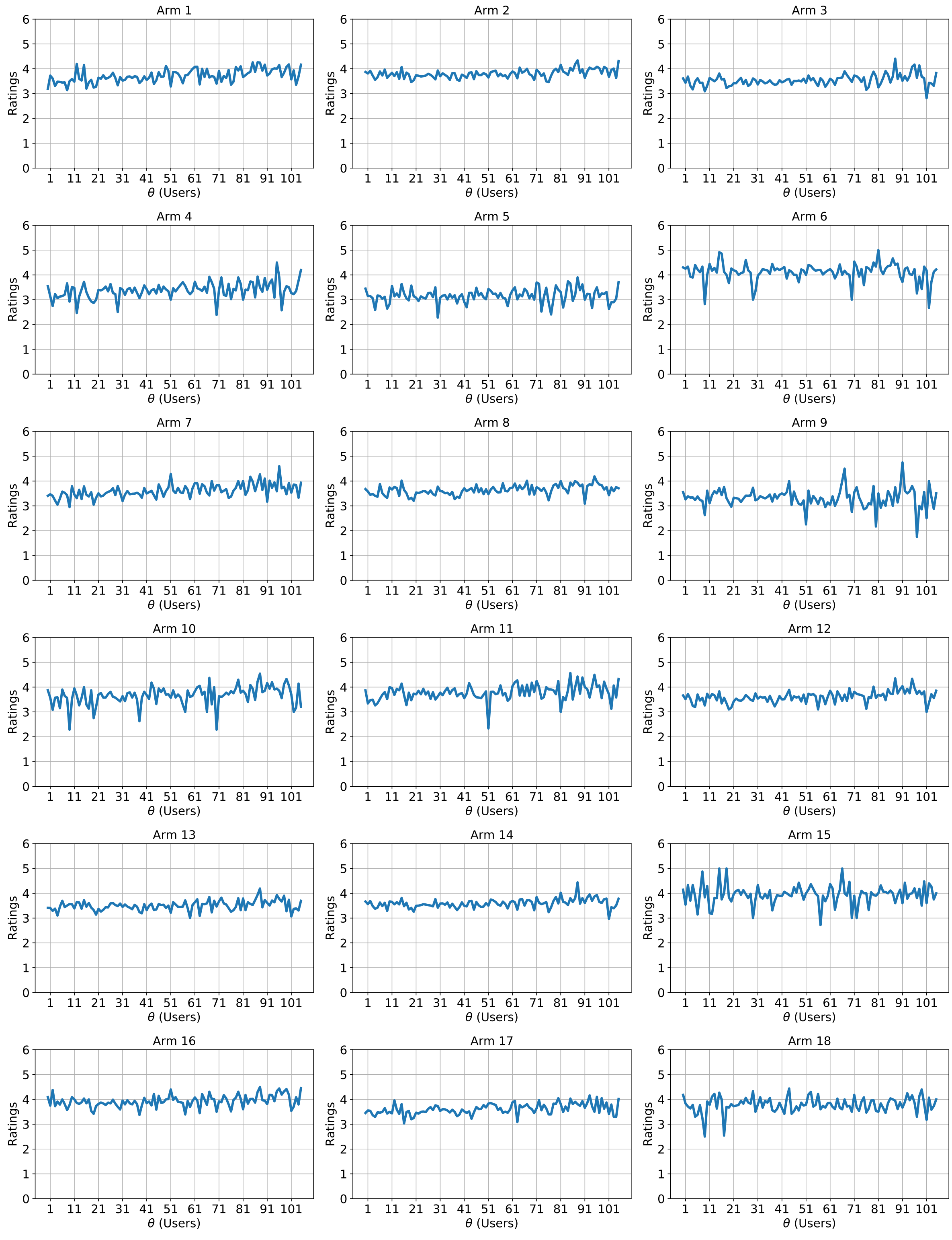


Figure 2.16: Learned reward mappings from 106 meta-users to each of the movie genres, i.e., the $\mu_k(\theta)$ in the problem setup, with θ representing different meta-users and k(arm) representing different movie genres.

Subsequently, we call an arm k to be $\hat{\Theta}_t-$Non-Competitive if $\mu_k^{(u)}(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell^{(l)}(\theta)$ for all $\theta \in \hat{\Theta}_t$. On doing so, our algorithmic approach can be extended to this framework as well. Additionally, we can re-define the notion of non-competitive and competitive arms in this setting as follows.

Definition 5 (Non-Competitive and Competitive arms in the noisy setting)**.** *For any $\epsilon > 0$, let*

$$\Theta^{*(\epsilon)} = \{\theta : \mu_{k^*}^{(l)}(\theta^*) - \epsilon < \mu_{k^*}^{(u)}(\theta), \ \ \mu_{k^*}^{(l)}(\theta) < \mu_{k^*}^{(u)}(\theta^*) + \epsilon\}.$$

An arm k is said to be non-competitive if there exists an $\epsilon > 0$ such that $\mu_k^{(u)}(\theta) < \max_{\ell \in \mathcal{K}} \mu_\ell^{(l)}(\theta)$ for all $\theta \in \Theta^{*(\epsilon)}$. Otherwise, the arm is said to be competitive; i.e., if for all $\epsilon > 0$, $\exists \theta \in \Theta^{*(\epsilon)}$ such that $\mu_k^{(u)}(\theta) \geq \max_{\ell \in \mathcal{K}} \mu_\ell^{(l)}(\theta)$. The number of competitive arms is denoted by $C(\theta^*)$.

After doing so, the analysis can be easily extended to prove the desired regret bounds in this setting. In particular, one can show that $\Pr(\theta^* \notin \hat{\Theta}_t) \leq 2Kt^{1-\alpha}$, and subsequently all other results of our work will follow through. As there is lesser information in this newer framework, one can expect the number of competitive arms to be greater than the number of competitive arms in the setting where $\mu_k^{(l)}(\theta) = \mu_k^{(u)}(\theta)$. As the algorithm, results and analysis can extend beyond the framework described in the work, it makes our proposed approach even more promising. This is an important advantage over other approaches which require the exact knowledge of $\mu_k(\theta)$, as the mean reward functions are often obtained empirically from past-data and it is unreasonable to expect *exact* knowledge on mean-reward functions from sampled data. To the best of our knowledge, our proposed algorithms are the only known algorithms in the setting where only upper and lower bounds on $\mu_k(\theta)$.

2.7.2 Knowledge on probabilistic upper and lower bounds on mean reward functions

Next, we focus on a setting where the upper and lower bounds on mean reward functions are probabilistic, i.e.,

$$\mu_k^{(l)}(\theta) \leq \mu_k(\theta) \leq \mu_k^{(u)}(\theta), \quad \text{w.p. } 1 - \eta,$$

where $\mu_k^{(l)}(\theta), \mu_k^{(u)}(\theta), \eta$ are known. In this setting, it can be shown that $O(\log T)$ regret is unavoidable. This can be seen by constructing an example where $\mu_k(\theta)$ fails to lie between $\mu_k^{(l)}(\theta)$ and $\mu_k^{(u)}(\theta)$ for all k. In such a setting, we need to adapt our algorithmic approach to account for the fact that the upper and lower bounds are probabilistic. We can do so by performing UCB-C/TS-C with probability $1 - \epsilon$ and select an arm using classical UCB/TS algorithm with probability ϵ. To further improve the empirical performance, we can tune ϵ as a function of t, based on the samples observed so far. For instance, the value of ϵ can be decreased as a function of t, if $\mu_k(\theta^*)$ is within $\mu_k^{(l)}(\theta^*)$ and $\mu_k^{(u)}(\theta^*)$ and increased otherwise.

2.8 Concluding Remarks

In this work, we studied a structured bandit problem in which the mean rewards of different arms are related through a common hidden parameter. Our problem setting makes no assumptions on mean reward functions, due to which it subsumes several previously studied frameworks [17, 16, 18]. We developed an approach that allows us to extend a classical bandit ALGORITHM to the structured bandit setting, which we refer to as ALGORITHM-C. We provide a regret analysis of UCB-C (structured bandit versions of UCB). A key insight from this analysis is that ALGORITHM-C pulls only $C(\theta^*) - 1$ of the $K - 1$ sub-optimal arms $O(\log T)$ times and all other arms, termed as *non-competitive* arms, are pulled only $O(1)$ times. Through experiments on the MOVIELENS dataset, we demonstrated that UCB-C and TS-C give significant improvements in regret as compared to previously proposed approaches. Thus, the main implication of this work is that it provides a unified approach to exploit the structured rewards to drastically reduce exploration in a principled manner.

For cases where non-competitive arms can provide information about θ that can shrink the confidence set $\hat{\Theta}_t$, we propose a variant of ALGORITHM-C called informative-ALGORITHM-C that takes the informativeness of arms into account without increasing unnecessary exploration. Linear bandit algorithms [51, 25, 26] shrink the confidence set $\hat{\Theta}_t$ in a better manner by taking advantage of the linearity of the mean reward functions to estimate θ^* as the solution to least squares problem [51]. Moreover, linearity helps them to use self-normalized concentration bound for vector valued martingale, (Theorem 1 in [51]) to construct the confidence intervals. Extending this approach to the general structured bandit setting is a non-trivial open question due to the absence of constraints on the nature of mean reward functions $\mu_k(\theta)$. The paper [14] proposes a statistical hypothesis testing method for the case of known conditional reward distributions. Generalizing it to the setting considered in this work is an open future direction. While we state our results for a scenario where mean reward functions are known, our algorithmic approach, analysis and results can also be extended to a setting where only lower and upper bounds on the mean reward function $\mu_k(\theta)$ are known. Another open direction in this field is to study the problem of structured best-arm identification where the goal is to conduct pure exploration and identify the best arm in the fewest number of rounds.

2.9 Full proofs

2.9.1 Lower bound: Proof for Proposition 1

Proof for Proposition 1: We use the following result of [14] to state Proposition 1.

Theorem 4 (Lower bound, Theorem 1 in [14].)**.** *For any uniformly good algorithm [1], and for any $\theta \in \Theta$, we have:*

$$\lim \inf_{T\to\infty} \frac{Reg(T)}{\log T} \geq L(\theta),$$

where $L(\theta)$ is the solution of the optimization problem:

$$\begin{aligned} &\min_{\eta(k)\geq 0, k\in\mathcal{K}} \sum_{k\in\mathcal{K}} \eta(k) \left(\max_{\ell\in\mathcal{K}} \mu_\ell(\theta) - \mu_k(\theta) \right) \\ &\textit{subject to} \sum_{k\in\mathcal{K}} \eta(k) D(\theta, \lambda, k) \geq 1, \forall \lambda \in \Lambda(\Theta), \\ \textit{where} \quad &\Lambda(\theta) = \{\lambda \in \Theta^* : k^* \neq \arg\max_{k\in\mathcal{K}} \mu_k(\lambda)\}. \end{aligned} \tag{2.9}$$

Here, $D(\theta, \lambda, k)$ is the KL-Divergence between distributions $f_R(R_k|\theta, k)$ and $f_R(R_k|\lambda, k)$. An algorithm, π, is uniformly good if $Reg^\pi(T, \theta) = o(T^a)$ for all $a > 0$ and all $\theta \in \Theta$.

We see that the solution to the optimization problem (2.9) is $L(\theta) = 0$ only when the set $\Lambda(\theta)$ is empty. The set $\Lambda(\theta)$ being empty corresponds to a case where all sub-optimal arms are Θ^*-non-competitive. This implies that sub-logarithmic regret is possible only when $\tilde{C}(\theta^*) = 1$, i.e there is only one Θ^*-competitive arm, which is the optimal arm, and all other arms are non-competitive. It is assumed that reward distribution of an arm k is parameterized by the mean μ_k of arm k; this ensures that if $\mu_k(\theta) = \mu_k(\lambda)$ then we have $D(\theta, \lambda, k) = 0$.

2.9.2 Results valid for any Algorithm-C

We now present results that depend only on Step 1 and Step 2 of our algorithm and hence are valid for any ALGORITHM-C, where ALGORITHM can be UCB, Thompson sampling or any other method designed for classical multi-armed bandits with independent arms.

Fact 1 (Hoeffding's inequality)**.** *Let $Z_1, Z_2, \ldots Z_T$ be i.i.d. random variables, where Z_i is σ^2 sub-gaussian with mean μ, then*

$$\Pr(|\hat{\mu} - \mu|) \geq \epsilon) \leq 2\exp\left(-\frac{\epsilon^2 T}{2\sigma^2}\right),$$

Here $\hat{\mu}$ is the empirical mean of the $Z_1, Z_2 \ldots Z_T$.

Lemma 1 (Standard result used in bandit literature (Used in Theorem 2.1 of [32])). *If $\hat{\mu}_{k,n_k(t)}$ denotes the empirical mean of arm k by pulling arm k $n_k(t)$ times through any algorithm and μ_k denotes the mean reward of arm k, then we have*

$$\Pr(|\hat{\mu}_{k,n_k(t)} - \mu_k| \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1) \leq \sum_{s=\tau_1}^{\tau_2} 2\exp\left(-\frac{s\epsilon^2}{2\sigma^2}\right)$$

Proof. Let $Z_1, Z_2, ...Z_t$ be the reward samples of arm k drawn separately. If the algorithm chooses to play arm k for m^{th} time, then it observes reward Z_m. Then the probability of observing the event $(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1)$ can be upper bounded as follows,

$$\begin{aligned}
\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) &= \Pr\left(\left(\frac{\sum_{i=1}^{n_k(t)} Z_i}{n_k(t)} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \\
&\leq \Pr\left(\left(\bigcup_{m=\tau_1}^{\tau_2}\left\{\frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right\}\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \quad (2.10) \\
&\leq \Pr\left(\bigcup_{m=\tau_1}^{\tau_2}\left\{\frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right\}\right) \\
&\leq \sum_{s=\tau_1}^{\tau_2} \Pr\left(\frac{\sum_{i=1}^{s} Z_i}{s} - \mu_k \geq \epsilon\right) \\
&\leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-\frac{s\epsilon^2}{2\sigma^2}\right). \quad (2.11)
\end{aligned}$$

□

Lemma 2. *The probability that the difference between the true mean of arm k and its empirical mean after t time slots is more than $\sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}$ is upper bounded by $2t^{1-\alpha}$, i.e.,*

$$\Pr\left(|\mu_k(\theta^*) - \hat{\phi}_k| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right) \leq 2t^{1-\alpha}.$$

Proof. See that,

$$\begin{aligned}
\Pr\left(|\mu_k(\theta^*) - \hat{\phi}_{k,n_k(t)}| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right) &\leq \sum_{m=1}^{t} \Pr\left(|\mu_k(\theta^*) - \hat{\phi}_{k,m}| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{m}}\right) \quad (2.12) \\
&\leq \sum_{m=1}^{t} 2t^{-\alpha} \quad (2.13) \\
&= 2t^{1-\alpha}.
\end{aligned}$$

We have (2.12) from union bound and is a standard trick to deal with the random variable $n_k(t)$ as it can take values from 1 to t (Lemma 1). The true mean of arm k is $\mu_k(\theta^*)$. Therefore, if $\hat{\mu}_{k,m}$ denotes the empirical mean of arm k taken over m pulls of arm k then, (2.13) follows from Fact 1 with ϵ in Fact 1 being equal to $\sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}$. □

Lemma 3. *Define $E_1(t)$ to be the event that arm k^* is Θ_t-non-competitive for the round $t+1$, then,*

$$\Pr(E_1(t)) \leq 2Kt^{1-\alpha}.$$

Proof. Observe that

$$\Pr(E_1(t)) \leq \Pr(\theta^* \notin \hat{\Theta}_t) = \Pr\left(\bigcup_{k \in \mathcal{K}} |\mu_k(\theta^*) - \hat{\mu}_{k,n_k(t)}| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right) \tag{2.14}$$

$$\leq \sum_{k=1}^{K} \Pr\left(|\mu_k(\theta^*) - \hat{\mu}_{k,n_k(t)}| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}\right) \tag{2.15}$$

$$\leq \sum_{k=1}^{K} \sum_{m=1}^{t} \Pr\left(|\mu_k(\theta^*) - \hat{\mu}_{k,m}| \geq \sqrt{\frac{2\alpha\sigma^2 \log t}{m}}\right) \tag{2.16}$$

$$\leq K \sum_{m=1}^{t} 2t^{-\alpha} \tag{2.17}$$

$$= 2Kt^{1-\alpha}.$$

In order for Arm k^* to be $\hat{\Theta}_t$-non-competitive, we need $\mu_{k^*}(\theta) < \max_{k \in \mathcal{K}} \mu_k(\theta) \quad \forall \theta \in \hat{\Theta}_t$. See that $\theta^* \in \hat{\Theta}_t$ implies that arm k^* is $\hat{\Theta}_t$-Competitive. Therefore, $\theta^* \notin \hat{\Theta}_t$ is a necessary condition for Arm k^* to be $\hat{\Theta}_t$-Non-Competitive. Due to this, we have $\Pr(E_1(t)) \leq \Pr(\theta^* \notin \hat{\Theta}_t)$ in (2.14). We are using $\hat{\mu}_{k,m}$ to denote the empirical mean of rewards from arm k obtained from it's m pulls. Here (2.14) follows from definition of confidence set and (2.15) follows from union bound. We have (2.16) from union bound and is a standard trick to deal with the random variable $n_k(t)$ as it can take values from 1 to t (Lemma 1). The inequality (2.17) follows from Hoeffding's lemma. □

Lemma 4. *Consider a suboptimal arm $k \neq k^*$, which is $\Theta^{*(\epsilon_k)}$-non-competitive. If $\epsilon_k \geq \sqrt{\frac{8\alpha\sigma^2 K \log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\Pr(k_{t+1} = k, k^* = k^{max}) \leq 2t^{1-\alpha},$$

where $k^{max} = \arg\max_{k \in \mathcal{K}} n_k(t)$.

Proof. We now bound this probability as,

$$
\begin{aligned}
&\Pr(k_{t+1} = k, k^* = k^{\max}) \\
&= \Pr(k \in \mathcal{C}_t, I_k = \max_{\ell \in \mathcal{C}} I_\ell, k^* = k^{\max}) \\
&\leq \Pr(k \in \mathcal{C}_t, k^* = k^{\max}) \\
&\leq \Pr(|\hat{\mu}_{k^*} - \mu_{k^*}(\theta^*)| > \frac{\epsilon_k}{2}, k^* = k^{\max}) && (2.18)\\
&\leq 2t \exp\left(-\frac{\epsilon^2 t}{8K\sigma^2}\right) && (2.19)\\
&\leq 2t^{1-\alpha} \quad \forall t > t_0. && (2.20)
\end{aligned}
$$

See that $|\hat{\mu}_{k^{\max}} - \mu_{k^{\max}}| < \frac{\epsilon_k}{2} \Rightarrow |\mu_{k^{\max}}(\theta) - \mu_{k^{\max}}(\theta^*)| < \epsilon$ for $\theta \in \tilde{\Theta}_t$. This holds as $\sqrt{\frac{2\alpha\sigma^2 \log t_0}{t_0}} \leq \frac{\epsilon_k}{2}$ and if $\theta \in \tilde{\Theta}_t$, then $|\mu_{k^{\max}}(\theta) - \hat{\mu}_{k^{\max}}| \leq \sqrt{\frac{2\alpha\sigma^2 \log t_0}{t_0}} \leq \frac{\epsilon_k}{2}$. Therefore in order for arm k to be Θ_t-competitive, we need at least $|\hat{\mu}_{k^*} - \mu_{k^*}(\theta^*)| > \epsilon_k/2$, which leads to (2.18) as arm k is ϵ_k non-competitive. Inequality (2.19) follows from Hoeffding's inequality. The term t before the exponent in (2.19) arises as the random variable n_{k^*} can take values from $\frac{t}{K}$ to t (Lemma 1). □

2.9.3 Unified Regret Analysis

In this section, we show a sketch of how regret analysis of ALGORITHM-C can be performed, where ALGORITHM is a bandit algorithm that works for classical multi-armed bandit problem. In later section, we provide rigorous proof for the regret analysis of the UCB-C algorithm.

Bound on expected number of pulls for Non-Competitive arms

For non-competitive arms, we show that the expected number of pulls is $O(1)$. Intuition behind the proof of the result is that non-competitive arms can be identified as sub-optimal based on *enough* pulls of the optimal arm, due to which pulling a sub-optimal arm $O(\log T)$ becomes unnecessary and it ends up being pulled only $O(1)$ times. We now provide a mathematical proof sketch of this idea:

We bound $\mathbb{E}[n_k(t)]$ as

$$
\begin{aligned}
\mathbb{E}[n_k(T)] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{k_t = k\}}\right] && (2.21)\\
&= \sum_{t=0}^{T-1} \Pr(k_{t+1} = k) \\
&= \sum_{t=1}^{Kt_0} \Pr(k_t = k) + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1} = k)
\end{aligned}
$$

$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1} = k, n_{k^*}(t) = \max_{k'} n_{k'}(t)) +$$
$$\sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \left(\Pr(n_{k'}(t) = \max_{k''} n_{k''}(t)) \times \Pr(k_{t+1} = k | n_{k'}(t) = \max_{k''} n_{k''}(t)) \right)$$
$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1} = k, n_{k^*}(t) = \max_{k'} n_{k'}(t)) + \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \Pr(n_{k'}(t) = \max_{k''} n_{k''}(t))$$
$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} 2t^{1-\alpha} + \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \Pr(n_{k'}(t) = \max_{k''} n_{k''}(t)) \tag{2.22}$$
$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} 2t^{1-\alpha} + \sum_{t=Kt_0}^{T} \sum_{k' \neq k^*} \Pr\left(n_{k'}(t) \geq \frac{t}{K}\right) \quad \forall t > t_0$$

Here (2.22) follows from plugging the result of Lemma 4.

In order to prove that ALGORITHM-C achieves bounded regret, we need to show that $\Pr\left(n_{k'}(t) \geq \frac{t}{K}\right) \leq \frac{\eta}{t^{1+\epsilon}}$, for $k' \neq k^*$, for some $\eta, \epsilon > 0$. Intuitively, this means that the probability of selecting a sub-optimal arm more than $\frac{t}{K}$ times decays with the number of rounds. This property should intuitively hold true for any good performing bandit algorithm. We prove this rigorously for UCB-C.

Bound on expected number of pulls for Competitive arms

For any suboptimal arm $k \neq k^*$,

$$\mathbb{E}\left[n_k(T)\right] \leq \sum_{t=1}^{T} \Pr(k_t = k)$$
$$= \sum_{t=1}^{T} \Pr((k_t = k, E_1(t)) \cup (E_1^c(t), k_t = k)) \tag{2.23}$$
$$\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t = k)$$
$$= \sum_{t=1}^{T} 2Kt^{1-\alpha} + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t = k) \tag{2.24}$$

Here (2.24) follows from the result of Lemma 3 with $E_1(t)$ being the event that arm k^* is Θ_t-non-competitive for the round $t+1$.

See that the event $E_1^{(c)}(t) \cap k_t = k$ corresponds to the event that a sub-optimal arm k and optimal arm k^* are both present in the competitive set, and the ALGORITHM selects the sub-optimal arm k. The analysis of this term is exactly equivalent to selecting a sub-optimal arm over optimal arm by ALGORITHM. This leads to a O$(\log T)$ bound on the expected pulls of the competitive arms as classical bandit algorithms pull each arm O$(\log T)$ times.

2.9.4 Proof for the UCB-C Algorithm

Lemma 5. *If* $\Delta_{min} \geq 4\sqrt{\frac{K\alpha\sigma^2 \log t_0}{t_0}}$ *for some constant* $t_0 > 0$*, then,*

$$\Pr(k_{t+1} = k, n_k(t) \geq s) \leq (2K+4)t^{1-\alpha} \quad \textit{for } s \geq \frac{t}{2K},$$

$\forall t > t_0$*, where* $k \neq k^*$ *is a suboptimal arm.*

Proof. The probability that arm k is pulled at step $t+1$, given it has been pulled s times can be bounded as follows:

$$\begin{aligned}
\Pr(k_{t+1} = k, n_k(t) \geq s) &= \Pr(I_k(t) = \max_{k' \in \mathcal{C}_t} I_{k'}(t), n_k(t) \geq s) \\
&\leq \Pr(E_1(t) \cup (E_1^c(t), I_k(t) > I_{k^*}(t)), n_k(t) \geq s) \\
&\leq \Pr(E_1(t), n_k(t) \geq s) + \Pr(E_1^c(t), I_k(t) > I_{k^*}(t), n_k \geq s) \quad (2.25) \\
&\leq 2Kt^{1-\alpha} + \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s\right) \quad (2.26)
\end{aligned}$$

Here, (2.25) follows from union bound and (2.26) follows from Lemma 3. We now bound the second term as,

$$\begin{aligned}
\Pr(I_k(t) > I_{k^*}(t), n_k(t) \geq s) &= \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + \\
&\quad \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s | \mu_{k^*} > I_{k^*}(t)\right) \times \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \quad (2.27) \\
&\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \quad (2.28) \\
&\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + 2t^{1-\alpha} \quad (2.29) \\
&= \Pr\left(I_k(t) > \mu_{k^*}, n_k(t) \geq s\right) + 2t^{1-\alpha} \\
&= \Pr\left(\hat{\mu}_k + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}} > \mu_{k^*}, n_k(t) \geq s\right) + 2t^{1-\alpha} \\
&= \Pr\left(\hat{\mu}_k - \mu_k(\theta^*) > \Delta_k - \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}, n_k(t) \geq s\right) + 2t^{1-\alpha} \\
&\leq 2t \exp\left(-\frac{s}{2\sigma^2}\left(\Delta_k - \sqrt{\frac{2\alpha\sigma^2 \log t}{s}}\right)^2\right) + 2t^{1-\alpha} \quad (2.30) \\
&= 2t^{1-\alpha} \exp\left(-\frac{s}{2\sigma^2}\left(\Delta_k^2 - 2\Delta_k\sqrt{\frac{2\alpha\sigma^2 \log t}{s}}\right)\right) + 2t^{1-\alpha} \\
&= 4t^{1-\alpha} \qquad \text{for all } t > t_0. \quad (2.31)
\end{aligned}$$

Equation (2.27) follows from the fact that $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$. Inequality (2.28) arrives from dropping $P(B)$ and $P(A|B^c)$ in the previous expression. We have (2.29) from Lemma 2 and the fact that $I_k(t) = \hat{\mu}_k + \sqrt{\frac{2\alpha\sigma^2 \log t}{n_k(t)}}$. Inequality (2.30) follows from the Hoeffding's inequality and the term t before the exponent in (2.30) arises as the random variable,$n_k(t)$, can take values between s and t (Lemma 1). Equation (2.31) results from the definition of t_0 and the fact that $s > \frac{t}{2K}$.

Plugging the result of (2.31) in the expression (2.26) completes the proof of Lemma 5.

□

Lemma 6. *Let t_0 be the minimum integer satisfying $\Delta_{min} \geq 4\sqrt{\frac{K\alpha\sigma^2 \log t_0}{t_0}}$ then $\forall t > Kt_0$, and $\forall k \neq k^*$, we have,*

$$\Pr\left(n_k(t) > \frac{t}{K}\right) \leq 6K^2\left(\frac{t}{K}\right)^{2-\alpha}.$$

Proof. We expand $\Pr\left(n_k(t) > \frac{t}{K}\right)$ as,

$$\begin{aligned}
\Pr\left(n_k(t) \geq \frac{t}{K}\right) = & \left(\Pr\left(n_k(t) \geq \frac{t}{K} \mid n_k(t-1) \geq \frac{t}{K}\right) \times \Pr\left(n_k(t-1) \geq \frac{t}{K}\right)\right) + \\
& \left(\Pr\left(k_t = k \mid n_k(t-1) = \frac{t}{K} - 1\right) \times \Pr\left(n_k(t-1) = \frac{n}{K} - 1\right)\right) \\
& \leq \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) + \Pr\left(k_t = k, n_k(t-1) = \frac{t}{K} - 1\right) \\
& \leq \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) + 6K(t-1)^{1-\alpha} \quad \forall (t-1) > t_0. \qquad (2.32)
\end{aligned}$$

Here (2.32) follows from Lemma 5.

This gives us that $\forall (t-1) > t_0$, we have,

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) - \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) \leq 6K(t-1)^{1-\alpha}.$$

Now consider the summation

$$\sum_{\tau=\frac{t}{K}}^{t} \Pr\left(n_k(\tau) \geq \frac{t}{K}\right) - \Pr\left(n_k(\tau-1) \geq \frac{t}{K}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} 6K(\tau-1)^{1-\alpha}.$$

This gives us,

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) - \Pr\left(n_k\left(\frac{t}{K} - 1\right) \geq \frac{t}{K}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} 6K(\tau-1)^{1-\alpha}.$$

Since $\Pr\left(n_k\left(\frac{t}{K} - 1\right) \geq \frac{t}{K}\right) = 0$, we have,

$$\begin{aligned}
\Pr\left(n_k(t) \geq \frac{t}{K}\right) & \leq \sum_{\tau=\frac{t}{K}}^{t} 6K(\tau-1)^{1-\alpha} \\
& \leq 6K^2\left(\frac{t}{K}\right)^{2-\alpha} \quad \forall t > Kt_0.
\end{aligned}$$

□

Proof of Theorem 2 We bound $\mathbb{E}\left[n_k(t)\right]$ as

$$
\begin{aligned}
\mathbb{E}\left[n_k(T)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{k_t=k\}}\right] \\
&= \sum_{t=0}^{T-1} \Pr(k_{t+1}=k) \\
&= \sum_{t=1}^{Kt_0} \Pr(k_t=k) + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k) \\
&\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k, n_{k^*}(t)=\max_{k'} n_{k'}(t)) + \\
&\quad \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \left(\Pr(n_{k'}(t)=\max_{k''} n_{k''}(t)) \times \Pr(k_{t+1}=k | n_{k'}(t)=\max_{k''} n_{k''}(t)) \right) \\
&\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k, n_{k^*}(t)=\max_{k'} n_{k'}(t)) + \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \Pr(n_{k'}(t)=\max_{k''} n_{k''}(t)) \\
&\leq Kt_0 + \sum_{t=Kt_0}^{T-1} 2t^{1-\alpha} + \sum_{t=Kt_0}^{T} \sum_{k' \neq k^*} \Pr\left(n_{k'}(t) \geq \frac{t}{K}\right) \qquad (2.33) \\
&\leq Kt_0 + \sum_{t=1}^{T} 2Kt^{1-\alpha} + K^2(K-1) \sum_{t=Kt_0}^{T} 6\left(\frac{t}{K}\right)^{2-\alpha}. \qquad (2.34)
\end{aligned}
$$

Here, (2.33) follows from Lemma 4 and (2.34) follows from Lemma 6.

Proof of Theorem 3 For any sub-optimal arm $k \neq k^*$,

$$
\begin{aligned}
\mathbb{E}\left[n_k(T)\right] &\leq \sum_{t=1}^{T} \Pr(k_t=k) \\
&= \sum_{t=1}^{T} \Pr((k_t=k, E_1(t)) \cup (E_1^c(t), k_t=k)) \qquad (2.35) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t=k) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t=k, I_k(t-1) > I_{k^*}(t-1)) \qquad (2.36) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=0}^{T-1} \Pr(I_k(t) > I_{k^*}(t), k_{t+1}=k) \\
&= \sum_{t=1}^{T} 2Kt^{1-\alpha} + \sum_{t=0}^{T-1} \Pr\left(I_k(t) > I_{k^*}(t), k_{t+1}=k\right) \qquad (2.37) \\
&\leq 8\alpha\sigma^2 \frac{\log(T)}{\Delta_k^2} + \frac{2\alpha}{\alpha-2} + \sum_{t=1}^{T} 2Kt^{1-\alpha}. \qquad (2.38)
\end{aligned}
$$

Here, (2.37) follows from Lemma 3. We have (2.38) from the analysis of UCB for the classical bandit problem. This is because the term $\sum_{t=0}^{T-1} \Pr\left(I_k(t) > I_{k^*}(t), k_{t+1}=k\right)$ counts the number of times $I_k(t) > I_{k^*}(t)$ and $k_{t+1}=k$, which is the exact same term counted in the analysis of the UCB algorithm [32] to bound the expected number of pulls of arm k. In particular, we can see from analysis of Theorem 1 in [32] (equivalently analysis of Theorem 2.1 in [30]) that $\Pr\left(I_k(t) > I_{k^*}(t), k_{t+1} = k, n_k(t) \geq \frac{8\alpha\sigma^2 \log T}{\Delta_k^2}\right) \leq 2t^{1-\alpha}$ and the event $\mathbb{1}\{k_{t+1} = k, I_k(t) > I_{k^*}(t), n_k(t) \leq 8\alpha\sigma^2 \frac{\log T}{\Delta_k^2}\}$ can happen only at-most $8\alpha\sigma^2 \frac{\log T}{\Delta_k^2}$ times. [32] does this analysis for $[0,1]$ bounded random variables (i.e., $\sigma = 1/2$) and $\alpha = 4$ in their proof of Theorem 1. The analysis is replicated in the proof of Theorem 2.1 in [30] for a general α. For further details we refer the reader to the analysis of UCB done in Theorem 2.1 of [30].

2.9.5 Regret analysis for the TS-C Algorithm

We now present results for TS-C in the scenario where $K = 2$ and Thompson sampling is employed with Beta priors [52]. In order to prove results for TS-C, we assume that rewards are either 0 or 1. The Thompson sampling algorithm with beta prior, maintains a posterior distribution on mean of arm k as $Beta\left(n_k(t) \times \hat{\mu}_k(t) + 1, n_k(t) \times (1 - \hat{\mu}_k(t)) + 1\right)$. Subsequently, it generates a sample $S_k(t) \sim Beta\left(n_k(t) \times \hat{\mu}_k(t) + 1, n_k(t) \times (1 - \hat{\mu}_k(t)) + 1\right)$ for each arm k and selects the arm $k_{t+1} = \arg\max_{k \in \mathcal{K}} S_k(t)$. The TS-C algorithm with Beta prior uses this Thompson sampling procedure in its last step, i.e., $k_{t+1} = \arg\max_{k \in \mathcal{C}_t} S_k(t)$, where $\mathcal{C}_t$ is the set of $\hat{\Theta}_t$-Competitive arms at round t. We show that in a 2-armed bandit problem, the regret is O(1) if the sub-optimal arm k is non-competitive and is O$(\log T)$ otherwise.

For the purpose of regret analysis of TS-C, we define two thresholds, a lower threshold $L_k(\theta^*)$, and an upper threshold $U_k(\theta^*)$ for arm $k \neq k^*$,

$$U_k(\theta^*) = \mu_k(\theta^*) + \frac{\Delta_k}{3}, \qquad L_k(\theta^*) = \mu_{k^*}(\theta^*) - \frac{\Delta_k}{3}. \tag{2.39}$$

Let $E_i^{\mu}(t)$ and $E_i^{S}(t)$ be the events that,

$$\begin{aligned} E_k^{\mu}(t) &= \{\hat{\mu}_k(t) \leq U_k(\theta^*)\} \\ E_k^{S}(t) &= \{S_k(t) \leq L_k(\theta^*)\}. \end{aligned} \tag{2.40}$$

To analyse the regret of TS-C, we first show that the number of times arm k is pulled jointly with the event that $n_k(t-1) \geq \frac{t}{2}$ is bounded above by an O(1) constant, which is independent of the total number of rounds T.

Lemma 7. *If $\Delta_k \geq 3\sqrt{\frac{\alpha \log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\sum_{t=2t_0}^{T} \Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$$

where $k \neq k^*$ is a sub-optimal arm.

Proof. We start by bounding the probability of the pull of k-th arm at round t as follows,

$$\begin{aligned}
\Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) &\overset{(a)}{\leq} \Pr\left(E_1(t), k_t = k, n_k(t-1) \geq \frac{t}{2}\right) + \\
&\quad \Pr\left(\overline{E_1(t)}, k_t = k, n_k(t-1) \geq \frac{t}{2}\right) \\
&\overset{(b)}{\leq} 2Kt^{1-\alpha} + \Pr\left(\overline{E_1(t)}, k_t = k, n_k(t-1) \geq \frac{t}{2}\right) \\
&\overset{(c)}{\leq} 2Kt^{1-\alpha} + \underbrace{\Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term A}} + \\
&\quad \underbrace{\Pr\left(k_t = k, E_k^{\mu}(t), \overline{E_k^{S}(t)}, n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term B}} + \\
&\quad \underbrace{\Pr\left(k_t = k, \overline{E_k^{\mu}(t)}, n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term C}} \qquad (2.41)
\end{aligned}$$

$$(2.42)$$

where (b), comes from Lemma 3. Now we treat each term in (2.41) individually. To bound term A, we note that $\Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right) \leq \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right)$. From the analysis in [52] (equation 6), we see that $\sum_{t=1}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right) = \mathrm{O}(1)$ as it is shown through Lemma 2 in [52] that,
$\sum_{t=1}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right) \leq \frac{224}{\Delta_k^2} + \sum_{j=0}^{T} \Theta\left(e^{-\frac{\Delta_k^2 j}{18}} + \frac{1}{e^{\frac{\Delta_k^2 j}{36}} - 1} + \frac{9}{(j+1)\Delta_k^2} e^{-D_k j}\right)$.
Here, $D_k = L_k(\theta^*) \log \frac{L_k(\theta^*)}{\mu_{k^*}(\theta^*)} + (1 - L_k(\theta^*)) \log \frac{1 - L_k(\theta^*)}{1 - \mu_{k^*}(\theta^*)}$. Due to this,
$\sum_{t=2t_0}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$.

We now bound the sum of term B from $t = 1$ to T by noting that
$\Pr\left(k_t = k, E_k^{\mu}(t), \overline{E_k^{S}(t)}, n_k(t-1) \geq \frac{t}{2}\right) \leq \Pr\left(k_t = k, \overline{E_k^{S}(t)}\right)$. Additionally, from Lemma 3 in [52], we get that $\sum_{t=1}^{T} \Pr\left(k_t = k, \overline{E_k^{S}(t)}\right) \leq \frac{1}{d(U_k(\theta^*), \mu_k(\theta^*))} + 1$, where $d(x, y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y}$. As a result, we see that $\sum_{t=1}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), \overline{E_k^{S}(t)}, n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$.

Finally, for the last term C we can show that,

$$\begin{aligned}
(\mathrm{C}) &= \Pr\left(k_t = k, \overline{E_k^{\mu}(t)}, n_k(t-1) \geq \frac{t}{2}\right) \\
&\leq \Pr\left(\overline{E_k^{\mu}(t)}, n_k(t-1) \geq \frac{t}{2}\right) \\
&= \Pr\left(\hat{\mu}_k - \mu_k > \frac{\Delta_k}{3}, n_k(t-1) \geq \frac{t}{2}\right) \\
&\leq t \exp\left(-\frac{t\Delta_k^2}{9}\right) \qquad (2.43) \\
&\leq t^{1-\alpha}
\end{aligned}$$

Here (2.43) follows from hoeffding's inequality and the union bound trick to handle random variable $n_k(t-1)$. After plugging these results in (2.41), we get that

$$\begin{aligned}
\sum_{t=2t_0}^{T} \Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) \leq \sum_{t=2t_0}^{T} 2Kt^{1-\alpha} + \sum_{t=2t_0}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right) + \\
\sum_{t=2t_0}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), \overline{E_k^{S}(t)}, n_k(t-1) \geq \frac{t}{2}\right) + \\
\sum_{t=2t_0}^{T} \Pr\left(k_t = k, \overline{E_k^{\mu}(t)}, n_k(t-1) \geq \frac{t}{2}\right) \quad (2.44)\\
\leq \sum_{t=2t_0}^{T} 2Kt^{1-\alpha} + \mathrm{O}(1) + \mathrm{O}(1) + \sum_{t=2t_0}^{T} t^{1-\alpha} \quad (2.45)\\
= \mathrm{O}(1) \quad (2.46)
\end{aligned}$$

□

We now show that the expected number of pulls by TS-C for a non-competitive arm is bounded above by an $\mathrm{O}(1)$ constant.

Expected number of pulls by TS-C for a non-competitive arm. We bound $\mathbb{E}\left[n_k(t)\right]$ as

$$\begin{aligned}
\mathbb{E}\left[n_k(T)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{k_t = k\}}\right] \\
&= \sum_{t=0}^{T-1} \Pr(k_{t+1} = k) \\
&= \sum_{t=1}^{2t_0} \Pr(k_t = k) + \sum_{t=2t_0}^{T-1} \Pr(k_{t+1} = k) \\
&\leq 2t_0 + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{2}\right) + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1} = k, n_k(t) \geq \frac{t}{2}\right) \quad (2.47)\\
&\leq 2t_0 + \sum_{t=2t_0}^{T-1} 2t^{1-\alpha} + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1} = k, n_k(t) \geq \frac{t}{2}\right) \quad (2.48)\\
&= \mathrm{O}(1) \quad (2.49)
\end{aligned}$$

Here, (2.48) follows from Lemma 4 and (2.49) follows from Lemma 7 and the fact that the sum of $2t^{1-\alpha}$ is bounded for $\alpha > 1$ and $t_0 = \inf\left\{\tau > 0 : \Delta_{\min}, \epsilon_k \geq 3\sqrt{\frac{\alpha \log \tau}{\tau}}\right\}$.

We now show that when the sub-optimal arm k is competitive, the expected pulls of arm k is $\mathrm{O}(\log T)$.

Expected number of pulls by TS-C for a competitive arm $k \neq k^*$: For any sub-optimal arm $k \neq k^*$,

$$\begin{aligned}
\mathbb{E}\left[n_k(T)\right] &\leq \sum_{t=1}^{T} \Pr(k_t = k) \\
&= \sum_{t=1}^{T} \Pr((k_t = k, E_1(t)) \cup (E_1^c(t), k_t = k)) && (2.50)\\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t = k) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t = k, S_k(t-1) > S_{k^*}(t-1)) && (2.51)\\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1} = k) \\
&= \sum_{t=1}^{T} 2Kt^{1-\alpha} + \sum_{t=0}^{T-1} \Pr\left(S_k(t) > S_{k^*}(t), k_{t+1} = k\right) && (2.52)\\
&\leq \frac{9\log(T)}{\Delta_k^2} + \mathrm{O}(1) + \sum_{t=1}^{T} 2Kt^{1-\alpha}. && (2.53)\\
&= \mathrm{O}(\log T). && (2.54)
\end{aligned}$$

Here, (2.52) follows from Lemma 3. We have (2.53) from the analysis of Thompson Sampling for the classical bandit problem in [52]. This arises as the term $\Pr\left(S_k(t) > S_{k^*}(t), k_{t+1} = k\right)$ counts the number of times $S_k(t) > S_{k^*}(t)$ and $k_{t+1} = k$. This is precisely the term analysed in Theorem 3 of [52] to bound the expected pulls of sub-optimal arms by TS. In particular, [52] analyzes the expected number of pull of sub-optimal arm (termed as $\mathbb{E}\left[k_i(T)\right]$ in their paper) by evaluating $\sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1} = k)$ and it is shown in their Section 2.1 (proof of Theorem 1 of [52]) that $\sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1} = k) \leq \mathrm{O}(1) + \frac{\log(T)}{d(x_i, y_i)}$. The term x_i is equivalent to $U_k(\theta^*)$ and y_i is equal to $L_k(\theta^*)$ in our notations. Moreover $d(U_k(\theta^*), L_k(\theta^*)) \leq \frac{\Delta_k^2}{9}$, giving us the desired result of (2.53).

Chapter 3

Multi-Armed Bandits with Correlated Arms

In the last chapter, we studied the structured multi-armed bandit framework where the mean rewards corresponding to different arms are a known function of a hidden parameter θ^*. While the structured bandit armed bandit framework is able to model structure in the mean rewards across different arms, the reward realizations across arms may not necessarily be correlated. In this chapter, we fill this gap by proposing a novel correlated MAB framework which explicitly captures the correlation in reward across different arms. We first motivate the need for correlated multi-armed bandits and contrast them with previously studied MAB models such as contextual and structured bandits.

3.1 Introduction

3.1.1 Background and Motivation

Correlated Multi-Armed Bandits. The classical MAB setting implicitly assumes that the rewards are independent across arms, i.e., pulling an arm k does not provide any information about the reward we would have received from arm ℓ. However, this may not be true in practice as the reward corresponding to different treatment/drugs/ad-versions are likely to be *correlated* with each other. For instance, similar ads/drugs may generate similar reward for the user/patient. These correlations, when modeled and accounted for, can allow us to significantly improve the cumulative reward by reducing the amount of *exploration* in bandit algorithms.

Motivated by this, we study a variant of the classical multi-armed bandit problem in which rewards corresponding to different arms are correlated to each other, i.e., the conditional reward distribution satisfies $f_{R_\ell|R_k}(r_\ell|r_k) \neq f_{R_\ell}(r_\ell)$, whence $\mathbb{E}[R_\ell|R_k] \neq \mathbb{E}[R_\ell]$. Such correlations can only be learned upon obtaining samples from different arms simultaneously, i.e., by pulling multiple arms at a time. As that is not allowed

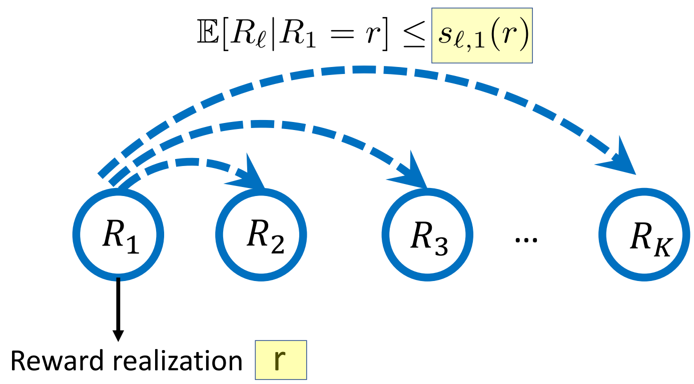


Figure 3.1: Upon observing a reward r from an arm k, pseudo-rewards $s_{\ell,k}(r)$, give us an upper bound on the conditional expectation of the reward from arm ℓ given that we observed reward r from arm k. These pseudo-rewards models the correlation in rewards corresponding to different arms.

in the classical Multi-Armed Bandit formulation, we assume the knowledge of such correlations in the form of prior knowledge that might be obtained through domain expertise or from controlled surveys. One way of capturing correlations is through the knowledge of the joint reward distribution. However, if the complete joint reward distribution is known, then the best-arm is known trivially. Instead, in our work, we only assume restrictive information about correlations in the form of *pseudo-rewards* that constitute an upper bound on conditional expected rewards. This makes our model more general and suitable for practical applications. Fig. 3.1 presents an illustration of our model, where the pseudo-rewards, denoted by $s_{\ell,k}(r)$, provide an upper bound on the reward that we could have received from arm ℓ given that pulling arm k led to a reward of r; i.e.,

$$\mathbb{E}[R_\ell|R_k = r] \leq s_{\ell,k}(r). \tag{3.1}$$

We show that the knowledge of such bounds, even when they are not all tight, can lead to significant improvement in the cumulative reward obtained by reducing the amount of *exploration* compared to classical MAB algorithms. Our proposed MAB model and algorithm can be applied in all real-world applications of the classical Multi-Armed bandit problem, where it is possible to know pseudo-rewards from domain knowledge or through surveyed data. In the next section, we illustrate the applicability of our novel correlated Multi-Armed Bandit model and its differences with the existing contextual and structured bandit works through the example of optimal *ad-selection*.

3.1.2 An Illustrative Example

Suppose that a company is to run a display advertising campaign for one of their products, and its creative team have designed several different versions that can be displayed. It is expected that the user engagement

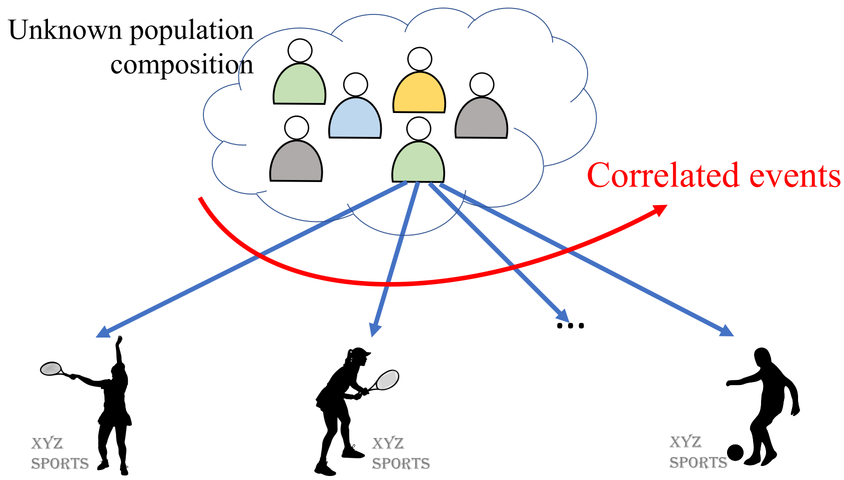


Figure 3.2: The ratings of a user corresponding to different versions of the same ad are likely to be correlated. For example, if a person likes first version, there is a good chance that they will also like the 2nd one as it also related to tennis. However, the population composition is unknown, i.e., the fraction of people liking the first/second or the last version is unknown.

(in terms of click probability and time spent looking at the ad) depends the version of the ad that is displayed. In order to maximize the total user engagement over the course of the ad campaign, multi-armed bandit algorithms can be used; different versions of the ad correspond to the *arms* and the reward from selecting an arm is given by the clicks or time spent looking at the ad version corresponding to that arm.

Personalized recommendations using Contextual and Structured bandits. Although the ad-selection problem can be solved by standard MAB algorithms, there are several specialized MAB variants that are designed to give better performance. For instance, the *contextual* bandit problem [11, 12] has been studied to provide *personalized* displays of the ads to the users. Here, before making a choice at each time step (i.e., deciding which version to show to a user), we observe the *context* associated with that user (e.g., age/occupation/income features). Contextual bandit algorithms learn the mappings from the context θ to the most favored version of ad $k^*(\theta)$ in an online manner and thus are useful for personalized recommendations. A closely related problem is the structured bandit problem [14, 13, 51, 53], in which the context θ (age/ income/ occupational features) is *hidden* but the mean rewards for different versions of ad (arms) as a function of hidden context θ are known. Such models prove useful for personalized recommendation in which the context of the user is unknown, but the reward mappings $\mu_k(\theta)$ are known through surveyed data.

Global Recommendations using Correlated-Reward Bandits. In this work we study a variant of the classical multi-armed bandit problem in which rewards corresponding to different arms are correlated to each other. In many practical settings, the reward we get from different arms at any given step are likely to be correlated. In the ad-selection example given in Figure 3.2, a user reacting positively (by clicking, ordering, etc.) to the first version of the ad with a girl playing tennis might also be more likely to click the second version as it is also related to tennis; of course one can construct examples where there is negative correlation between click events to different ads. The model we study in this chapter explicitly captures these correlations through the knowledge of pseudo-rewards $s_{\ell,k}(r)$ (See Figure 3.1). Similar to the classical MAB setting, the goal here is to display versions of the ad to maximize user engagement. In addition, unlike contextual bandits, we do not observe the context (age/occupational/income) features of the user and do not focus on providing personalized recommendation. Instead our goal is to provide global recommendations to a population whose demographics is unknown. Unlike *structured bandits*, we do not assume that the mean rewards are functions of a hidden context parameter θ. In structured bandits, although the *mean* rewards depend on θ the reward realizations can still be independent. See Section 3.2.4 for more details.

3.1.3 Main Contributions and Organization

i) A General and Previously Unexplored Correlated Multi-Armed Bandit Model. In Section 3.2 we describe our novel correlated multi-armed bandit model, in which rewards of a user corresponding to different arms are correlated with each other. This correlation is captured by the knowledge of *pseudo-rewards*, which are upper bounds on the conditional mean reward of arm ℓ given reward of arm k. In practice, pseudo-rewards can be obtained via expert/domain knowledge (for example, common ingredients in two drugs that are being considered to treat an ailment) or controlled surveys (for example, beta-testing users who are asked to rate different versions of an ad). A key advantage of our framework is that pseudo-rewards are just upper bounds on the conditional expected rewards and can be arbitrarily loose. This also makes the proposed framework and algorithm directly usable in practice – if some pseudo-rewards are unknown due to lack of domain knowledge/data, they can simply be replaced by the maximum possible reward entries, which serves a natural upper bound.

ii) An approach to generalize algorithms to the Correlated MAB setting. We propose a novel approach in Section 3.3 that extends any classical bandit (such as UCB, TS, KL-UCB etc.) algorithm to the correlated MAB setting studied in this chapter. This is done by making use of the pseudo-rewards to reduce exploration in standard bandit algorithms. We refer to this algorithm as C-BANDIT where BANDIT refers to the classical bandit algorithm used in the last step of the algorithm (i.e., UCB/TS/KL-UCB).

iii) Unified Regret Analysis We study the performance of our proposed algorithms by analyzing their expected *regret*, $\mathbb{E}[\text{Reg}(T)]$. The regret of an algorithm is defined as the difference between the cumulative reward of a *genie* policy, that always pulls the optimal arm k^*, and the cumulative reward obtained by the algorithm over T rounds. By doing regret analysis of C-UCB, we obtain the following upper bound on the expected regret of C-UCB.

Proposition 2 (Upper Bound on Expected Regret)**.** *The expected cumulative regret of the C-UCB algorithm is upper bounded as*

$$\mathbb{E}\left[Reg(T)\right] \leq (C-1) \cdot \mathrm{O}(\log T) + \mathrm{O}(1), \tag{3.2}$$

Here C denotes the number of *competitive* arms. An arm k is said to be *competitive* if expected pseudo-reward of arm k with respect to the optimal arm k^* is larger than the mean reward of arm k^*, that is, if $\mathbb{E}\left[s_{k,k^*}(r)\right] \geq \mu_{k^*}$, otherwise, the arm is said to be non-competitive. The result in Proposition 2 arises from the fact that the C-UCB algorithm ends up pulling the non-competitive arms only $\mathrm{O}(1)$ times and only the competitive sub-optimal arms are pulled $\mathrm{O}(\log T)$ times. In contrast to UCB, that pulls all $K-1$ sub-optimal arms $\mathrm{O}(\log T)$ times, our proposed C-UCB algorithm pulls only $C-1 \leq K-1$ arms $\mathrm{O}(\log T)$ times. In fact, when $C=1$, our proposed algorithm achieves *bounded* regret meaning that after some finite step, no arm but the optimal one will be selected. In this sense, we reduce a K-armed bandit problem to a C-armed bandit problem. We emphasize that k^*, μ^* and C are *all* unknown to the algorithm at the beginning.

We present our detailed regret bounds and analysis in Section 3.4. A rigorous analysis of the regret achieved under C-UCB is given through a unified technique. This technique can be of broad interest as we also provide a recipe to obtain regret analysis for any *C-Bandit* algorithm. For instance, the analysis of C-KL-UCB can be easily done through our provided outline.

iv) Evaluation using real-world datasets.

We perform simulations to validate our theoretical results in Section 3.5. In Section 3.6, we do extensive validation of our results by performing experiments on two real-world datasets, namely MovieLens and Goodreads, which show that the proposed approach yields drastically smaller regret than classical Multi-Armed Bandit strategies.

3.2 Problem Formulation

3.2.1 Correlated Multi-Armed Bandit Model

Consider a Multi-Armed Bandit setting with K arms $\{1, 2, \ldots K\}$. At each round t, a user enters the system and we need to decide an arm k_t to display to the user. Upon pulling arm k_t, we receive a random reward

r	$s_{2,1}(r)$
0	0.7
1	0.4

r	$s_{1,2}(r)$
0	0.8
1	0.5

(a)	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.4
$R_2 = 1$	0.2	0.2

(b)	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.3
$R_2 = 1$	0.4	0.1

Table 3.1: The top row shows the pseudo-rewards of arms 1 and 2, i.e., upper bounds on the conditional expected rewards (which are known to the player). The bottom row depicts two possible joint probability distribution (unknown to the player). Under distribution (a), Arm 1 is optimal whereas Arm 2 is optimal under distribution (b).

$R_{k_t} \in [0, B]$. Our goal is to maximize the cumulative reward over time. The expected reward of arm k, is denoted by μ_k. If we knew the arm with highest mean, i.e., $k^* = \arg\max_{k \in \mathcal{K}} \mu_k$ beforehand, then we would always pull arm k^* to maximize expected cumulative reward. We now define the cumulative regret, minimizing which is equivalent to maximizing cumulative reward:

$$Reg(T) = \sum_{t=1}^{T} \mu_{k_t} - \mu_{k^*} = \sum_{k \neq k^*} n_k(T) \Delta_k. \tag{3.3}$$

Here, $n_k(T)$ denotes the number of times a sub-optimal arm is pulled till round T and Δ_k denotes the *sub-optimality gap* of arm k, i.e., $\Delta_k = \mu_{k^*} - \mu_k$.

The classical multi-Armed bandit setting implicitly assumes the rewards $R_1, R_2 \dots R_K$ are independent, that is, $\Pr(R_\ell = r_\ell | R_k = r) = \Pr(R_\ell = r_\ell) \quad \forall r_\ell, r \& \forall \ell, k$, which implies that, $\mathbb{E}\left[R_\ell | R_k = r\right] = \mathbb{E}\left[R_\ell\right] \quad \forall r, \ell, k$. However, in most practical scenarios this assumption is unlikely to be true. In fact, rewards of a user corresponding to different arms are likely to be correlated. Motivated by this we consider a setup where the conditional distribution of the reward from arm ℓ given reward from arm k is not equal to the probability distribution of the reward from arm ℓ, i.e., $f_{R_\ell|R_k}(r_\ell|r_k) \neq f_{R_\ell}(r_\ell)$, with $f_{R_\ell}(r_\ell)$ denoting the probability distribution function of the reward from arm ℓ. Consequently, due to such correlations, we have $\mathbb{E}\left[R_\ell | R_k\right] \neq \mathbb{E}\left[R_\ell\right]$.

In our problem setting, we consider that the player has partial knowledge about the joint distribution of correlated arms in the form of *pseudo-rewards*, as defined below:

Definition 6 (Pseudo-Reward)**.** *Suppose we pull arm k and observe reward r, then the pseudo-reward of arm ℓ with respect to arm k, denoted by $s_{\ell,k}(r)$, is an upper bound on the conditional expected reward of arm ℓ, i.e.,*

$$\mathbb{E}[R_\ell | R_k = r] \leq s_{\ell,k}(r), \tag{3.4}$$

without loss of generality, we define $s_{\ell,\ell}(r) = r$.

The pseudo-rewards information consists of a set of $K \times K$ functions $s_{\ell,k}(r)$ over $[0, B]$. This information can be obtained in practice through either domain/expert knowledge or from controlled surveys. For

r	$s_{2,1}(r)$	$s_{3,1}(r)$
0	0.7	2
1	0.8	1.2
2	2	1

r	$s_{1,2}(r)$	$s_{3,2}(r)$
0	0.5	1.5
1	1.3	2
2	2	0.8

r	$s_{1,3}(r)$	$s_{2,3}(r)$
0	1.5	2
1	2	1.3
2	0.7	0.75

Table 3.2: If some pseudo-reward entries are unknown (due to lack of prior-knowledge/data), those entries can be replaced with the maximum possible reward and then used in the C-BANDIT algorithm. We do that here by entering 2 for the entries where pseudo-rewards are unknown.

instance, in the context of medical testing, where the goal is to identify the best drug to treat an ailment from among a set of K possible options, the effectiveness of two drugs is correlated when the drugs share some common ingredients. Through domain knowledge of doctors, it is possible answer questions such as "what are the chances that drug B would be effective given drug A was not effective?", through which we can infer the pseudo-rewards.

3.2.2 Computing Pseudo-Rewards from prior-data/surveys

The pseudo-rewards can also be learned from prior-available data, or through *offline* surveys in which users are presented with *all* K arms allowing us to sample $R_1, \ldots, R_K$ jointly. Through such data, we can evaluate an estimate on the conditional expected rewards. For example in Table 3.1, we can look at all users who obtained 0 reward for Arm 1 and calculate their average reward for Arm 2, say $\hat{\mu}_{2,1}(0)$. This average provides an estimate on the conditional expected reward. Since we only need an upper bound on $\mathbb{E}\left[R_2 | R_1 = 0\right]$, we can use several approaches to construct the pseudo-rewards.

1. If the training data is *large*, one can use the empirical estimate $\hat{\mu}_{2,1}(0)$ directly as $s_{2,1}(0)$, because through law of large numbers, the empirical average equals the $\mathbb{E}\left[R_2 | R_1 = 0\right]$.

2. Alternatively, we can set $s_{2,1}(0) = \hat{\mu}_{2,1}(0) + \hat{\sigma}_{2,1}(0)$, with $\hat{\sigma}_{2,1}(0)$ denoting the empirical standard deviation on the conditional reward of arm 2, to ensure that pseudo-reward is an upper bound on the conditional expected reward.

3. In addition, pseudo-rewards for any unknown conditional mean reward could be filled with the maximum possible reward for the corresponding arm. Table 3.2 shows an example of a 3-armed bandit problem where some pseudo-reward entries are unknown, e.g., due to lack of data. We can fill these missing entries with maximum possible reward $(i.e., 2)$ as shown in Table 3.2 to complete the pseudo-reward entries.

4. If through the training data, we obtain a soft upper bound u on $\mathbb{E}\left[R_2 | R_1 = 0\right]$ that holds with probability $1 - \delta$, then we can translate it to the pseudo-reward $s_{2,1}(0) = u \times (1 - \delta) + 2 \times \delta$, (assuming maximum possible reward is 2).

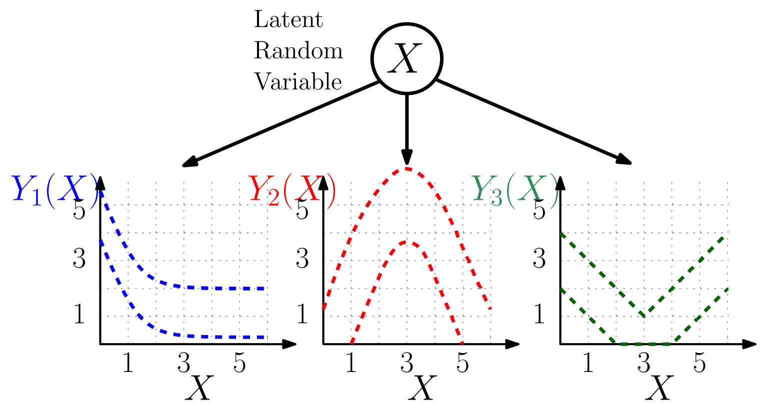


Figure 3.3: A special case of our proposed problem framework is a setting in which rewards for different arms are correlated through a hidden random variable X. At each round X takes a realization in $\mathcal{X}$. The reward obtained from an arm k is $Y_k(X)$. The figure illustrates lower bounds and upper bounds on $Y_k(X)$ (through dotted lines). For instance, when X takes the realization 1, reward of arm 3 is a random variable bounded between 1 and 3.

Remark 5. *Note that the pseudo-rewards are upper bounds on the expected conditional reward and not hard bounds on the conditional reward itself. This makes our problem setup practical as upper bounds on expected conditional reward are easier to obtain, as illustrated in the previous paragraph.*

Remark 6 (Reduction to Classical Multi-Armed Bandits)**.** *When all pseudo-reward entries are unknown, then all pseudo-reward entries can be filled with maximum possible reward for each arm, that is, $s_{\ell,k}(r) = B \; \forall r, \ell, k$. In such a case, the problem framework studied in this chapter reduces to the setting of the classical Multi-Armed Bandit problem and our proposed* C-Bandit *algorithm performs exactly as standard* bandit *(for e.g., UCB, TS etc.) algorithms.*

While the pseudo-rewards are known in our setup, the underlying joint probability distribution of rewards is unknown. For instance, Table 3.1 (a) and Table 3.1 (b) show two joint probability distributions of the rewards that are both possible given the pseudo-rewards at the top of Table 3.1. If the joint distribution is as given in Table 3.1 (a), then Arm 1 is optimal, while Arm 2 is optimal if the joint distribution is as given in Table 3.1(b).

Remark 7. *For a setting where reward domain is* large *or there are a large number of arms, it may be difficult to learn the pseudo-reward entries from prior-data. In such scenarios, the knowledge of additional correlation structure may be helpful to know the value of pseudo-rewards. We describe one such structure in the next section where rewards are correlated through a latent random source and show how to evaluate pseudo-rewards in such a scenario.*

3.2.3 Special Case: Correlated Bandits with a Latent Random Source

Our proposed correlated multi-armed bandit framework subsumes many interesting and previously unexplored multi-armed bandit settings. One such special case is the correlated multi-armed bandit model

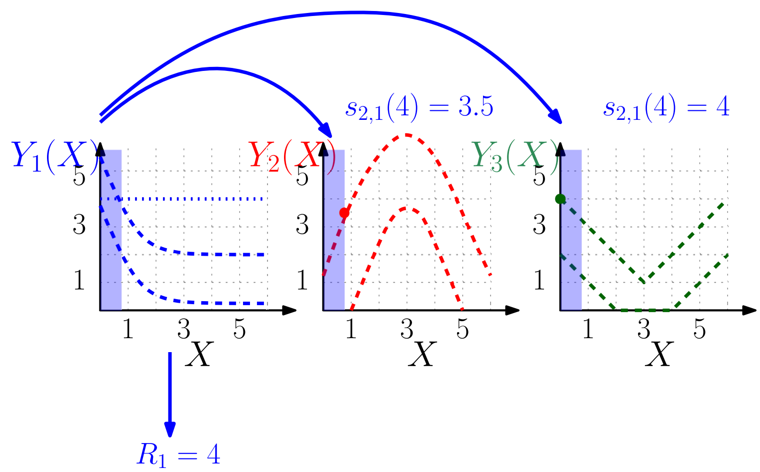


Figure 3.4: An illustration on how to calculate pseudo-rewards in CMAB with latent random source. Upon observing a reward of 4 from arm 1, we can see that the maximum possible reward for arms 2 and 3 is 3.5 and 4 respectively. Therefore, $s_{2,1}(4) = 3.5$ and $s_{3,1}(4) = 4$.

where the rewards depend on a common latent source of randomness [54]. More concretely, the rewards of different arms are correlated through a hidden random variable X (see Figure 3.3). At each round t, X takes a an i.i.d. realization $X_t \in \mathcal{X}$ (unobserved to the player) and upon pulling arm k, we observe a random reward $Y_k(X_t)$. The latent random variable X here could represent the *features* (i.e., age/occupation etc.) of the user arriving to the system, to whom we show one of the K arms. These *features* of the user are hidden in the problem due to privacy concerns. The random reward $Y_k(X_t)$ represents the preference of user with context X_t for the k^{th} version of the ad, for the application of ad-selection.

In this problem setup, upper and lower bounds on $Y_k(X)$, namely $\bar{g}_k(X)$ and $\underline{g}_k(X)$ are known. For instance, the information on upper and lower bounds of $Y_k(X_t)$ could represent knowledge of the form that *children of age 5-10 rate documentaries only in the range 1-3 out of 5*. Such information can be known or learned through prior available data. While the bounds on $Y_k(X)$ are known, the distribution of X and reward distribution within the bounds is unknown, due to which the optimal arm is not known beforehand. Thus, an online approach is needed to minimize the regret.

It is possible to translate this setting to the general framework described in the problem by transforming the mappings $Y_k(X)$ to pseudo-rewards $s_{\ell,k}(r)$. Recall the pseudo-rewards represent an upper bound on the conditional expectation of the rewards. In this framework, $s_{\ell,k}(r)$ can be calculated as:

$$s_{\ell,k}(r) = \max_{\{x:\underline{g}_k(x) \leq r \leq \bar{g}_k(x)\}} \bar{g}_\ell(x),$$

where $\underline{g}_k(x)$ and $\bar{g}_k(x)$ represent upper and lower bounds on $Y_k(x)$. Upon observing a realization from arm k, it is possible to estimate the maximum possible reward that would have been obtained from arm ℓ through the knowledge of bounds on $Y_k(X)$.

Figure 3.4 illustrates how pseudo-reward is evaluated when we obtain a reward $r = 4$ by pulling arm 1. We first infer that X lies in $[0, 0.8]$ if $r = 4$ and then find the maximum possible reward for arm 2 and arm 3 in $[0, 0.8]$. Once these pseudo-rewards are constructed, the problem fits in the general framework described in this chapter and we can use the algorithms proposed for this setting directly.

Remark 8. *In the scenario where $\underline{g}_k(x)$ and $\bar{g}_k(x)$ are soft lower and upper bounds, i.e., $\underline{g}_k(x) \leq Y_k(x) \leq \bar{g}_k(x)$ w.p. $1-\delta$, we can still construct pseudo-reward as follows:*

$$s_{\ell,k}(r) = (1-\delta)^2 \times \left(\max_{\{x: \underline{g}_k(x) \leq r \leq \bar{g}_k(x)\}} \bar{g}_\ell(x) \right) + (1 - (1-\delta)^2) \times M,$$

where M is the maximum possible reward an arm can provide. Thus our proposed framework and algorithms work under this setting as well.[1]

3.2.4 Comparison with parametric (structured) models

As mentioned in Section 3.1, a seemingly related model is the structured bandits model [14, 13, 55]. Structured bandits is a class of problems that cover linear bandits [51], generalized linear bandits [25], Lipschitz bandits [56], global bandits [17], regional bandits [16] etc. In the structured bandits setup, mean rewards corresponding to different arms are related to one another through a hidden parameter θ. The underlying value of θ is fixed and the mean reward mappings $\theta \to \mu_k(\theta)$ are known. Similarly, [57] studies a dependent armed bandit problem, that also has mean rewards corresponding to different arms related to one another. It considers a parametric model, where mean rewards of different arms are drawn from one of the K clusters, each having an unknown parameter π_i. All of these models are fundamentally different from the problem setting considered in this chapter. We list some of the differences with the structured bandits (and the model in [57]) below.

1. In this work we explicitly model the correlations in the rewards of a user corresponding to different arms. While mean rewards are related to each other in structured bandits and [57], the reward realizations are not necessarily correlated.

2. The model studied here is non-parametric in the sense that there is no hidden feature space as is the case in structured bandits and the work of Pandey et al. [57].

3. In structured bandits, the reward mappings from θ to $\mu_k(\theta)$ need to be *exact*. If they happen to be incorrect, then the algorithms for structured bandit cannot be used as they rely on the correctness of

[1] We evaluate a range of values within which x lies based on the reward with probability $1-\delta$. The maximum possible reward of arm ℓ for values of x is then identified with probability $1-\delta$. Due to this, with probability $(1-\delta)^2$, conditional reward of arm ℓ is at-most $\max_{\{x: \underline{g}_k(x) \leq r \leq \bar{g}_k(x)\}} \bar{g}_\ell(x)$.

$\mu_k(\theta)$ to construct confidence intervals on the unknown parameter θ. In contrast, the model studied here relies on the pseudo-rewards being upper bounds on conditional expectations. These bounds need not be tight and the proposed C-Bandit algorithms adjust accordingly and perform at least as well as the corresponding classical bandit algorithm.

4. Similar to the structured bandits, the unimodal bandit framework [58, 59] also assumes a structure on the mean rewards and does not capture the reward correlations explicitly. Under the unimodal framework, it is assumed that the mean reward μ_k as a function of the arms k has a single mode. Instead of assuming that mean rewards are related to one another, our framework explicitly captures the inherent correlations in the form of pseudo-reward. Unimodal bandits have often been used to model the problem of link-rate adaptation in wireless networks, where the mean-reward corresponding to different choices of arms is a unimodal function [60, 61, 62]. The same problem can also be dealt by modeling the correlations explicitly through the pseudo-reward framework described in this chapter.

3.3 The Proposed C-BANDIT Algorithms

We now propose an approach that extends the classical multi-armed bandit algorithms (such as UCB, Thompson Sampling, KL-UCB) to the correlated MAB setting. At each round $t+1$, the UCB algorithm [32] selects the arm with the highest UCB index $I_{k,t}$, i.e.,

$$k_{t+1} = \arg\max_{k \in \mathcal{K}} I_{k,t}, \quad I_{k,t} = \hat{\mu}_k(t) + B\sqrt{\frac{2\log(t)}{n_k(t)}}, \tag{3.5}$$

where $\hat{\mu}_k(t)$ is the empirical mean of the rewards received from arm k until round t, and $n_k(t)$ is the number of times arm k is pulled till round t. The second term in the UCB index causes the algorithm to explore arms that have been pulled only a few times (i.e., those with small $n_k(t)$). Recall that we assume all rewards to be bounded within an interval of size B. When the index t is implied by context, we abbreviate $\hat{\mu}_k(t)$ and $I_k(t)$ to $\hat{\mu}_k$ and I_k respectively in the rest of the chapter.

Under Thompson sampling [52], the arm $k_{t+1} = \arg\max_{k \in \mathcal{K}} S_{k,t}$ is selected at time step $t+1$. Here, $S_{k,t}$ is the sample obtained from the posterior distribution of μ_k, That is,

$$k_{t+1} = \arg\max_{k \in \mathcal{K}} S_{k,t}, \quad S_{k,t} \sim \mathcal{N}\left(\hat{\mu}_k(t), \frac{\beta B}{n_k(t)+1}\right), \tag{3.6}$$

here β is a hyperparameter for the Thompson Sampling algorithm

In the correlated MAB framework, the rewards observed from one arm can help estimate the rewards from other arms. Our key idea is to use this information to reduce the amount of exploration required. We do so by evaluating the *empirical pseudo-reward* of every other arm ℓ with respect to an arm k at each round t. Using this additional information, we identify some arms as *empirically non-competitive* at round t, and only for this round, do not consider them as a candidate in the UCB/Thompson Sampling/(any other bandit algorithm).

3.3.1 Empirical Pseudo-Rewards

In our correlated MAB framework, pseudo-reward of arm ℓ with respect to arm k provides us an estimate on the reward of arm ℓ through the reward sample obtained from arm k. We now define the notion of empirical pseudo-reward which can be used to obtain an *optimistic estimate* of μ_ℓ through just reward samples of arm k.

Definition 7 (Empirical and Expected Pseudo-Reward)**.** *After t rounds, arm k is pulled $n_k(t)$ times. Using these $n_k(t)$ reward realizations, we can construct the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ for each arm ℓ with respect to arm k as follows.*

$$\hat{\phi}_{\ell,k}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{k_\tau = k}\, s_{\ell,k}(r_{k_\tau})}{n_k(t)}, \qquad \ell \in \{1, \ldots, K\} \setminus \{k\}, \tag{3.7}$$

The expected pseudo-reward of arm ℓ with respect to arm k is defined as

$$\phi_{\ell,k} \triangleq \mathbb{E}\left[s_{\ell,k}(R_k)\right]. \tag{3.8}$$

For convenience, we set $\hat{\phi}_{k,k}(t) = \hat{\mu}_k(t)$ and $\phi_{k,k} = \mu_k$.

Observe that $\mathbb{E}\left[s_{\ell,k}(R_k)\right] \geq \mathbb{E}\left[\mathbb{E}\left[R_\ell | R_k = r\right]\right] = \mu_\ell$. Due to this, empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ can be used to obtain an estimated upper bound on μ_ℓ. Note that the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ is defined with respect to arm k and it is only a function of the rewards observed by pulling arm k.

3.3.2 The C-Bandit Algorithm

Using the notion of empirical pseudo-rewards, we now describe a 3-step procedure to fundamentally generalize classical bandit algorithms for the correlated MAB setting.

Step 1: Identify the set $\mathcal{S}_t$ of significant arms: At each round t, define $\mathcal{S}_t$ to be the set of arms that have at least t/K samples, i.e., $\mathcal{S}_t = \{k \in \mathcal{K} : n_k(t) > \frac{t}{K}\}$. As $\mathcal{S}_t$ is the set of arms that have relatively *large* number of samples, we use these arms for the purpose of identifying *empirically competitive* and *empirically non-competitive* arms. Furthermore, define $k^{\text{emp}}(t)$ to be the arm that has the highest empirical mean in set $\mathcal{S}_t$, i.e., $k^{\text{emp}}(t) = \arg\max_{k \in \mathcal{S}_t} \hat{\mu}_k(t)$. [2]

[2] If one were to use all arms (even those that have few samples) to identify empirically non-competitive arms, it can lead to incorrect inference, as pseudo-rewards with few samples will have larger noise, which can in-turn lead to elimination of the optimal arm. Using only the arms that have been pulled $\frac{t}{K}$ times in $\mathcal{S}_t$, allows us to ensure that the non-competitive arms are pulled only O(1) times as we show in Section 3.4.

Step 2: Identify the set of *empirically competitive* arms $\mathcal{A}_t$:

Using the empirical mean, $\hat{\mu}_{k^{\text{emp}}}(t)$, of the arm with highest empirical reward in the set $\mathcal{S}_t$, we define the notions of empirically non-competitive and empirically competitive arms below.

Definition 8 (Empirically Non-Competitive arm at round t)**.** *An arm k is said to be Empirically Non-Competitive at round t, if $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) < \hat{\mu}_{k^{emp}}(t)$.*

Definition 9 (Empirically Competitive arm at round t)**.** *An arm k is said to be Empirically Competitive at round t if $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) \geq \hat{\mu}_{k^{emp}}(t)$. The set of all empirically competitive arms at round t is denoted by $\mathcal{A}_t$.*

The expression $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t)$ provides the tightest estimated upper bound on mean of arm k, through the samples of arms in $\mathcal{S}_t$. If this estimated upper bound is smaller than the estimated mean of $k^{\text{emp}}(t)$, then we call arm k as *empirically non-competitive* as it seems unlikely to be optimal through the samples of arms in $\mathcal{S}_t$. If the estimated upper bound of arm k is greater than $\hat{\mu}_{k^{\text{emp}}}(t)$, i.e., $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) \geq \hat{\mu}_{k^{\text{emp}}}(t)$, we call arm k as empirically competitive at round t, as it cannot be inferred as sub-optimal through samples of arms in $\mathcal{S}_t$. Note that the set of empirically competitive and empirically non-competitive arms is evaluated at each round t and hence an arm that is empirically non-competitive at round t may be empirically competitive in subsequent rounds.

Step 3: Play BANDIT algorithm in $\{\mathcal{A}_t \cup \{k^{\text{emp}}(t)\}\}$ As empirically non-competitive arm seem sub-optimal to be selected at round t, we only consider the set of empirically competitive arms along with $k^{\text{emp}}(t)$ in this step of the algorithm. At round t, we play a BANDIT algorithm from the set $\mathcal{A}_t \cup \{k^{\text{emp}}(t)\}$. For instance, the C-UCB pulls the arm

$$k_t = \arg \max_{k \in \{\mathcal{A}_t \cup k^{emp}(t)\}} I_{k,t-1},$$

where $I_{k,t-1}$ is the UCB index defined in (3.5).

Similarly, C-TS pulls the arm

$$k_t = \arg \max_{k \in \{\mathcal{A}_t \cup k^{emp}(t)\}} S_{k,t-1},$$

where $S_{k,t}$ is the Thompson sample defined in (3.6)). At the end of each round we update the empirical pseudo-rewards $\hat{\phi}_{\ell,k_t}(t)$ for all ℓ, the empirical reward for arm k_t.

Note that our C-BANDIT approach allows using any classical Multi-Armed Bandit algorithm in the correlated Multi-Armed Bandit setting. This is important because some algorithms such as Thompson Sampling and KL-UCB are known to obtain much better empirical performance over UCB. Extending those to the correlated MAB setting allows us to have the superior empirical performance over UCB even in the correlated setting. This benefit is demonstrated in our simulations and experiments described in Section 3.5 and Section 3.6.

Algorithm 4 C-UCB Correlated UCB Algorithm

1: **Input:** Pseudo-rewards $s_{\ell,k}(r)$
2: **Initialize:** $n_k = 0, I_k = \infty$ for all $k \in \{1, 2, \ldots K\}$
3: **for** each round t **do**
4: Find $\mathcal{S}_t = \{k : n_k(t) \geq \frac{t}{K}\}$, the arm that have been pulled significant number of times till $t-1$. Define $k^{\text{emp}}(t) = \arg\max_{k \in \mathcal{S}_t} \hat{\mu}_k(t)$.
5: Initialize the empirically competitive set $\mathcal{A}_t$ as an empty set $\{\}$.
6: **for** $k \in \mathcal{K}$ **do**
7: **if** $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) \geq \hat{\mu}_{k^{\text{emp}}}(t)$ **then**
8: Add arm k to the empirically competitive set: $\mathcal{A}_t = \mathcal{A}_t \cup \{k\}$
9: **end if**
10: **end for**
11: Apply UCB1 over arms in $\mathcal{A}_t \cup \{k^{\text{emp}}(t)\}$ by pulling arm $k_t = \arg\max_{k \in \mathcal{A}_t \cup \{k^{\text{emp}}(t)\}} I_k(t-1)$
12: Receive reward r_t, and update $n_{k_t}(t) = n_{k_t}(t) + 1$
13: Update Empirical reward: $\hat{\mu}_{k_t}(t) = \frac{\hat{\mu}_{k_t}(t-1)(n_{k_t}(t)-1)+r_t}{n_{k_t}(t)}$
14: Update the UCB Index: $I_{k_t}(t) = \hat{\mu}_{k_t}(t) + B\sqrt{\frac{2\log t}{n_{k_t}(t)}}$
15: Update empirical pseudo-rewards for all $k \neq k_t$: $\hat{\phi}_{k,k_t}(t) = \sum_{\tau : k_\tau = k_t} s_{k,k_\tau}(r_\tau)/n_{k_t}(t)$
16: **end for**

Algorithm 5 C-TS Correlated TS Algorithm

1: Steps 1 - 10 as in C-UCB
2: **Apply TS over arms in** $\mathcal{A}_t \cup \{k^{\textbf{emp}}(t)\}$ by pulling arm $k_t = \arg\max_{k \in \mathcal{A}_t \cup \{k^{\text{emp}}(t)\}} S_{k,t}$, where $S_{k,t} \sim \mathcal{N}\left(\hat{\mu}_k(t), \frac{\beta B}{n_k(t)+1}\right)$.
3: Receive reward r_t, and update $n_{k_t}(t)$, $\hat{\mu}_{k_t}(t)$ and empirical pseudo-rewards $\hat{\phi}_{k,k_t}(t)$.

Remark 9 (Pseudo-lower bounds)**.** *If suppose one had the information about pseudo-lower bounds (which are lower bounds on conditional expected rewards), then it is possible to use this in our correlated bandit framework. In step 2 of our algorithm, we identify an arm k as empirically non-competitive if* $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) < \hat{\mu}_k^{emp}(t)$*. We can maintain empirical pseudo-lower bound* $\hat{w}_{i,j}(t)$ *of each arm i with respect to every other arm j. Then, we can replace the step 2 of our algorithm by calling an arm empirically non-competitive if* $\min_{\ell \in \mathcal{S}_t} \hat{\phi}_{k,\ell}(t) < \max_{i \in \mathcal{S}_t} \max_{j \in \mathcal{S}_t} \hat{w}_{i,j}(t)$*. In the situation where pseudo-lower bounds are unknown, they can be set to* $-\infty$ *and the algorithm reduces to the C-Bandit algorithm proposed in the chapter. We can expect the empirical performance of this algorithm (which is aware of pseudo-lower bounds) to be slightly better than the C-Bandit algorithm. However, its regret guarantees will be the same as that of the C-Bandit algorithm. This is because pseudo-upper bounds are crucial to deciding whether an arm is competitive/non-competitive (defined in the next section), and pseudo-lower bounds are not. Put differently, even in the presence of pseudo-lower bound, the definition of non-competitive and competitive arms (Definition 5) remains the same.*

3.4 Regret Analysis and Bounds

We now characterize the performance of the C-UCB algorithm by analyzing the expected value of the cumulative regret (3.3). The expected regret can be expressed as

$$\mathbb{E}\left[Reg(T)\right] = \sum_{k=1}^{K} \mathbb{E}\left[n_k(T)\right] \Delta_k, \tag{3.9}$$

where $\Delta_k = \mu_{k^*} - \mu_k$ is the sub-optimality gap of arm k with respect to the optimal arm k^*, and $n_k(T)$ is the number of times arm k is pulled in T slots.

For the regret analysis, we assume without loss of generality that the rewards are between 0 and 1 for all $k \in \{1, 2, \ldots K\}$. Note that the C-Bandit algorithms do not require this condition, and the regret analysis can also be generalized to any bounded rewards.

3.4.1 Competitive and Non-competitive arms with respect to Arm k

For the purpose of regret analysis in Section 3.4, we need to understand which arms are empirically competitive as $t \to \infty$. We do so by defining the notions of Competitive and Non-Competitive arms.

Definition 10 (Non-Competitive and Competitive arms)**.** *An arm ℓ is said to be non-competitive if the expected reward of optimal arm k^* is larger than the expected pseudo-reward of arm ℓ with respect to the optimal arm k^*, i.e, if, $\tilde{\Delta}_{\ell,k^*} \triangleq \mu_{k^*} - \phi_{\ell,k^*} > 0$. Similarly, an arm ℓ is said to be competitive if $\tilde{\Delta}_{\ell,k^*} = \mu_{k^*} - \phi_{\ell,k^*} <= 0$. The unique best arm k^* has $\tilde{\Delta}_{k^*,k^*} = \mu_{k^*} - \phi_{k^*,k^*} = 0$ and is counted in the set of competitive arms.*[3]

We refer to $\tilde{\Delta}_{\ell,k^*}$ as the pseudo-gap of arm ℓ in the rest of the chapter. These notions of competitiveness are used in the regret analysis in Section 3.4. The central idea behind our correlated C-BANDIT approach is that after pulling the optimal arm k^* sufficiently large number of times, the non-competitive (and thus sub-optimal) arms can be classified as empirically non-competitive with increasing confidence, and thus need not be explored. As a result, the non-competitive arms will be pulled only $\mathrm{O}(1)$ times. However, the competitive arms cannot be discerned as sub-optimal by just using the rewards observed from the optimal arm, and have to be explored $\mathrm{O}(\log T)$ times each. Thus, we are able to reduce a K-armed bandit to a C-armed bandit problem, where C is the number of competitive arms. [4] We show this by bounding the regret of C-BANDIT approach.

[3]As $t \to \infty$, only the optimal arm will remain in $\mathcal{S}_t$, and hence the definition of competitive arms only compares the expected mean of arm k^* and expected pseudo-reward of arm k with respect to arm k^*

[4]Observe that k^* and subsequently C are both unknown to the algorithm. Before the start of the algorithm, it is not known which arm is optimal/competitive/non-competitive. Algorithm works in an online manner by evaluating the noisy notions of competitiveness, i.e., empirically competitive arms, and ensures that only $C - 1$ of the arms are pulled $\mathrm{O}(\log T)$ times.

3.4.2 Regret Bounds

In order to bound $\mathbb{E}\left[Reg(T)\right]$ in (3.9), we can analyze the expected number of times sub-optimal arms are pulled, that is, $\mathbb{E}\left[n_k(T)\right]$, for all $k \neq k^*$. Theorem 5 and Theorem 6 below show that $\mathbb{E}\left[n_k(T)\right]$ scales as $O(1)$ and $O(\log T)$ for non-competitive and competitive arms respectively. Recall that a sub-optimal arm is said to be non-competitive if its pseudo-gap $\tilde{\Delta}_{k,k^*} > 0$, and competitive otherwise.

Theorem 5 (Expected Pulls of a Non-competitive Arm)**.** *The expected number of times a non-competitive arm with pseudo-gap $\tilde{\Delta}_{k,k^*}$ is pulled by C-UCB is upper bounded as*

$$\mathbb{E}\left[n_k(T)\right] \leq Kt_0 + K^3 \sum_{t=Kt_0}^{T} 2\left(\frac{t}{K}\right)^{-2} + \sum_{t=1}^{T} 3t^{-3}, \tag{3.10}$$

$$= \mathrm{O}(1), \tag{3.11}$$

where,

$$t_0 = \inf\left\{\tau \geq 2 : \Delta_{min}, \tilde{\Delta}_{k,k^*} \geq 4\sqrt{\frac{2K\log\tau}{\tau}}\right\}.$$

Theorem 6 (Expected Pulls of a Competitive Arm)**.** *The expected number of times a competitive arm is pulled by C-UCB algorithm is upper bounded as*

$$\mathbb{E}\left[n_k(T)\right] \leq 8\frac{\log(T)}{\Delta_k^2} + \left(1 + \frac{\pi^2}{3}\right) + \sum_{t=1}^{T} 2Kt\exp\left(-\frac{t\Delta_{min}^2}{2K}\right), \tag{3.12}$$

$$= \mathrm{O}(\log T) \quad \textit{where } \Delta_{min} = \min_k \Delta_k > 0. \tag{3.13}$$

Substituting the bounds on $\mathbb{E}\left[n_k(T)\right]$ derived in Theorem 5 and Theorem 6 into (3.9), we get the following upper bound on expected regret.

Corollary 1 (Upper Bound on Expected Regret)**.** *The expected cumulative regret of the C-UCB and C-TS algorithms is upper bounded as*

$$\mathbb{E}\left[Reg(T)\right] \leq \sum_{k \in \mathcal{C}\setminus\{k^*\}} \Delta_k U_k^{(c)}(T) + \sum_{k' \in \{1,\ldots,K\}\setminus\{\mathcal{C}\}} \Delta_{k'} U_{k'}^{(nc)}(T), \tag{3.14}$$

$$= (C-1)\cdot \mathrm{O}(\log T) + \mathrm{O}(1), \tag{3.15}$$

where $\mathcal{C} \subseteq \{1,\ldots,K\}$ is set of competitive arms with cardinality C, $U_k^{(c)}(T)$ is the upper bound on $\mathbb{E}\left[n_k(T)\right]$ for competitive arms given in (6)*, and $U_k^{(nc)}(T)$ is the upper bound for non-competitive arms given in* (5)*.*

3.4.3 Proof Sketch

We now present an outline of our regret analysis of C-UCB. A key strength of our analysis is that it can be extended very easily to any C-BANDIT algorithm. The results independent of last step in the algorithm are presented in Section 3.8.2, while the rigorous regret upper bounds for C-UCB is presented in Section 3.8.3. We also present a regret analysis for C-TS in a scenario where $K = 2$, and TS is employed with Beta priors in Section 3.8.5.

There are three key components to prove the result in Theorem 5 and Theorem 6. The first two components hold independent of which bandit algorithm (UCB/TS/KL-UCB) is used for selecting the arm from the set of competitive arms, which makes our analysis easy to extend to any C-BANDIT algorithm. The third step is specific to the last step in C-BANDIT algorithm. We analyse the third component for C-UCB to provide its rigorous regret results.

i) Probability of optimal arm being identified as empirically non-competitive at round t (denoted by $\Pr(E_1(t))$) is small. In particular, we show that

$$\Pr(E_1(t)) \leq 2Kt \exp\left(-\frac{t\Delta_{\min}^2}{2K}\right).$$

This ensures that the optimal arm is identified as empirically non-competitive only O(1) times. We show that the number of times a competitive arm is pulled is bounded as

$$\mathbb{E}\left[n_k(T)\right] \leq \sum_{t=1}^{T} \Pr(E_1(t)) + \Pr(E_1^c(t), k_t = k, I_{k,t-1} > I_{k^*,t-1}). \tag{3.16}$$

The first term sums to a constant, while the second term is upper bounded by the number of times UCB pulls the sub-optimal arm k. Due to this the upper bound on the number of pulls of competitive arm by C-UCB / C-TS is only an additive constant more than the upper bound on the number of pulls for an arm by UCB / TS algorithms and hence we have same pre-log constants for the upper bound on the pulls of competitive arms.

ii) Probability of identifying a non-competitive arm as empirically competitive jointly with optimal arm being pulled more than $\frac{t}{K}$ times is small. Notice that the first two steps of our algorithm involve identifying the set of arms $\mathcal{S}_t$ that have been pulled at least $\frac{t}{K}$ times, and eliminating arms which are empirically non-competitive with respect to the set $\mathcal{S}_t$ for round t. We show that the joint event that arm $k^* \in \mathcal{S}_t$ and a non-competitive arm k is identified as empirically non-competitive is small. Formally,

$$\Pr\left(k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) \leq t \exp\left(-\frac{t\tilde{\Delta}_{k,k^*}}{2K}\right). \tag{3.17}$$

This occurs because upon obtaining a *large* number of samples of arm k^*, expected reward of arm k^* (i.e., μ_{k^*}) and expected pseudo-reward of arm k with respect to arm k^* (i.e., ϕ_{k,k^*}) can be estimated *fairly* *accurately*. Since the pseudo-gap of arm k is positive (i.e., $\mu_{k^*} > \phi_{k,k^*}$), the probability that arm k is identified as empirically competitive is small. An implication of (3.17) is that the expected number of times a non-competitive arm is identified as empirically competitive jointly with the optimal arm having at least $\frac{t}{K}$ pulls at round t is bounded above by a constant.

iii) **Probability that a sub-optimal arm is pulled more than t/K times at round t is small.** Formally, we show that for C-UCB, we have

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) \leq (2K+2)\left(\frac{t}{K}\right)^{-2} \quad \forall t > Kt_0, k \neq k^* \tag{3.18}$$

This component of our analysis is specific to the classical bandit algorithm used in C-BANDIT. Intuitively, a result of this kind should hold for any *good performing* classical multi-armed bandit algorithm. We reach the result of (3.18) in C-UCB by showing that

$$\Pr\left(k_{t+1} = k, n_k(t) > \frac{t}{2K}\right) \leq t^{-3} \quad \forall t > t_0, k \neq k^* \tag{3.19}$$

The probability of selecting a sub-optimal arm k after it has been pulled *significantly* many times is small as with more number of pulls, the exploration component in UCB index of arm k becomes small, and consequently it is likely to be smaller than the UCB index of optimal arm k^* (as it has larger empirical mean reward or has been pulled fewer number of times). Our analysis in Lemma 16 shows how the result in (3.19) can be translated to obtain (3.18) (this translation is again not dependent on which bandit algorithm is used in C-BANDIT).

We show that the expected number of pulls of a non-competitive arm k can be bounded as

$$\mathbb{E}\left[n_k(T)\right] \leq \sum_{t=1}^{T} \Pr\left(k_{t+1} = k, k^* = \arg\max_k n_k(t)\right) + \Pr\left(k^* \neq \arg\max_k n_k(t)\right) \tag{3.20}$$

The first term in (3.20) is $O(1)$ due to (3.17) and the second term is $O(1)$ due to (3.18). Refer to Section 3.8.3 for rigorous regret analysis of C-UCB.

3.4.4 Discussion on Regret Bounds

Competitive Arms. Recall than an arm is said to be competitive if μ_{k^*} (i.e., expected reward from arm k^*) $> \mathbb{E}\left[\phi_{k,k^*}\right] = \mathbb{E}\left[\tilde{\mathbb{E}}[R_{k'}|R_k]\right]$. Since the distribution of reward of each arm is unknown, initially the Algorithm does not know which arm is *competitive* and which arm is *non-competitive*.

Reduction in effective number of arms. Interestingly, our result from Theorem 5 shows that the C-UCB algorithm, that operates in a sequential fashion, makes sure that *non-competitive* arms are pulled only $O(1)$ times. Due to this, only the competitive arms are pulled $O(\log T)$ times. Moreover, the pre-log terms in the upper bound of UCB and C-UCB for these arms is the same. In this sense, our C-BANDIT approach

$p_1(r)$	**r**	$s_{2,1}(r)$	$s_{3,1}(r)$
0.2	**0**	0.7	2
0.2	**1**	0.8	1.2
0.6	**2**	2	1

Table 3.3: Suppose Arm 1 is optimal and its unknown probability distribution is $(0.2, 0.2, 0.6)$, then $\mu_1 = 1.4$, while $\phi_{2,1} = 1.5$ and $\phi_{3,1} = 1.2$. Due to this Arm 2 is Competitive while Arm 3 is non-competitive

reduces a K-armed bandit problem to a C-armed bandit problem. Effectively only $C - 1 \leq K - 1$ arms are pulled $\mathrm{O}(\log T)$ times, while other arms are stopped being pulled after a finite time.

Depending on the joint probability distribution, different arms can be optimal, competitive or non-competitive. Table 3.3 shows a case where arm 1 is optimal and the reward distribution of arm 1 is $(0.2, 0.2, 0.6)$, which leads to $\mu_1 = 1.4 > \phi_{3,1} = 1.2$ and $\mu_1 = 1.4 < \phi_{2,1} = 1.5$. Due to this Arm 2 is competitive while Arm 3 is non-competitive.

Achieving Bounded Regret. If the set of competitive arms $\mathcal{C}$ is a singleton set containing only the optimal arm (i.e., the number of competitive arms $C = 1$), then our algorithm will lead to (see (3.15)) an expected regret of $\mathrm{O}(1)$, instead of the typical $\mathrm{O}(\log T)$ regret scaling in classic multi-armed bandits. One such scenario in which this can happen is if pseudo-rewards s_{k,k^*} of all arms with respect to optimal arm k^* match the conditional expectation of arm k. Formally, if $s_{k,k^*} = \mathbb{E}\left[R_k | R_{k^*}\right] \forall k$, then $\mathbb{E}\left[s_{k,k^*}\right] = \mathbb{E}\left[R_k\right] = \mu_k < \mu_{k^*}$. Due to this, all sub-optimal arms are non-competitive and our algorithms achieve only $\mathrm{O}(1)$ regret. We now evaluate a lower bound result for a special case of our model, where rewards are correlated through a latent random variable X as described in Section 3.2.3.

We present a lower bound on the expected regret for the model described in Section 3.2.3. Intuitively, if an arm ℓ is *competitive*, it can not be deemed sub-optimal by only pulling the optimal arm k^* infinitely many times. This indicates that exploration is necessary for competitive sub-optimal arms. The proof of this bound closely follows that of the 2-armed classical bandit problem [1]; i.e., we construct a new bandit instance under which a previously sub-optimal arm becomes optimal without affecting reward distribution of any other arm.

Theorem 7 (Lower Bound for Correlated MAB with latent random source)**.** *For any algorithm that achieves a sub-polynomial regret, the expected cumulative regret for the model described in Section 3.2.3 is lower bounded as*

$$\lim_{T \to \infty} \inf \frac{\mathbb{E}\left[Reg(T)\right]}{\log(T)} \geq \begin{cases} \max_{k \in \mathcal{C}} \frac{\Delta_k}{D(f_{R_k} || f_{\tilde{R}_k})} & \textit{if } C > 1 \\ 0 & \textit{if } C = 1. \end{cases} \tag{3.21}$$

Here f_{R_k} is the reward distribution of arm k, which is linked with f_X since $R_k = Y_k(X)$. The term $f_{\tilde{R}_k}$ represents the reward distribution of arm k in the new bandit instance where arm k becomes optimal and

r	$s_{2,1}(r)$
0	0.7
1	0.4

r	$s_{1,2}(r)$
0	0.8
1	0.5

(a)	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.4
$R_2 = 1$	0.2	0.2

(b)	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.3
$R_2 = 1$	0.4	0.1

Table 3.4: The top row shows the pseudo-rewards of arms 1 and 2, i.e., upper bounds on the conditional expected rewards (which are known to the player). The bottom row depicts two possible joint probability distribution (unknown to the player). Under distribution (a), Arm 1 is optimal whereas Arm 2 is optimal under distribution (b).

distribution $f_{R_{k^*}}$ is unaffected. The divergence term represents "the amount of distortion needed in reward distribution of arm k to make it better than arm k^*", and hence captures the problem difficulty in the lower bound expression.

Bounded regret whenever possible for the special case of Section 3.2.3. From Corollary 1, we see that whenever $C > 1$, our proposed algorithm achieves $\mathrm{O}(\log T)$ regret matching the lower bound given in Theorem 7 order-wise. Also, when $C = 1$, our algorithm achieves $\mathrm{O}(1)$ regret. Thus, our algorithm achieves bounded regret whenever possible, i.e., when $C = 1$ for the model described in Section 3.2.3. In the general problem setting, a lower bound $\Omega(\log T)$ exists whenever it is possible to change the joint distribution of rewards such that the marginal distribution of optimal arm k^* is unaffected and pseudo-rewards $s_{\ell,k}(r)$ still remain an upper bound on $\mathbb{E}\left[R_\ell | R_k = r\right]$ under the new joint probability distribution. In general, this can happen even if $C = 1$, we discuss one such scenario in the Section 3.8.6 and explain the challenges that need to come from the algorithmic side to meet the lower bound.

3.5 Simulations

We now present the empirical performance of proposed algorithms. For all the results presented in this section, we compare the performance of all algorithms on the same reward realizations and plot the cumulative regret averaged over 100 independent trials. The shaded area represents error bars with one standard deviation. We set $\beta = 1$ for all TS and C-TS plots.

3.5.1 Simulations with known pseudo-rewards

Consider the example shown in Table 3.1, with the top row showing the pseudo-rewards, which are known to the player, and the bottom row showing two possible joint probability distributions (a) and (b), which are unknown to the player. We show the simulation result of our proposed algorithms C-UCB, C-TS against UCB, TS in Figure 3.5 for the setting considered in Table 3.1.

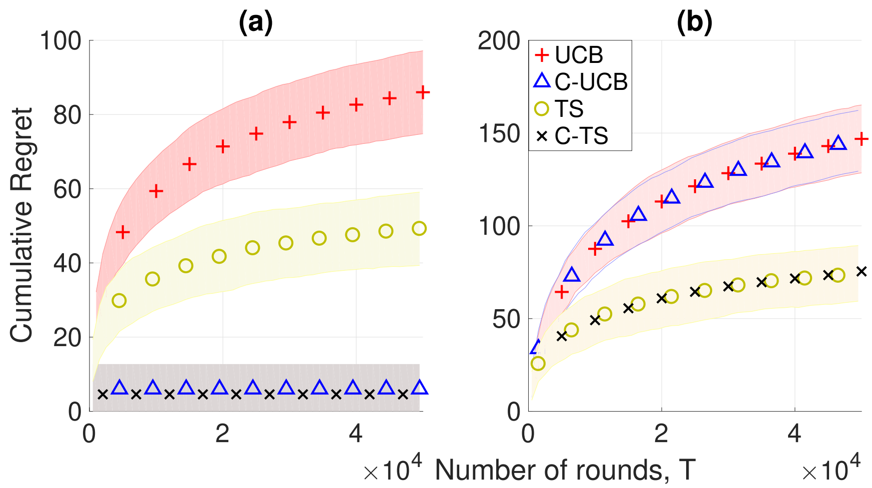


Figure 3.5: Cumulative regret for UCB, C-UCB, TS and C-TS corresponding to the problem shown in Table 3.4. For the setting (a) in Table 3.4, Arm 1 is optimal and Arm 2 is non-competitive, in setting (b) of Table 3.4 Arm 2 is optimal while Arm 1 is competitive.

Case (a): Bounded regret. For the probability distribution (a), notice that Arm 1 is optimal with $\mu_1 = 0.6, \mu_2 = 0.4$. Moreover, $\phi_{2,1} = 0.4 \times 0.7 + 0.6 \times 0.4 = 0.52$. Since $\phi_{2,1} < \mu_1$, Arm 2 is non-competitive. Hence, in Figure 3.5(a), we see that our proposed C-UCB and C-TS Algorithms achieve bounded regret, whereas UCB, TS show logarithmic regret.

Case (b): All competitive arms. For the probability distribution (b), Arm 2 is optimal with $\mu_2 = 0.5$ and $\mu_1 = 0.4$. The expected pseudo-reward of arm 1 w.r.t to arm 2 in this case is $\phi_{1,2} = 0.8 \times 0.5 + 0.5 \times 0.5 = 0.65$. Since $\phi_{1,2} > \mu_2$, the sub-optimal arm (i.e., Arm 1) is competitive and hence C-UCB and C-TS also end up exploring Arm 1. Due to this we see that C-UCB, C-TS achieve a regret similar to UCB, TS in Figure 3.5(b). C-TS has empirically smaller regret than C-UCB as Thompson Sampling performs better empirically than the UCB algorithm. The design of our C-Bandit approach allows the use of any other bandit algorithm in the last step, e.g., KL-UCB.

3.5.2 Simulations for the latent random source model in Section 3.2.3

We now show the performance of C-UCB and C-TS against UCB, TS for the model considered in Section 3.2.3, where rewards corresponding to different arms are correlated through a latent random variable X. We consider a setting where reward obtained from Arm 1, given a realization x of X, is bounded between $2x - 1$ and $2x + 1$, i.e., $2X - 1 \leq Y_1(X) \leq 2X + 1$. Similarly, conditional reward of Arm 2 is, $(3 - X)^2 - 1 \leq Y_2(X) \leq (3 - X)^2 + 1$. Figure 3.6 demonstrates these upper and lower bounds on $Y_k(X)$. We run C-UCB, C-TS, TS and UCB for this setting for two different distributions of X. For the simulations, we set the conditional reward of both the arms to be distributed uniformly between the upper and lower bounds, however this information is not known to the Algorithms.

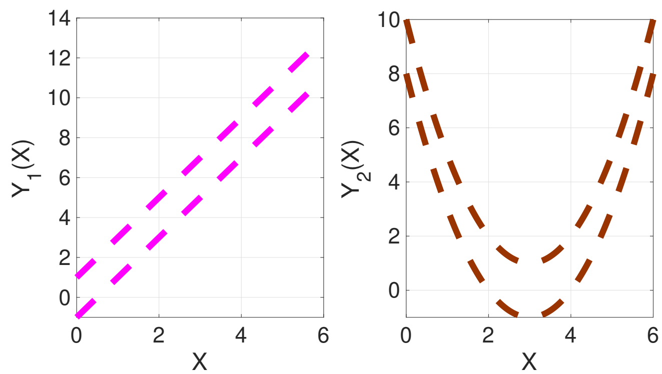


Figure 3.6: Rewards corresponding to the two arms are correlated through a random variable X lying in $(0,6)$. The lines represent lower and upper bounds on reward of Arms 1 ,$Y_1(X)$, and 2, $Y_2(X)$, given the realization of random variable X.

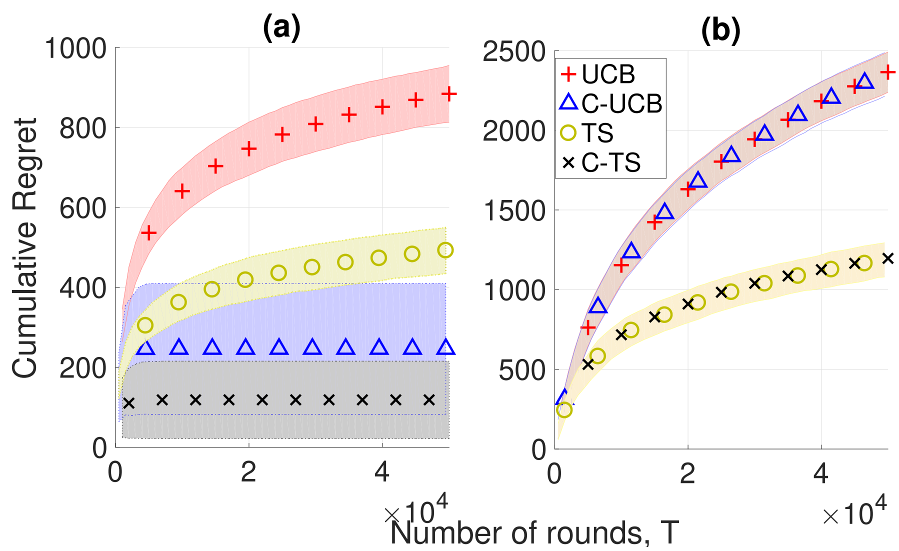


Figure 3.7: Simulation results for the example shown in Figure 3.6. In (a), $X \sim \text{Beta}(1,1)$ and in (b) $X \sim \text{Beta}(1.5,5)$. In case (a), Arm 1 is optimal while Arm 2 is non-competitive (C = 1), due to which we see that C-UCB and C-TS obtain bounded regret. Arm 2 is optimal for the distribution in (b) and Arm 1 is competitive, due to which $C = 2$ and we see that C-UCB and C-TS attain a performance similar to UCB and TS.

Case (a): $X \sim$ **Beta**$(1,1)$. When X is distributed as $X \sim \text{Beta}(1,1)$, Arm 1 is optimal while Arm 2 is non-competitive. Due to this, we observe that C-UCB and C-TS achieve bounded regret in Figure 3.7(a).

Case (b): $X \sim$ **Beta**$(1.5,5)$. In the scenario where X has the distribution $\text{Beta}(1.5,5)$, Arm 2 is optimal while Arm 1 is competitive. Due to this, C-UCB and C-TS do not stop exploring Arm 1 in finite time and we see the cumulative regret similar to UCB, TS in Figure 3.7(b).

Our next simulation result considers a setting where the known upper and lower bounds on $Y_k(X)$ match and the reward Y_k corresponding to a realization of X is deterministic, i.e., $Y_k(X) = g_k(X)$. We show our simulation results for the reward functions described in Figure 3.8 with three different distributions of X. Corresponding to $X \sim \text{Beta}(4,4)$, Arm 1 is optimal and Arms 2,3 are non-competitive leading to

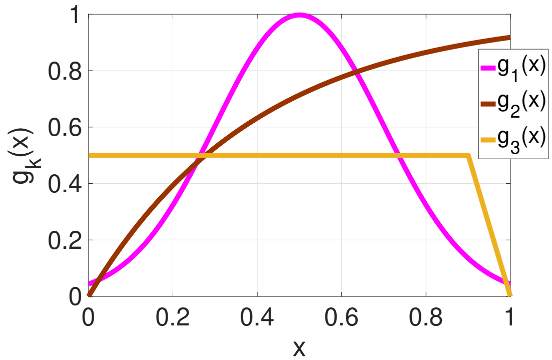


Figure 3.8: Reward Functions used for the simulation results presented in Figure 3.9. The reward $g_k(X)$ is a function of a latent random variable X. For instance, when $X = 0.5$, reward from Arms 1,2 and 3 are $g_1(X) = 1$, $g_2(X) = 0.7135$ and $g_3(X) = 0.5$.

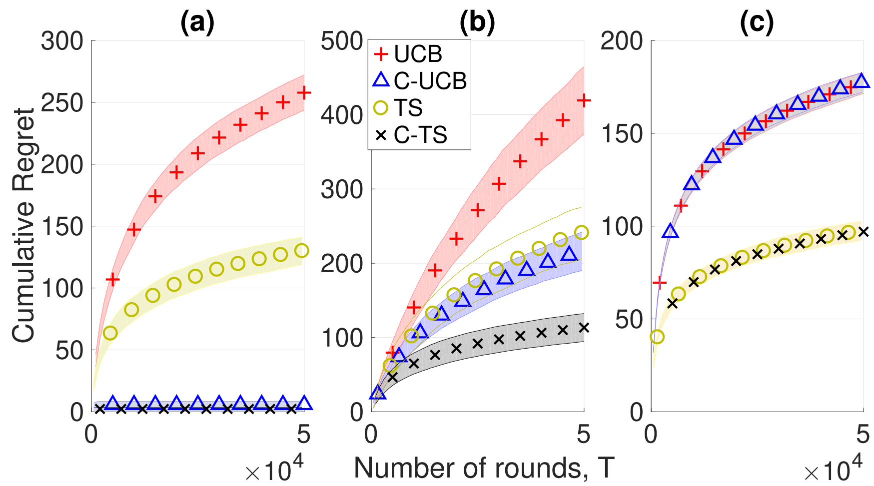


Figure 3.9: The cumulative regret of C-UCB and C-TS depend on C, the number of *competitive* arms. The value of C depends on the *unknown* joint probability distribution of rewards and is not known beforehand. We consider a setup where $C = 1$ in (a), $C = 2$ in (b) and $C = 3$ in (c). Our proposed algorithm pull only the $C - 1$ competitive sub-optimal arms $O(\log T)$ times, as opposed to UCB, TS that pull all $K - 1$ sub-optimal arms $O(\log T)$ times. Due to this, we see that our proposed algorithms achieve bounded regret when $C = 1$. When $C = 3$, our proposed algorithms perform as well as the UCB, TS algorithms.

bounded regret for C-UCB, C-TS in Figure 3.9(a). In setting (b), we consider $X \sim \text{Beta}(2, 5)$ in which Arm 1 is optimal, Arm 2 is competitive and Arm 3 is non-competitive. Due to this, our proposed C-UCB and C-TS Algorithms stop pulling Arm 3 after some time and hence achieve significantly reduced regret relative to UCB in Figure 3.9(b). For third scenario (c), we set $X \sim \text{Beta}(1, 5)$, which makes Arm 3 optimal while Arms 1 and 2 are competitive. Hence, our algorithms explore both the sub-optimal arms and have a regret comparable to that of UCB, TS in Figure 3.9(c).

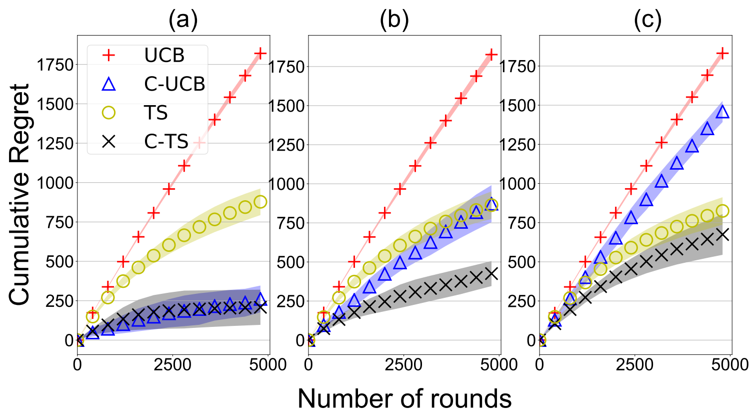


Figure 3.10: Cumulative regret for UCB, C-UCB, TS and C-TS for the application of recommending the best genre in the Movielens dataset, where p fraction of the pseudo-entries are replaced with maximum reward *i.e.*, 5. In (a), $p = 0.25$, for (b), $p = 0.50$ and $p = 0.7$ in (c). The value of C is $4, 11$ and 13 in (a), (b) and (c) respectively. As C is smaller than K (i.e., 18) in each case, we see that C-UCB and C-TS outperform UCB and TS significantly.

3.6 Experiments

We now show the performance of our proposed algorithms in real-world settings. Through the use of MovieLens and Goodreads datasets, we demonstrate how the correlated MAB framework can be used in practical settings for recommendation system applications. In such systems, it is possible to use the prior available data (from a certain population) to learn the correlation structure in the form of pseudo-rewards. When trying to design a campaign to maximize user engagement in a new unknown demographic, the learned correlation information in the form of pseudo-rewards can help significantly reduce the regret as we show from our results described below.

3.6.1 Experiments on the MovieLens dataset

The MovieLens dataset [19] contains a total of 1M ratings for a total of 3883 Movies rated by 6040 Users. Each movie is rated on a scale of 1-5 by the users. Moreover, each movie is associated with one (and in some cases, multiple) genres. For our experiments, of the possibly several genres associated with each movie, one is picked uniformly at random. To perform our experiments, we split the data into two parts, with the first half containing ratings of the users who provided the most number of ratings. This half is used to learn the pseudo-reward entries, the other half is the test set which is used to evaluate the performance of the proposed algorithms. Doing such a split ensures that the rating distribution is different in the training and test data.

Recommending the Best Genre. In our first experiment, the goal is to provide the best genre recommendations to a population with unknown demographic. We use the training dataset to learn the pseudo-reward

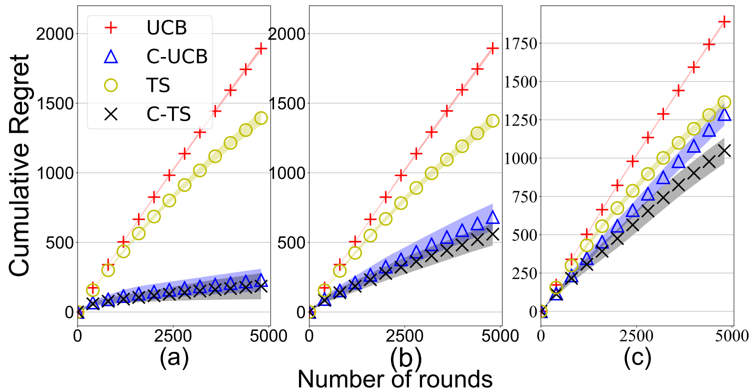


Figure 3.11: Cumulative regret of UCB, C-UCB, TS and C-TS for providing the best movie recommendations in the Movielens dataset. Each pseudo-reward entry is added by 0.1 in (a), 0.4 in (b) and 0.6 in (c). The value of C is $6, 24$ and 39 in $(a), (b)$ and (c) respectively. As C is smaller than K (i.e., 50) in each case, we see the superior performance of C-UCB, C-TS over UCB and TS.

entries. The pseudo-reward entry $s_{\ell,k}(r)$ is evaluated by taking the empirical average of the ratings of genre ℓ that are rated by the users who rated genre k as r. To capture the fact that it might not be possible in practice to fill all pseudo-reward entries, we randomly remove p-fraction of the pseudo-reward entries. The removed pseudo-reward entries are replaced by the maximum possible rating, i.e., 5 (as that gives a natural upper bound on the conditional mean reward). Using these pseudo-rewards, we evaluate our proposed algorithms on the test data. Upon recommending a particular genre (arm), the rating (reward) is obtained by sampling one of the ratings for the chosen arm in the test data. Our experimental results for this setting are shown in Figure 3.10, with $p = 0.25, 0.50$ and 0.70 (i.e., fraction of pseudo-reward entries that are removed). We see that the proposed C-UCB and C-TS algorithms significantly outperform UCB and TS in all three settings. For each of the three cases we also evaluate the value of C (which is unknown to the algorithm), by always pulling the optimal arm and finding the size of empirically competitive set at $T = 5000$. The value of C turned out to be $4, 11$ and 13 for $p = 0.25, 0.50$ and 0.70. As $C < 18$ in each case, some of the 18 arms are stopped being pulled after some time and due to this, C-UCB and C-TS significantly outperform UCB and TS respectively. This shows that even when only a subset of the correlations are known, it is possible to exploit them to improve the performance of classical bandit algorithms.

Recommending the Best Movie. We now consider the goal of providing the best movie recommendations to the population. To do so, we consider the 50 most rated movies in the dataset. containing 109,804 user-ratings given by 6,025 users. In the testing phase, the goal is to recommend one of these 50 movies to each user. As was the case in previous experiment, we learn the pseudo-reward entries from the training data. Instead of using the learned pseudo-reward directly, we add a *safety buffer* to each of the pseudo-reward entry; i.e., we set the pseudo-reward as the empirical conditional mean *plus* the SAFETY BUFFER. Adding a buffer will be needed in practice, as the conditional expectations learned from the training data are likely to

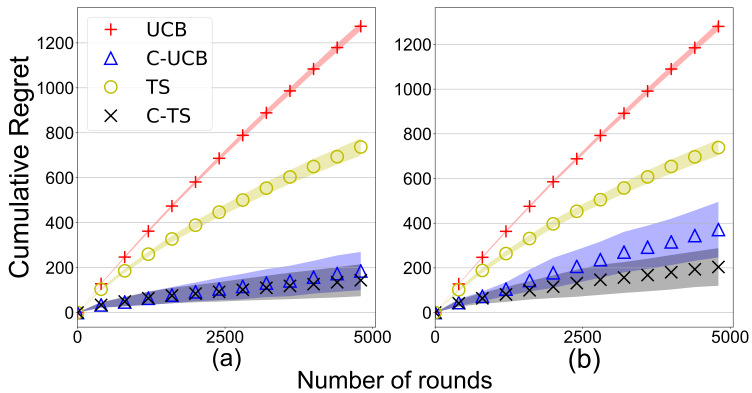


Figure 3.12: Cumulative regret of UCB, C-UCB, TS and C-TS for providing best poetry book recommendation in the Goodreads dataset. Every pseudo-reward entry is added by q and p fraction of the pseudo-reward entries are removed, with (a) $p = 0.1, q = 0.1$ and (b) $p = 0.3, q = 0.1$. The value of C is 8 and 11 in (a) and (b) respectively. As C is much smaller than K (i.e., 25) in each case, we see that C-UCB and C-TS outperform UCB and TS significantly.

have some noise and adding a safety buffer allows us to make sure that pseudo-rewards constitute an upper bound on the conditional expectations. Our experimental result in Figure 3.11 shows the performance of C-UCB and C-TS relative to UCB for this setting with safety buffer set to 0.1 in Figure 3.11(a), 0.4 in Figure 3.11(b) and 0.6 in Figure 3.11(c). In all three cases, even after addition of safety buffers, our proposed C-UCB and C-TS algorithms outperform the UCB algorithm.

3.6.2 Experiments on the Goodreads dataset

The Goodreads dataset [20] contains the ratings for 1,561,465 books by a total of 808,749 users. Each rating is on a scale of 1-5. For our experiments, we only consider the poetry section and focus on the goal of providing best poetry recommendations to the whole population whose demographics is unknown. The poetry dataset has 36,182 different poems rated by 267,821 different users. We do the pre-processing of goodreads dataset in the same manner as that of the MovieLens dataset, by splitting the dataset into two halves, train and test. The train dataset contains the ratings of the users with most number of recommendations.

Recommending the best poetry book. We consider the 25 most rated books in the dataset and use these as the set of arms to recommend in the testing phase. These 25 poems have 349,523 user-ratings given by 171,433 users. As with the MovieLens dataset, the pseudo-reward entries are learned on the training data. In practical situations it might not be possible to obtain all pseudo-reward entries. Therefore, we randomly select p fraction of the pseudo-reward entries and replace them with maximum possible reward (i.e. 5). Among the remaining pseudo-reward entries we add a safety buffer of q to each entry. Our result in Figure 3.12 shows the performance of C-UCB and C-TS relative to UCB and TS in two scenarios. In

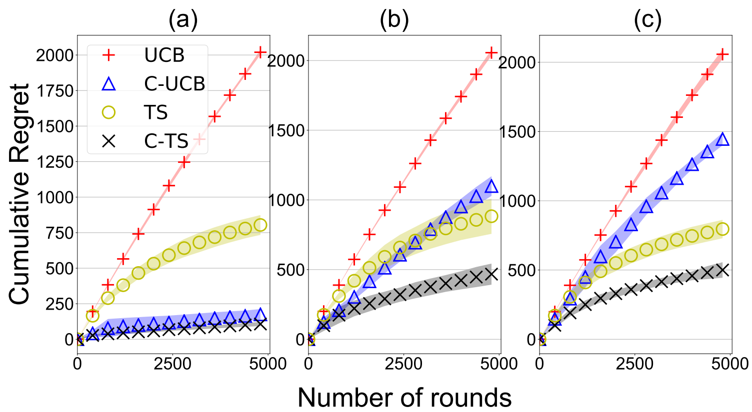


Figure 3.13: Cumulative regret for UCB, C-UCB, TS and C-TS for the application of recommending the best genre in the Movielens dataset, where p fraction of the pseudo-entries are replaced with maximum reward *i.e.*, 5. In (a), $p = 0.25$, for (b), $p = 0.50$ and $p = 0.7$ in (c). We used 10% of the dataset to learn the pseudo-reward entry and the algorithms are tested on the remaining dataset. The value of C is $5, 11$ and 15 in (a), (b) and (c) respectively. As C is smaller than K (i.e., 18) in each case, we see that C-UCB and C-TS outperform UCB and TS significantly. Note that the value of C is larger in the case where only 10% data is used for learning the pseudo-reward.

scenario (a), 10% of the pseudo-reward entries are replaced by 5 and remaining are padded with a safety buffer of 0.1. For case (b), 30% entries are replaced by 5 and safety buffer is 0.1. Under both cases, our proposed C-UCB and C-TS algorithms are able to outperform UCB and TS significantly.

3.6.3 Pseudo-rewards learned on a smaller dataset

In our previous set of experiments, half of the dataset was used to learn the pseudo-reward entries. We did additional experiments in a setup where only 10% of the data was used for learning the pseudo-reward entries and tested our algorithms on the remaining dataset. On doing so, we observed that C-UCB and C-TS were still able to outperform UCB and TS in most of our experimental setups. One setting in which the performance of C-UCB was similar to that of UCB is in a scenario where each pseudo-reward entry was padded by 0.6. As the padding was large, the C-UCB algorithm was not able to identify many arms as non-competitive, leading to a performance that is similar to that of UCB. In all other scenarios, we noted that C-UCB and C-TS significantly outperformed UCB and TS, suggesting that even when smaller dataset is used for learning pseudo-rewards, the C-UCB and C-TS can be quite effective. The results are presented in Figure 3.13, Figure 3.14 and Figure 3.15.

3.7 Concluding remarks

This work presents a new correlated Multi-Armed bandit problem in which rewards obtained from different arms are correlated. We capture this correlation through the knowledge of *pseudo-rewards*. These pseudo-

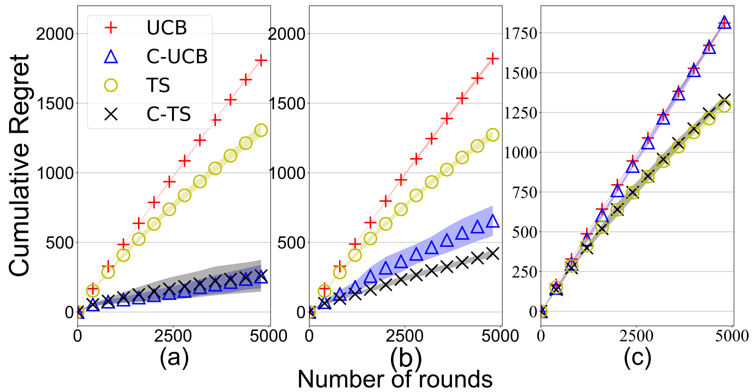


Figure 3.14: Cumulative regret of UCB, C-UCB, TS and C-TS for providing the best movie recommendations in the Movielens dataset. In this experiment 10% of the dataset is used for learning the pseudo-reward entry and the algorithms are tested on the remaining dataset. Each pseudo-reward entry is added by 0.1 in (a), 0.4 in (b) and 0.6 in (c). The value of C is 14, 29 and 42 in $(a), (b)$ and (c) respectively. Note that the value of C is larger in the case where only 10% data is used for learning the pseudo-reward. As C is still smaller than K (i.e., 50) in each case, we see the superior performance of C-UCB, C-TS over UCB and TS.

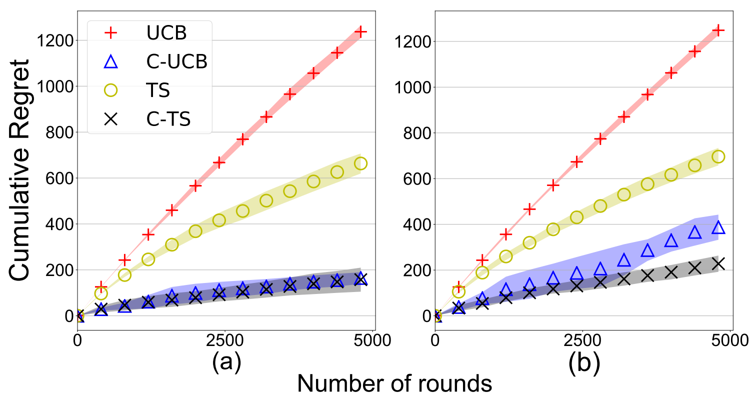


Figure 3.15: Cumulative regret of UCB, C-UCB, TS and C-TS for providing best poetry book recommendation in the Goodreads dataset. We used 10% of the dataset to learn the pseudo-reward entry and the algorithms are tested on the remaining dataset. Every pseudo-reward entry is added by q and p fraction of the pseudo-reward entries are removed, with (a) $p = 0.1, q = 0.1$ and (b) $p = 0.3, q = 0.1$. The value of C is 7 and 12 in (a) and (b) respectively. As C is much smaller than K (i.e., 25) in each case, we see that C-UCB and C-TS outperform UCB and TS significantly.

rewards, which represent upper bound on conditional mean rewards, could be known in practice from either domain knowledge or learned from prior data. Using the knowledge of these pseudo-rewards, we the propose *C-Bandit* algorithm which fundamentally generalizes any classical bandit algorithm to the correlated multi-armed bandit setting. A key strength of our work is that it allows pseudo-rewards to be loose (in case there is not much prior information) and even then our *C-Bandit* algorithms adapt and provide performance at least as good as that of classical bandit algorithms.

We provide a unified method to analyze the regret of C-Bandit algorithms. In particular, the analysis shows that C-UCB ends up pulling *non-competitive* arms only $O(1)$ times; i.e., they stop pulling certain arms after a finite time t. Due to this, C-UCB pulls only $C - 1 \leq K - 1$ of the $K - 1$ sub-optimal arms $O(\log T)$

times, as opposed to UCB that pulls *all* $K-1$ sub-optimal arms $O(\log T)$ times. In this sense, our C-Bandit algorithms reduce a K-armed bandit to a C-armed bandit problem. We present several cases where $C = 1$ for which C-UCB achieves bounded regret. For the special case when rewards are correlated through a latent random variable X, we provide a lower bound showing that bounded regret is possible only when $C = 1$; if $C > 1$, then $O(\log T)$ regret is not possible to avoid. Thus, our C-UCB algorithm achieves bounded regret whenever possible. Simulation results validate the theoretical findings and we perform experiments on MOVIELENS and GOODREADS datasets to demonstrate the applicability of our framework in the context of recommendation systems. The experiments on real-world datasets show that our C-UCB and C-TS algorithms significantly outperform the UCB and TS algorithms.

There are several interesting open problems and extensions of this work, some of which we describe below.

Extension to light tailed and heavy tailed rewards In this work, we assume that the rewards have a bounded support. The algorithm and analysis can be extended to settings with sub-gaussian rewards as well. In particular, in step 3 of the algorithm, one would play UCB/TS for sub-gaussian rewards. For instance, the UCB index in the scenario of sub-gaussian rewards can be redefined as $\hat{\mu}_k + \sqrt{\frac{2\sigma^2 \log t}{n_k(t)}}$, where σ is the sub-Gaussianity parameter of the reward distribution. Similar regret bounds will hold in this setting as well because the Hoeffding's inequality used in our regret analysis is valid for sub-Gaussian rewards as well. For heavy-tailed rewards, the Hoeffding's inequality is not valid. Due to which, one would need to construct confidence bounds for UCB in a different manner as done in [63]. On doing so, the C-Bandit algorithm can be employed in heavy-tailed reward settings. However, the regret analysis may not extend directly as one would need to use modified concentration inequalities to obtain bounds on mean reward of arm k as done in Lemma 1 of [63].

Designing better algorithms. While our proposed algorithms are order-optimal for the model in Section 2.3, they do not match the pre-log constants in the lower bound of the regret. It may be possible to design algorithms that have smaller pre-log constants in their regret upper bound. Further discussion along these lines is presented in Section 3.8.6. A key advantage of our approach is that our algorithms are easy to implement and they incorporate the classical bandit algorithms nicely for the problem of correlated multi-armed bandits.

Best-Arm Identification. We plan to study the problem of best-arm identification in the correlated multi-armed bandit setting, i.e., to identify the best arm with a confidence $1-\delta$ in as few samples as possible. Since rewards are correlated with each other, we believe the sample complexity can be significantly improved relative to state of the art algorithms, such as LIL-UCB [31, 40], which are designed for classical multi-armed bandits. Another open direction is to improve the C-Bandit algorithm to make sure that it achieves bounded regret whenever possible in the general framework studied in this chapter.

3.8 Full proofs

3.8.1 Standard Results from Previous Works

Fact 2 (Hoeffding's inequality). *Let $Z_1, Z_2 \ldots Z_n$ be i.i.d random variables bounded between $[a, b] : a \leq Z_i \leq b$, then for any $\delta > 0$, we have*

$$\Pr\left(\left|\frac{\sum_{i=1}^{n} Z_i}{n} - \mathbb{E}\left[Z_i\right]\right| \geq \delta\right) \leq \exp\left(\frac{-2n\delta^2}{(b-a)^2}\right).$$

Lemma 8 (Standard result used in bandit literature). *If $\hat{\mu}_{k,n_k(t)}$ denotes the empirical mean of arm k by pulling arm k $n_k(t)$ times through any algorithm and μ_k denotes the mean reward of arm k, then we have*

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) \leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right).$$

Proof. Let $Z_1, Z_2, \ldots Z_t$ be the reward samples of arm k drawn separately. If the algorithm chooses to play arm k for m^{th} time, then it observes reward Z_m. Then the probability of observing the event $\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1$ can be upper bounded as follows,

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) = \Pr\left(\left(\frac{\sum_{i=1}^{n_k(t)} Z_i}{n_k(t)} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{3.22}$$

$$\leq \Pr\left(\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{3.23}$$

$$\leq \Pr\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right) \tag{3.24}$$

$$\leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right). \tag{3.25}$$

□

Lemma 9 (From Proof of Theorem 1 in [32]). *Let $I_k(t)$ denote the UCB index of arm k at round t, and $\mu_k = \mathbb{E}\left[g_k(X)\right]$ denote the mean reward of that arm. Then, we have*

$$\Pr(\mu_k > I_k(t)) \leq t^{-3}.$$

Observe that this bound does not depend on the number $n_k(t)$ of times arm k is pulled.

Proof. This proof follows directly from [32]. We present the proof here for completeness as we use this frequently in the chapter.

$$\Pr(\mu_k > I_k(t)) = \Pr\left(\mu_k > \hat{\mu}_{k,n_k(t)} + \sqrt{\frac{2\log t}{n_k(t)}}\right) \tag{3.26}$$

$$\leq \sum_{m=1}^{t} \Pr\left(\mu_k > \hat{\mu}_{k,m} + \sqrt{\frac{2\log t}{m}}\right) \tag{3.27}$$

$$= \sum_{m=1}^{t} \Pr\left(\hat{\mu}_{k,m} - \mu_k < -\sqrt{\frac{2\log t}{m}}\right) \tag{3.28}$$

$$\leq \sum_{m=1}^{t} \exp\left(-2m\frac{2\log t}{m}\right) \tag{3.29}$$

$$= \sum_{m=1}^{t} t^{-4} \tag{3.30}$$

$$= t^{-3}. \tag{3.31}$$

where (3.27) follows from the union bound and is a standard trick (Lemma 8) to deal with random variable $n_k(t)$. We use this trick repeatedly in the proofs. We have (3.29) from the Hoeffding's inequality. □

Lemma 10. *Let $\mathbb{E}\left[\mathbb{1}_{I_k > I_{k^*}}\right]$ be the expected number of times $I_k(t) > I_{k^*}(t)$ in T rounds. Then, we have*

$$\mathbb{E}\left[\mathbb{1}_{I_k > I_{k^*}}\right] = \sum_{t=1}^{T} Pr(I_k > I_{k^*}) \leq \frac{8\log(T)}{\Delta_k^2} + \left(1 + \frac{\pi^2}{3}\right).$$

The proof follows the analysis in Theorem 1 of [32]. The analysis of $Pr(I_k > I_{k^*})$ is done by by evaluating the joint probability $\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq \frac{8\log T}{\Delta_k^2}\right)$. Authors in [32] show that the probability of pulling arm k jointly with the event that it has had at-least $\frac{8\log T}{\Delta_k^2}$ pulls decays down with t, i.e., $\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq \frac{8\log T}{\Delta_k^2}\right) \leq t^{-2}$.

Lemma 11 (Theorem 2 [1])**.** *Consider a two armed bandit problem with reward distributions $\Theta = \{f_{R_1}(r), f_{R_2}(r)\}$, where the reward distribution of the optimal arm is $f_{R_1}(r)$ and for the sub-optimal arm is $f_{R_2}(r)$, and $\mathbb{E}\left[f_{R_1}(r)\right] > \mathbb{E}\left[f_{R_2}(r)\right]$; i.e., arm 1 is optimal. If it is possible to create an alternate problem with distributions $\Theta' = \{f_{R_1}(r), \tilde{f}_{R_2}(r)\}$ such that $\mathbb{E}\left[\tilde{f}_{R_2}(r)\right] > \mathbb{E}\left[f_{R_1}(r)\right]$ and $0 < D(f_{R_2}(r)||\tilde{f}_{R_2}(r)) < \infty$ (equivalent to assumption 1.6 in [1]), then for any policy that achieves sub-polynomial regret, we have*

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[n_2(T)\right]}{\log T} \geq \frac{1}{D(f_{R_2}(r)||\tilde{f}_{R_2}(r))}.$$

Proof. Proof of this is derived from the analysis done in [64]. We show the analysis here for completeness. A bandit instance v is defined by the reward distribution of arm 1 and arm 2. Since policy π achieves sub-polynomial regret, for any instance v, $\mathbb{E}_{v,\pi}\left[(Reg(T))\right] = O(T^p)$ as $T \to \infty$, for all $p > 0$. Consider the bandit instances $\Theta = \{f_{R_1}(r), f_{R_2}(r)\}$, $\Theta' = \{f_{R_1}(r), \tilde{f}_{R_2}(r)\}$, where $\mathbb{E}\left[f_{R_2}(r)\right] < \mathbb{E}\left[f_{R_1}(r)\right] < \mathbb{E}\left[\tilde{f}_{R_2}(r)\right]$. The bandit instance Θ' is constructed by changing the reward distribution of arm 2 in the original instance, in such a way that arm 2 becomes optimal in instance Θ' without changing the reward distribution of arm 1 from the original instance.

From divergence decomposition lemma (derived in [64]), it follows that

$$D(\mathbb{P}_{\Theta,\Pi}||\mathbb{P}_{\Theta',\Pi}) = \mathbb{E}_{\Theta,\pi}\left[n_2(T)\right] D(f_{R_2}(r)||\tilde{f}_{R_2}(r)).$$

The high probability Pinsker's inequality (Lemma 2.6 from [65], originally in [66]) gives that for any event A,

$$\mathbb{P}_{\Theta,\pi}(A) + \mathbb{P}_{\Theta',\pi}(A^c) \geq \frac{1}{2}\exp\left(-D(\mathbb{P}_{\Theta,\pi}||\mathbb{P}_{\Theta',\pi})\right),$$

or equivalently,

$$D(\mathbb{P}_{\Theta,\pi}||\mathbb{P}_{\Theta',\pi}) \geq \log\frac{1}{2(\mathbb{P}_{\Theta,\pi}(A) + \mathbb{P}_{\Theta',\pi}(A^c))}.$$

If arm 2 is suboptimal in a 2-armed bandit problem, then $\mathbb{E}\left[Reg(T)\right] = \Delta_2\mathbb{E}\left[n_2(T)\right]$. Expected regret in Θ is

$$\mathbb{E}_{\Theta,\pi}\left[Reg(T)\right] \geq \frac{T\Delta_2}{2}\mathbb{P}_{\Theta,\pi}\left(n_2(T) \geq \frac{T}{2}\right),$$

Similarly regret in bandit instance Θ' is

$$\mathbb{E}_{\Theta',\pi}\left[Reg(T)\right] \geq \frac{T\delta}{2}\mathbb{P}_{\Theta',\pi}\left(n_2(T) < \frac{T}{2}\right),$$

since suboptimality gap of arm 1 in Θ' is δ. Define $\kappa(\Delta_2,\delta) = \frac{\min(\Delta_2,\delta)}{2}$. Then we have,

$$\mathbb{P}_{\Theta,\pi}\left(n_2(T) \geq \frac{T}{2}\right) + \mathbb{P}_{\Theta',\pi}\left(n_2(T) < \frac{T}{2}\right) \leq \frac{\mathbb{E}_{\Theta,\pi}\left[Reg(T)\right] + \mathbb{E}_{\Theta',\pi}\left[Reg(T)\right]}{\kappa(\Delta_2,\delta)T}.$$

On applying the high probability Pinsker's inequality and divergence decomposition lemma stated earlier, we get

$$\begin{aligned} D(f_{R_2}(r)||\tilde{f}_{R_2}(r))\mathbb{E}_{\Theta,\pi}\left[n_2(T)\right] &\geq \log\left(\frac{\kappa(\Delta_2,\delta)T}{2(\mathbb{E}_{\Theta,\pi}\left[Reg(T)\right] + \mathbb{E}_{\Theta',\pi}\left[Reg(T)\right])}\right) && (3.32)\\ &= \log\left(\frac{\kappa(\Delta_2,\delta)}{2}\right) + \log(T) \\ &\quad - \log(\mathbb{E}_{\Theta,\pi}\left[Reg(T)\right] + \mathbb{E}_{\Theta',\pi}\left[Reg(T)\right]). && (3.33)\end{aligned}$$

Since policy π achieves sub-polynomial regret for any bandit instance, $\mathbb{E}_{\Theta,\pi}[Reg(T)] + \mathbb{E}_{\Theta',\pi}[Reg(T)] \leq \gamma T^p$ for all T and any $p > 0$, hence,

$$\liminf_{T\to\infty} D(f_{R_2}(r)||\tilde{f}_{R_2}(r)) \frac{\mathbb{E}_{\Theta,\pi}[n_2(T)]}{\log T} \geq 1 - \limsup_{T\to\infty} \frac{\mathbb{E}_{\Theta,\pi}[Reg(T)] + \mathbb{E}_{\Theta',\pi}[Reg(T)]}{\log T} + \liminf_{T\to\infty} \frac{\log\left(\frac{\kappa(\Delta_2,\delta)}{2}\right)}{\log T} \tag{3.34}$$

$$= 1. \tag{3.35}$$

Hence, $\liminf_{T\to\infty} \frac{\mathbb{E}_{\Theta,\pi}[n_2(T)]}{\log T} \geq \frac{1}{D(f_{R_2}(r)||\tilde{f}_{R_2}(r))}$.

□

3.8.2 Results for any C-Bandit Algorithm

Lemma 12. *Define $E_1(t)$ to be the event that arm k^* is empirically non-competitive in round $t+1$, then,*

$$\Pr(E_1(t)) \leq 2Kt \exp\left(\frac{-t\Delta_{min}^2}{2K}\right),$$

where $\Delta_{min} = \min_k \Delta_k$, the gap between the best and second-best arms.

Proof. The arm k^* is empirically non-competitive at round t if $k^* \neq k^{\text{emp}}$ and the empirical psuedo-reward of arm k^* with respect to arms $\ell \in \mathcal{S}_t$ is smaller than $\hat{\mu}_{k^{\text{emp}}}(t)$. This event can only occur if at-least one of the two following conditions is satisfied, i) the empirical mean of $k^{\text{emp}} \neq k^*$ is greater than $\mu_k^* - \frac{\Delta_{\min}}{2}$ or ii) the empirical pseudo-reward of arm k^* with respect to arms in $\mathcal{S}_t$ is smaller than $\mu_{k^*} - \frac{\Delta_{\min}}{2}$. We use this observation to analyse the $\Pr(E_1(t))$.

$$\Pr(E_1(t)) \leq \Pr\left(\left(\max_{\{\ell: n_\ell(t) > t/K, \ell \neq k^*\}} \hat{\mu}_\ell(t) > \mu_{k^*} - \frac{\Delta_{\min}}{2}\right) \bigcup \left(\min_{\{\ell: n_\ell(t) > t/K\}} \hat{\phi}_{k^*,\ell}(t) < \mu_{k^*} - \frac{\Delta_{\min}}{2}\right)\right) \tag{3.36}$$

$$\leq \Pr\left(\max_{\{\ell: n_\ell(t) > t/K, \ell \neq k^*\}} \hat{\mu}_\ell(t) > \mu_{k^*} - \frac{\Delta_{\min}}{2}\right) + \Pr\left(\min_{\{\ell: n_\ell(t) > t/K\}} \hat{\phi}_{k^*,\ell}(t) < \mu_{k^*} - \frac{\Delta_{\min}}{2}\right) \tag{3.37}$$

$$\leq \sum_{\ell \neq k^*} \Pr\left(\hat{\mu}_\ell(t) > \mu_{k^*} - \frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right) + \sum_{\ell=1}^{K} \Pr\left(\hat{\phi}_{k^*,\ell}(t) < \mu_{k^*} - \frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right) \tag{3.38}$$

$$= \sum_{\ell \neq k^*} \Pr\left(\hat{\mu}_\ell(t) - \mu_\ell > \mu_{k^*} - \mu_\ell - \frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right)$$
$$+ \sum_{\ell=1}^{K} \Pr\left(\hat{\phi}_{k^*,\ell}(t) - \phi_{k^*,\ell} < \mu_{k^*} - \phi_{k^*,\ell} - \frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right) \tag{3.39}$$

$$\leq \sum_{\ell \neq k^*} \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = \ell\}} r_\tau}{n_\ell(t)} - \mu_\ell > \frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right)$$
$$+ \sum_{\ell=1}^{K} \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = \ell\}} s_{k^*,\ell}(r_\tau)}{n_\ell(t)} - \phi_{k^*,\ell} < -\frac{\Delta_{\min}}{2}, n_\ell(t) > \frac{t}{K}\right) \tag{3.40}$$

$$\leq 2Kt \exp\left(\frac{-t\Delta_{\min}^2}{2K}\right), \tag{3.41}$$

Here (3.37) follows from union bound. We have (3.41) from the Hoeffding's inequality, as we note that rewards $\{r_\tau : \tau = 1, \ldots, t,\ k_\tau = k\}$ and pseudo-rewards $\{s_{k^*,l} : \tau_1, \ldots, t,\ k_\tau = l\}$ form a collection of i.i.d. random variables each of which is bounded between $[-1, 1]$ with mean μ_k and $\phi_{k^*,l}$. The term t before the exponent in (3.41) arises as the random variable $n_k(t)$ can take values from t/K to t (Lemma 8).

□

Lemma 13. *For a sub-optimal arm $k \neq k^*$ with sub-optimality gap Δ_k,*

$$\Pr\left(k = k^{emp}(t), n_{k^*}(t) \geq \frac{t}{K}\right) \leq t \exp\left(\frac{-t\Delta_k^2}{2K}\right).$$

Proof. We bound this probability as,

$$
\begin{aligned}
&\Pr\left(k = k^{\text{emp}}(t), n_{k^*}(t) \geq \frac{t}{K}\right) \\
&= \Pr\left(k = k^{\text{emp}}(t), n_{k^*}(t) \geq \frac{t}{K}, n_k(t) \geq \frac{t}{K}\right) && (3.42) \\
&\leq \Pr\left(\hat{\mu}_k(t) \geq \hat{\mu}_{k^*}(t), n_k(t) \geq \frac{t}{K}, n_{k^*}(t) \geq \frac{t}{K}\right) && (3.43) \\
&\leq \Pr\left(\left(\hat{\mu}_{k^*}(t) < \mu_{k^*} - \frac{\Delta_k}{2} \bigcup \hat{\mu}_k(t) > \mu_{k^*} - \frac{\Delta_k}{2}\right), n_k(t) \geq \frac{t}{K}, n_{k^*}(t) \geq \frac{t}{K}\right) && (3.44) \\
&= \Pr\left(\left(\hat{\mu}_{k^*}(t) < \mu_{k^*} - \frac{\Delta_k}{2} \bigcup \hat{\mu}_k(t) > \mu_k + \frac{\Delta_k}{2}\right), n_k(t) \geq \frac{t}{K}, n_{k^*}(t) \geq \frac{t}{K}\right) && (3.45) \\
&\leq \Pr\left(\hat{\mu}_{k^*}(t) - \mu_{k^*} < -\frac{\Delta_k}{2}, n_{k^*}(t) \geq \frac{t}{K}\right) + \Pr\left(\hat{\mu}_k(t) - \mu_k > \frac{\Delta_k}{2}, n_k(t) \geq \frac{t}{K}\right) && (3.46) \\
&\leq 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.47)
\end{aligned}
$$

We have (3.42) as arm k needs to be pulled at least $\frac{t}{K}$ in order to be arm $k^{\text{emp}}(t)$ at round t. The selection of k^{emp} is only done from the set of arms that have been pulled atleast $\frac{t}{K}$ times. Here, (3.47) follows from the Hoeffding's inequality. The term t before the exponent in (3.47) arises as the random variable $n_k(t)$ can take values from t/K to t (Lemma 8).

□

Lemma 14. *If for a suboptimal arm $k \neq k^*$, $\tilde{\Delta}_{k,k^*} > 0$, then,*

$$\Pr(k_{t+1} = k, n_{k^*}(t) = \max_k n_k(t)) \leq t \exp\left(\frac{-2t\tilde{\Delta}_{k,k^*}^2}{K}\right).$$

Moreover, if $\tilde{\Delta}_{k,k^*} \geq \sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$. Then,

$$\Pr(k_{t+1} = k, n_{k^*}(t) = \max_k n_k(t)) \leq t^{-3} \quad \forall t > t_0.$$

Proof. We now bound this probability as,

$$
\begin{aligned}
&\Pr(k_{t+1} = k, n_{k^*}(t) = \max_k n_k(t)) \\
&\leq \Pr\left(k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) && (3.48) \\
&= \Pr\left(k \in \{\mathcal{A}_t \cup \{k_{\text{emp}}(t)\}\}, k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) && (3.49) \\
&\leq \Pr\left(k \in \mathcal{A}_t, k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) + \Pr\left(k = k^{\text{emp}}(t), n_{k^*}(t) \geq \frac{t}{K}\right) && (3.50) \\
&\leq \Pr\left(k \in \mathcal{A}_t, k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.51) \\
&\leq \Pr\left(\hat{\mu}_{k^*}(t) < \hat{\phi}_{k,k^*}(t), k_{t+1} = k, n_{k^*}(t) \geq \frac{t}{K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.52) \\
&\leq \Pr\left(\hat{\mu}_{k^*}(t) < \hat{\phi}_{k,k^*}(t), n_{k^*}(t) \geq \frac{t}{K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.53) \\
&\leq \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = k^*\}} r_\tau}{n_{k^*}(t)} < \frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = k^*\}} s_{k,k^*}(r_\tau)}{n_{k^*(t)}}, n_{k^*}(t) \geq \frac{t}{K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.54) \\
&= \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = k^*\}} (r_\tau - s_{k,k^*})}{n_{k^*}(t)} - (\mu_{k^*} - \phi_{k,k^*}) < -\tilde{\Delta}_{k,k^*}, n_{k^*} \geq \frac{t}{K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.55) \\
&\leq t \exp\left(\frac{-t\tilde{\Delta}_{k,k^*}^2}{2K}\right) + 2t \exp\left(\frac{-t\Delta_k^2}{2K}\right) && (3.56) \\
&\leq 3t^{-3} \quad \forall t > t_0. && (3.57)
\end{aligned}
$$

We have (3.48) as $n_{k^*}(t)$ needs to be at-least $\frac{t}{K}$ for $n_{k^*}(t)$ to be $\max_k n_k(t)$. Equation (3.49) holds as arm k needs to be in the set $\{\mathcal{A}_t \cup \{k^{\text{emp}}(t)\}\}$ to be selected by C-BANDIT at round t. Inequality (3.51) arises from the result of Lemma 13. The inequality (3.52) follows as $\phi_{k,k^*} > \hat{\mu}_{k^*}$ is a necessary condition for arm k to be in the competitive set $\mathcal{A}_t$ at round t. Here, (3.55) follows from the Hoeffding's inequality as we note that rewards $\{r_\tau - s_{k,k^*}(r_\tau) : \tau = 1, \ldots, t,\ k_\tau = k^*\}$ form a collection of i.i.d. random variables each of which is bounded between $[-1, 1]$ with mean $(\mu_k - \phi_{k,k^*})$. The term t before the exponent in (3.55) arises as the random variable $n_k(t)$ can take values from t/K to t (Lemma 8). Step (3.57) follows from the fact that $\tilde{\Delta}_{k,k^*} \geq 2\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$. □

3.8.3 Algorithm specific results for C-UCB

Lemma 15. *If $\Delta_{min} \geq 4\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\Pr(k_{t+1} = k, n_k(t) \geq s) \leq 3t^{-3} \quad \textit{for } s > \frac{t}{2K}, \forall t > t_0.$$

Proof. By noting that $k_{t+1} = k$ corresponds to arm k having the highest index among the set of arms that are not empirically *non-competitive* (denoted by $\mathcal{A}$), we have,

$$\Pr(k_{t+1} = k, n_k(t) \geq s) = \Pr(I_k(t) = \arg\max_{k' \in \mathcal{A}} I_{k'}(t), n_k(t) \geq s) \tag{3.58}$$

$$\leq \Pr(E_1(t) \cup (E_1^c(t), I_k(t) > I_{k^*}(t)), n_k(t) \geq s) \tag{3.59}$$

$$\leq \Pr(E_1(t), n_k(t) \geq s) + \Pr(E_1^c(t), I_k(t) > I_{k^*}(t), n_k(t) \geq s) \tag{3.60}$$

$$\leq 2Kt \exp\left(\frac{-t\Delta_{\min}^2}{2K}\right) + \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s\right). \tag{3.61}$$

Here $E_1(t)$ is the event described in Lemma 12. If arm k^* is not empirically non-competitive at round t, then arm k can only be pulled in round $t+1$ if $I_k(t) > I_{k^*}(t)$, due to which we have (3.59). Inequalities (3.60) and (3.61) follow from union bound and Lemma 12 respectively.

We now bound the second term in (3.61).

$$\Pr(I_k(t) > I_{k^*}(t), n_k(t) \geq s) =$$

$$\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) +$$

$$\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s | \mu_{k^*} > I_{k^*}(t)\right) \times \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \tag{3.62}$$

$$\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \tag{3.63}$$

$$\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + t^{-3} \tag{3.64}$$

$$= \Pr\left(I_k(t) > \mu_{k^*}, n_k(t) \geq s\right) + t^{-4} \tag{3.65}$$

$$= \Pr\left(\hat{\mu}_k(t) + \sqrt{\frac{2\log t}{n_k(t)}} > \mu_{k^*}, n_k(t) \geq s\right) + t^{-3} \tag{3.66}$$

$$= \Pr\left(\hat{\mu}_k(t) - \mu_k > \mu_{k^*} - \mu_k - \sqrt{\frac{2\log t}{n_k(t)}}, n_k(t) \geq s\right) + t^{-3} \tag{3.67}$$

$$= \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = k\}} r_\tau}{n_k(t)} - \mu_k > \Delta_k - \sqrt{\frac{2\log t}{n_k(t)}}, n_k(t) \geq s\right) + t^{-3} \tag{3.68}$$

$$\leq t \exp\left(-2s\left(\Delta_k - \sqrt{\frac{2\log t}{s}}\right)^2\right) + t^{-3} \tag{3.69}$$

$$\leq t^{-3} \exp\left(-2s\left(\Delta_k^2 - 2\Delta_k\sqrt{\frac{2\log t}{s}}\right)\right) + t^{-3} \tag{3.70}$$

$$\leq 2t^{-3} \quad \text{for all } t > t_0. \tag{3.71}$$

We have (3.62) holds because of the fact that $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$, Inequality (3.64) follows from Lemma 9. From the definition of $I_k(t)$ we have (3.66). Inequality (3.69) follows from Hoeffding's inequality and the term t before the exponent in (3.69) arises as the random variable $n_k(t)$ can take values from s to t (Lemma 8). Inequality (3.71) follows from the fact that $s > \frac{t}{2K}$ and $\Delta_k \geq 4\sqrt{\frac{2K\log t_0}{t_0}}$ for some constant $t_0 > 0$.

Plugging this in the expression of $\Pr(k_t = k, n_k(t) \geq s)$ (3.61) gives us,

$$\Pr(k_{t+1} = k, n_k(t) \geq s) \leq 2Kt\exp\left(\frac{-t\Delta_{\min}^2}{2K}\right) + \Pr(I_k(t) > I_{k^*}(t), n_k(t) \geq s) \tag{3.72}$$

$$\leq 2Kt\exp\left(\frac{-t\Delta_{\min}^2}{2K}\right) + 2t^{-3} \tag{3.73}$$

$$\leq 2(K+1)t^{-3}. \tag{3.74}$$

Here, (3.74) follows from the fact that $\Delta_{\min} \geq 4\sqrt{\frac{2K\log t_0}{t_0}}$ for some constant $t_0 > 0$. □

Lemma 16. *If $\Delta_{min} \geq 4\sqrt{\frac{2K\log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\Pr\left(n_k(t) > \frac{t}{K}\right) \leq (2K+2)K\left(\frac{t}{K}\right)^{-2} \quad \forall t > Kt_0.$$

Proof. We expand $\Pr\left(n_k(t) > \frac{t}{K}\right)$ as,

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) = \Pr\left(n_k(t) \geq \frac{t}{K} \mid n_k(t-1) \geq \frac{t}{K}\right)\Pr\left(n_k(t-1) \geq \frac{t}{K}\right) + \Pr\left(k_t = k, n_k(t-1) = \frac{t}{K} - 1\right) \tag{3.75}$$

$$\leq \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) + \Pr\left(k_t = k, n_k(t-1) = \frac{t}{K} - 1\right) \tag{3.76}$$

$$\leq \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) + (2K+2)(t-1)^{-3} \quad \forall (t-1) > t_0. \tag{3.77}$$

Here, (3.77) follows from Lemma 15. This gives us

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) - \Pr\left(n_k(t-1) \geq \frac{t}{K}\right) \leq (2K+2)(t-1)^{-3}, \quad \forall (t-1) > t_0.$$

Now consider the summation

$$\sum_{\tau=\frac{t}{K}}^{t} \Pr\left(n_k(\tau) \geq \frac{t}{K}\right) - \Pr\left(n_k(\tau-1) \geq \frac{t}{K}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3}.$$

This gives us,

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) - \Pr\left(n_k\left(\frac{t}{K} - 1\right) \geq \frac{t}{K}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3}.$$

Since $\Pr\left(n_k\left(\frac{t}{K}-1\right) \geq \frac{t}{K}\right) = 0$, we have,

$$\Pr\left(n_k(t) \geq \frac{t}{K}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3} \tag{3.78}$$

$$\leq (2K+2)K\left(\frac{t}{K}\right)^{-2} \quad \forall t > Kt_0. \tag{3.79}$$

□

3.8.4 Regret Bounds for C-UCB

Proof of Theorem 5 We bound $\mathbb{E}\left[n_k(T)\right]$ as,

$$\mathbb{E}\left[n_k(T)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{k_t=k\}}\right] \tag{3.80}$$

$$= \sum_{t=0}^{T-1} \Pr(k_{t+1}=k) \tag{3.81}$$

$$= \sum_{t=1}^{Kt_0} \Pr(k_t=k) + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k) \tag{3.82}$$

$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k, n_{k^*}(t) = \max_{k'} n_{k'}(t))$$
$$+ \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \Pr(n_{k'}(t) = \max_{k''} n_{k''}(t)) \Pr(k_{t+1}=k | n_{k'}(t) = \max_{k''} n_{k''}(t)) \tag{3.83}$$

$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} \Pr(k_{t+1}=k, n_{k^*}(t) = \max_{k'} n_{k'}(t))$$
$$+ \sum_{t=Kt_0}^{T-1} \sum_{k' \neq k^*} \Pr(n_{k'}(t) = \max_{k''} n_{k''}(t)) \tag{3.84}$$

$$\leq Kt_0 + \sum_{t=Kt_0}^{T-1} 3t^{-3} + \sum_{t=Kt_0}^{T} \sum_{k' \neq k^*} \Pr\left(n_{k'}(t) \geq \frac{t}{K}\right) \tag{3.85}$$

$$\leq Kt_0 + \sum_{t=1}^{T} 3t^{-3} + (K+1)K(K-1) \sum_{t=Kt_0}^{T} 2\left(\frac{t}{K}\right)^{-2}. \tag{3.86}$$

Here, (3.85) follows from Lemma 14 and (3.86) follows from Lemma 16.

Proof of Theorem 6

For any suboptimal arm $k \neq k^*$,

$$\mathbb{E}\left[n_k(T)\right] \leq \sum_{t=1}^{T} \Pr(k_t = k) \tag{3.87}$$

$$= \sum_{t=1}^{T} \Pr(E_1(t), k_t = k \cup (E_1^c(t), I_k > I_{k^*}), k_t = k) \tag{3.88}$$

$$\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \Pr(E_1^c(t), I_k(t-1) > I_{k^*}(t-1), k_t = k) \tag{3.89}$$

$$\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \Pr(E_1^c(t), I_k(t-1) > I_{k^*}(t-1))$$

$$\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \Pr(I_k(t-1) > I_{k^*}(t-1), k_t = k) \tag{3.90}$$

$$= \sum_{t=1}^{T} 2Kt \exp\left(-\frac{t\Delta_{\min}^2}{2K}\right) + \sum_{t=0}^{T-1} \Pr\left(I_k(t) > I_{k^*}(t), k_t = k\right) \tag{3.91}$$

$$= \sum_{t=1}^{T} 2Kt \exp\left(-\frac{t\Delta_{\min}^2}{2K}\right) + \mathbb{E}\left[\mathbb{1}_{I_k > I_{k^*}}(T)\right] \tag{3.92}$$

$$\leq 8\frac{\log(T)}{\Delta_k^2} + \left(1 + \frac{\pi^2}{3}\right) + \sum_{t=1}^{T} 2Kt \exp\left(-\frac{t\Delta_{\min}^2}{2K}\right). \tag{3.93}$$

Here, (3.91) follows from Lemma 12. We have (3.92) from the definition of $\mathbb{E}\left[n_{I_k > I_{k^*}}(T)\right]$ in Lemma 10, and (3.93) follows from Lemma 10.

3.8.5 Regret analysis for the C-TS Algorithm

We now present results for C-TS in the scenario where $K = 2$ and Thompson sampling is employed with Beta priors [52]. In order to prove results for C-TS, we assume that rewards are either 0 or 1. The Thompson sampling algorithm with beta prior, maintains a posterior distribution on mean of arm k as $Beta\left(n_k(t) \times \hat{\mu}_k(t) + 1, n_k(t) \times (1 - \hat{\mu}_k(t)) + 1\right)$. Subsequently, it generates a sample $S_k(t) \sim Beta\left(n_k(t) \times \hat{\mu}_k(t) + 1, n_k(t) \times (1 - \hat{\mu}_k(t)) + 1\right)$ for each arm k and selects the arm $k_{t+1} = \arg\max_{k \in \mathcal{K}} S_k(t)$. The C-TS algorithm with Beta prior uses this Thompson sampling procedure in its last step, i.e., $k_{t+1} = \arg\max_{k \in \mathcal{C}_t} S_k(t)$, where $\mathcal{C}_t$ is the set of empirically competitive arms at round t. We show that in a 2-armed bandit problem, the regret is $O(1)$ if the sub-optimal arm k is non-competitive and is $O(\log T)$ otherwise.

For the purpose of regret analysis of C-TS, we define two thresholds, a lower threshold L_k, and an upper threshold U_k for arm $k \neq k^*$,

$$U_k = \mu_k + \frac{\Delta_k}{3}, \qquad L_k = \mu_{k^*} - \frac{\Delta_k}{3}. \tag{3.94}$$

Let $E_i^{\mu}(t)$ and $E_i^{S}(t)$ be the events that,

$$\begin{aligned} E_k^{\mu}(t) &= \{\hat{\mu}_k(t) \leq U_k\} \\ E_k^{S}(t) &= \{S_k(t) \leq L_k\}. \end{aligned} \tag{3.95}$$

To analyse the regret of C-TS, we first show that the number of times arm k is pulled jointly with the event that $n_k(t-1) \geq \frac{t}{2}$ is bounded above by an O(1) constant, which is independent of the total number of rounds T.

Lemma 17. *If $\Delta_k \geq 4\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\sum_{t=2t_0}^{T} \Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$$

where $k \neq k^*$ is a sub-optimal arm.

Proof. We start by bounding the probability of the pull of k-th arm at round t as follows,

$$\begin{aligned} \Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) &\overset{(a)}{\leq} \Pr\left(E_1(t), k_t = k, n_k(t-1) \geq \frac{t}{2}\right) + \\ &\quad \Pr\left(\overline{E_1(t)}, k_t = k, n_k(t-1) \geq \frac{t}{2}\right) \\ &\overset{(b)}{\leq} 2Kt \exp\left(\frac{-t\Delta_{\min}^2}{2K}\right) + \Pr\left(\overline{E_1(t)}, k_t = k, n_k(t-1) \geq \frac{t}{2}\right) \\ &\overset{(c)}{\leq} 2Kt^{-3} + \underbrace{\Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term A}} + \\ &\quad \underbrace{\Pr\left(k_t = k, E_k^{\mu}(t), \overline{E_k^{S}(t)}, n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term B}} + \\ &\quad \underbrace{\Pr\left(k_t = k, \overline{E_k^{\mu}(t)}, n_k(t-1) \geq \frac{t}{2}\right)}_{\textbf{term C}} \end{aligned} \tag{3.96}$$

where (b), comes from Lemma 12. Here, (3.96) follows from the fact that $\Delta_{\min} \geq 4\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$. Now we treat each term in (3.96) individually. To bound term A, we note that $\Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right) \leq \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right)$. From the analysis in [52] (equation 6), we see that $\sum_{t=1}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right) = \mathrm{O}(1)$ as it is shown through Lemma 2 in [52] that,
$\sum_{t=1}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t)\right) \leq \frac{216}{\Delta_k^2} + \sum_{j=0}^{T} \Theta\left(e^{-\frac{\Delta_k^2 j}{18}} + \frac{1}{e^{\frac{\Delta_k^2 j}{36}} - 1} + \frac{9}{(j+1)\Delta_k^2} e^{-D_k j}\right)$.
Here, $D_k = L_k \log \frac{L_k}{\mu_{k^*}} + (1 - L_k) \log \frac{1-L_k}{1-\mu_{k^*}}$. Due to this,
$\sum_{t=2t_0}^{T} \Pr\left(k_t = k, E_k^{\mu}(t), E_k^{S}(t), n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$.

We now bound the sum of term B from $t = 1$ to T by noting that $\Pr\left(k_t = k, E^{\mu}_k(t), \overline{E^S_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) \leq \Pr\left(k_t = k, \overline{E^S_k(t)}\right)$. Additionally, from Lemma 3 in [52], we get that $\sum_{t=1}^{T} \Pr\left(k_t = k, \overline{E^S_k(t)}\right) \leq \frac{1}{d(U_k,\mu_k)} + 1$, where $d(x, y) = x \log \frac{x}{y} + (1-x) \log \frac{1-x}{1-y}$. As a result, we see that $\sum_{t=1}^{T} \Pr\left(k_t = k, E^{\mu}_k(t), \overline{E^S_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) = \mathrm{O}(1)$.

Finally, for the last term C we can show that,

$$
\begin{aligned}
(\mathrm{C}) &= \Pr\left(k_t = k, \overline{E^{\mu}_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) \\
&\leq \Pr\left(\overline{E^{\mu}_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) \\
&= \Pr\left(\hat{\mu}_k - \mu_k > \frac{\Delta_k}{3}, n_k(t-1) \geq \frac{t}{2}\right) \\
&\leq t \exp\left(-2\frac{t}{2}\frac{\Delta_k^2}{9}\right) \qquad (3.97) \\
&\leq t^{-3}
\end{aligned}
$$

Here (3.97) follows from hoeffding's inequality and the union bound trick to handle random variable $n_k(t-1)$. After plugging these results in (3.96), we get that

$$
\begin{aligned}
\sum_{t=2t_0}^{T} \Pr\left(k_t = k, n_k(t-1) \geq \frac{t}{2}\right) &\leq \sum_{t=2t_0}^{T} 2Kt^{-3} + \sum_{t=2t_0}^{T} \Pr\left(k_t = k, E^{\mu}_k(t), E^S_k(t), n_k(t-1) \geq \frac{t}{2}\right) + \\
&\quad \sum_{t=2t_0}^{T} \Pr\left(k_t = k, E^{\mu}_k(t), \overline{E^S_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) + \\
&\quad \sum_{t=2t_0}^{T} \Pr\left(k_t = k, \overline{E^{\mu}_k(t)}, n_k(t-1) \geq \frac{t}{2}\right) \qquad (3.98) \\
&\leq \sum_{t=2t_0}^{T} 2Kt^{-3} + \mathrm{O}(1) + \mathrm{O}(1) + \sum_{t=2t_0}^{T} t^{-3} \qquad (3.99) \\
&= \mathrm{O}(1) \qquad (3.100)
\end{aligned}
$$

□

We now show that the expected number of pulls by C-TS for a non-competitive arm is bounded above by an $\mathrm{O}(1)$ constant.

Expected number of pulls by C-TS for a non-competitive arm. We bound $\mathbb{E}[n_k(t)]$ as

$$\begin{aligned}
\mathbb{E}[n_k(T)] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{k_t=k\}}\right] \\
&= \sum_{t=0}^{T-1} \Pr(k_{t+1}=k) \\
&= \sum_{t=1}^{2t_0} \Pr(k_t=k) + \sum_{t=2t_0}^{T-1} \Pr(k_{t+1}=k) \\
&\leq 2t_0 + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1}=k, n_{k^*}(t) \geq \frac{t}{2}\right) + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1}=k, n_k(t) \geq \frac{t}{2}\right) && (3.101) \\
&\leq 2t_0 + \sum_{t=2t_0}^{T-1} 3t^{-3} + \sum_{t=2t_0}^{T-1} \Pr\left(k_{t+1}=k, n_k(t) \geq \frac{t}{2}\right) && (3.102) \\
&= \mathrm{O}(1) && (3.103)
\end{aligned}$$

Here, (3.102) follows from Lemma 14 and (3.103) follows from Lemma 17 and the fact that the sum of $3t^{-3}$ is bounded and $t_0 = \inf\left\{\tau > 0 : \Delta_{\min}, \epsilon_k \geq 4\sqrt{\frac{2K\log\tau}{\tau}}\right\}$.

We now show that when the sub-optimal arm k is competitive, the expected pulls of arm k is $\mathrm{O}(\log T)$.

Expected number of pulls by C-TS for a competitive arm $k \neq k^*$.: For any sub-optimal arm $k \neq k^*$,

$$\begin{aligned}
\mathbb{E}[n_k(T)] &\leq \sum_{t=1}^{T} \Pr(k_t=k) \\
&= \sum_{t=1}^{T} \Pr((k_t=k, E_1(t)) \cup (E_1^c(t), k_t=k)) && (3.104) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t=k) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=1}^{T} \Pr(E_1^c(t), k_t=k, S_k(t-1) > S_{k^*}(t-1)) && (3.105) \\
&\leq \sum_{t=1}^{T} \Pr(E_1(t)) + \sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1}=k) \\
&= \sum_{t=1}^{T} 2Kt^{-3} + \sum_{t=0}^{T-1} \Pr\left(S_k(t) > S_{k^*}(t), k_{t+1}=k\right) && (3.106) \\
&\leq \frac{9\log(T)}{\Delta_k^2} + \mathrm{O}(1) + \sum_{t=1}^{T} 2Kt^{-3}. && (3.107) \\
&= \mathrm{O}(\log T). && (3.108)
\end{aligned}$$

Here, (3.106) follows from Lemma 12. We have (3.107) from the analysis of Thompson Sampling for the classical bandit problem in [52]. This arises as the term $\Pr\left(S_k(t) > S_{k^*}(t), k_{t+1}=k\right)$ counts the number of times $S_k(t) > S_{k^*}(t)$ and $k_{t+1}=k$. This is precisely the term analysed in Theorem 3 of [52] to bound the expected pulls of sub-optimal arms by TS. In particular, [52] analyzes the expected number of pull of sub-optimal arm (termed as $\mathbb{E}[k_i(T)]$ in their chapter) by evaluating $\sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1} = k)$ and it is shown in their Section 2.1 (proof of Theorem 1 of [52]) that $\sum_{t=0}^{T-1} \Pr(S_k(t) > S_{k^*}(t), k_{t+1} = k) \leq \mathrm{O}(1) + \frac{\log(T)}{d(x_i, y_i)}$. The term x_i is equivalent to U_k and y_i is equal to L_k in our notations. Moreover $d(U_k, L_k) \leq \frac{\Delta_k^2}{9}$, giving us the desired result of (3.107).

3.8.6 Lower Bounds

For the proof we define $R_k = Y_k(X)$ and $\tilde{R}_k = g_k(\tilde{X})$, where $f_X(x)$ is the probability density function of random variable X and $f_{\tilde{X}}(x)$ is the probability density function of random variable $\tilde{X}$. Similarly, we define $f_{R_k}(r)$ to be the reward distribution of arm k.

Proof of Theorem 7

Let arm k be a *Competitive* sub-optimal arm, i.e $\tilde{\Delta}_{k,k^*} < 0$. To prove that regret is $\Omega(\log T)$ in this setting, we need to create a new bandit instance, in which reward distribution of optimal arm is unaffected, but a previously competitive sub-optimal arm k becomes optimal in the new environment. We do so by constructing a bandit instance with latent randomness $\tilde{X}$ and random rewards $\tilde{Y}_k(X)$. Let's denote to $\tilde{Y}_k(\tilde{X})$ to be the random reward obtained on pulling arm k given the realization of $\tilde{X}$. To make arm k optimal in the new bandit instance, we construct $\tilde{Y}_k(X)$ and $\tilde{X}$ in the following manner. Let $\mathcal{Y}_k$ denote the support of $Y_k(X)$.

Define

$$\tilde{Y}_k(X) = \begin{cases} \bar{g}_k(X) \quad \text{w.p. } 1 - \epsilon_1 \\ \tilde{Y}_k(X) \sim \text{Uniform}(\mathcal{Y}_k) \quad \text{w.p. } \epsilon_1 \end{cases}$$

This changes the conditional reward of arm k in the new bandit instance (with increased mean).

Furthermore, Define

$$\tilde{X} = \begin{cases} S(R_{k^*}) \quad w.p. 1 - \epsilon_2 \\ \text{Uniform} \sim \mathcal{X} \quad w.p. \epsilon_2. \end{cases},$$

with $S(R_{k^*}) = \arg\max_{\underline{g}_{k^*}(x) < R_{k^*} < \bar{g}_{k^*}(x)} \bar{g}_k(x)$.

Here R_{k^*} represents the random reward of arm k^* in the original bandit instance.

This construction of $\tilde{X}$ is possible for some $\epsilon_1, \epsilon_2 > 0$, whenever arm k is competitive by definition. Moreover, under such a construction one can change reward distribution of $\tilde{Y}_{k^*}(\tilde{X})$ such that reward $\tilde{R}_{k^*}$ has the same distribution as R_{k^*}. This is done by changing the conditional reward distribution, $f_{\tilde{Y}_{k^*}|X}(r) = \frac{f_{Y_{k^*}|X}(r) f_X(x)}{f_{\tilde{X}}(x)}$.

Due to this, if an arm is competitive, there exists a new bandit instance with latent randomness $\tilde{X}$ and conditional rewards $\tilde{Y}_{k^*}|X$ and $\tilde{Y}_k|X$ such that $f_{R_{k^*}} = f_{\tilde{R}_k^*}$ and $\mathbb{E}\left[\tilde{R}_k\right] > \mu_{k^*}$, with f_{R_k} denoting the probability distribution function of the reward from arm k and $\tilde{R}_k$ representing the reward from arm k in the new bandit instance.

Therefore, if these are the only two arms in our problem, then from Lemma 11,

$$\lim_{T\to\infty} \inf \frac{\mathbb{E}\left[n_k(T)\right]}{\log T} \geq \frac{1}{D(f_{R_k}(r)||f_{\tilde{R}_k}(r))},$$

where $f_{\tilde{R}_k}(r)$ represents the reward distribution of arm k in the new bandit instance.

Moreover, if we have more $K-1$ sub-optimal arms, instead of just 1, then

$$\lim_{T\to\infty} \inf \frac{\mathbb{E}\left[\sum_{\ell\neq k^*} n_\ell(T)\right]}{\log T} \geq \frac{1}{D(f_{R_k}(r)||f_{\tilde{R}_k}(r))}.$$

Consequently, since $\mathbb{E}\left[Reg(T)\right] = \sum_{ell=1}^{K} \Delta_\ell \mathbb{E}\left[n_\ell(T)\right]$, we have

$$\lim_{T\to\infty} \inf \frac{\mathbb{E}\left[Reg(T)\right]}{\log(T)} \geq \max_{k\in\mathcal{C}} \frac{\Delta_k}{D(f_{R_k}||f_{\tilde{R}_k})}. \tag{3.109}$$

A stronger lower bound valid for the general case A stronger lower bound for the general case can be shown by using the result in Proposition 1 of [67]. If $\mathcal{P}$ denotes the set of all possible joint probability distribution under which all pseudo-reward constraints are satisfied and P denotes the underlying unknown joint probability distribution which has k^* as the optimal arm. Then, the expected cumulative regret for any algorithm that achieves a sub-polynomial regret is lower bounded as

$$\lim \inf_{T\to\infty} \frac{Reg(T)}{\log T} \geq L(P),$$

where $L(P)$ is the solution of the optimization problem:

$$\begin{aligned}
&\min_{\eta(k)\geq 0, k\in\mathcal{K}} \sum_{k\in\mathcal{K}} \eta(k) \left(\max_{\ell\in\mathcal{K}} \mu_\ell - \mu_k\right) \\
&\text{subject to} \sum_{k\in\mathcal{K}} \eta(k) D(P,Q,k) \geq 1, \quad \forall Q \in \mathcal{Q}, \qquad (3.110)\\
\textit{where} \quad &\mathcal{Q} = \{Q \in \mathcal{P} : f_R(R_{k^*}|Q,k^*) = f_R(R_{k^*}|P,k^*) \;\text{ and }\; k^* \neq \arg\max_{k\in\mathcal{K}} \mu_k(Q)\}.
\end{aligned}$$

Here, $D(P,Q,k)$ is the KL-Divergence between reward distributions of arm k under joint probability distributions P and Q, i.e., $f_R(R_k|\theta,k)$ and $f_R(R_k|\lambda,k)$. The term $\mu_k(Q)$ represents the mean reward of arm k under the joint probability distribution Q.

To interpret the lower bound, one can think of $\mathcal{Q}$ as the set of all joint probability distributions, under which the reward distribution of arm k^* remains the same, but some other arm $k' \neq k^*$ is optimal under the joint probability distribution. The optimization problem reflects the amount of samples needed to distinguish these two joint probability distributions. This result is based on the original result of [27], which has been used recently in [14, 67] for studying other bandit problems.

r	$s_{2,1}(r)$
0	$\frac{2}{3}$
1	$\frac{6}{7}$

r	$s_{1,2}(r)$
0	$\frac{3}{4}$
1	$\frac{2}{3}$

(a)	$R_2 = 0$	$R_2 = 1$
$R_1 = 0$	0.1	0.2
$R_1 = 1$	0.3	0.4

(b)	$R_2 = 0$	$R_2 = 1$
$R_1 = 0$	a	b
$R_1 = 1$	c	d

Table 3.5: The top row shows the pseudo-rewards of arms 1 and 2, i.e., upper bounds on the conditional expected rewards (which are known to the player). The bottom row depicts two possible joint probability distribution (unknown to the player). Under distribution (a), Arm 1 is optimal and all pseudo-reward except $s_{2,1}(1)$ are tight.

Lower bound discussion in general framework

Consider the example shown in Table 3.5, for the joint probability distribution (a), Arm 1 is optimal. Moreover, all pseudo-rewards except $s_{2,1}(1)$ are tight, i.e.,$s_{\ell,k}(r) = \mathbb{E}\left[R_\ell | R_k = r\right]$. For the joint probability distribution shown in (a), expected pseudo-reward of Arm 2 is 0.8 and hence it is competitive. Due to this, our C-UCB and C-TS algorithms pull Arm 2 $\mathrm{O}(\log T)$ times.

However, it is not possible to construct an alternate bandit environment with joint probability distribution shown in Table 3.5(b), such that Arm 2 becomes optimal while maintaining the same marginal distribution for Arm 1, and making sure that the pseudo-rewards still remain upper bound on conditional expected rewards. Formally, there does not exist a, b, c, d such that $c + d = 0.7$, $\frac{c}{a+c} < 3/4$, $\frac{b}{a+b} < 2/3$, $\frac{d}{b+d} < 2/3$, $\frac{d}{d+c} < 6/7$ and $a + b + c + d = 1$. This suggests that there should be a way to achieve $\mathrm{O}(1)$ regret in this scenario. We believe this can be done by using all the constraints (imposed by the knowledge of pair-wise pseudo-rewards to shrink the space of possible joint probability distributions) when calculating empirical pseudo-reward. However, this becomes tough to implement as the ratings can have multiple possible values and the number of arms is more than 2. We leave the task of coming up with a practically feasible and easy to implement algorithm that achieves bounded regret whenever possible in a general setup as an interesting open problem.

Chapter 4

Best-Arm Identification in Correlated Bandits

In the last chapter, we proposed the novel correlated bandit framework and studied the task of cumulative reward maximization. In this chapter, we focus on an alternative objective, where the goal is to identify the best-arm in as few samples as possible. This problem is termed as the best-arm identification and it often requires different design of algorithms (and subsequently different analysis techniques) relative to the task of cumulative reward maximization. In this chapter, we present a way to design efficient algorithms for the task of best-arm identification in the correlated multi-armed bandit framework. We begin our discussion by first contrasting the task of best-arm identification with respect to cumulative reward maximization.

4.1 Introduction

Best-arm Identification in Bandits with Independent Arms. Instead of maximizing the cumulative reward, an alternative objective in the Multi-Armed Bandit setting is to identify the *best arm* (i.e., the arm with the largest mean reward) from as few samples as possible. While reward maximization has been studied extensively, the best-arm identification problem is seldom explored in settings outside of the *classical MAB framework*, i.e., the setting where rewards corresponding to different arms are independent of each other. The best-arm identification problem can be formulated in two different ways, namely fixed confidence [31] and fixed budget [38]. In the fixed confidence setting, the player is provided with a *confidence parameter* δ and their goal is to achieve the fastest (i.e., with the least number of samples) possible identification of the best arm with a probability of at least $1 - \delta$. In the fixed budget setting, the number of samples that the player can receive is fixed, and the goal is to identify the best arm with the highest possible confidence. In this chapter, we focus on the fixed confidence setting.

The best arm identification problem has been explored in the classical MAB framework [40, 68, 35, 69, 70, 71, 72] and three distinct approaches have shown promise, namely, the racing/successive elimination,

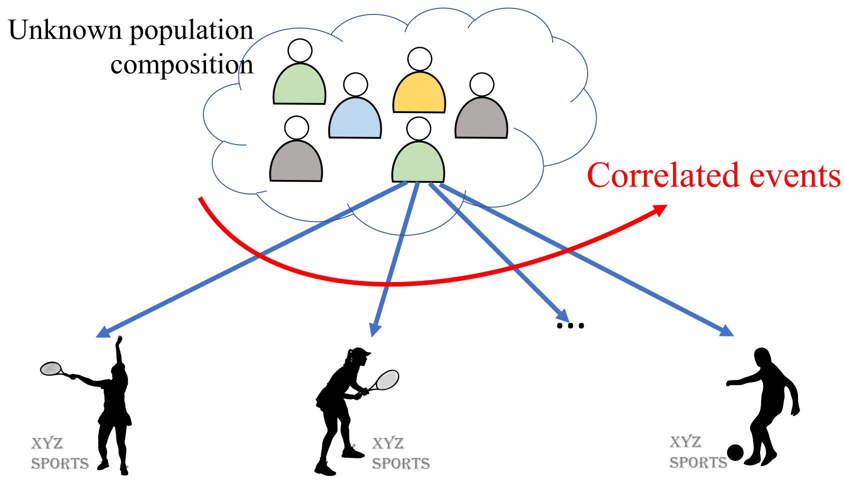


Figure 4.1: The ratings of a user corresponding to different versions of the same ad are likely to be correlated. For example, if a person likes first version, there is a good chance that they will also like the 2nd one as it also related to tennis. However, the population composition is unknown, i.e., the fraction of people liking the first/second or the last version is unknown.

law of iterated logarithm upper confidence bound (lil'UCB) and lower and upper confidence bound (LUCB) based approaches. These algorithms maintain upper and lower confidence bound indices for each arm and usually stop once the lower confidence index of one arm becomes larger than upper confidence bound of all other arms (discussed in more detail in Section 4.3). These three approaches differ in their approach of sampling arms. The successive elimination approach samples arms in a round robin manner, lil'UCB samples the arm with the largest upper confidence bound index at round t and LUCB samples two distinct arms at each round, first it samples the arm with the largest empirical mean and then amongst the rest it samples an arm with the largest upper confidence bound index.

These best-arm identification algorithms have found their use in a wide variety of application settings, such as clinical trials [4] , ad-selection campaigns [10] , crowd-sourced ranking [35] and hyperparameter optimization [34] by treating different different drugs/treatments, advertisements, items to be ranked and hyperparameters as the arms in the multi-armed bandit problem.

Best-arm Identification when Rewards are Correlated across arms. The aforementioned best-arm identification algorithms all operate under the assumption that the rewards from different arms are independent of each other; e.g., at a given round t, the reward obtained from arm k does not provide any information about the reward that one might have received if they sampled another arm ℓ. However, this may not be the case in many applications of MABs. For instance, the response of a user for different advertisements in an ad-campaign is likely to be correlated as the ad designs may be related or starkly different with each other (see Figure 4.1). One way to learn these correlations would be to pull multiple arms at each round t. Since this is not allowed in the standard MAB setup, we assume that *partial* information about such correlations

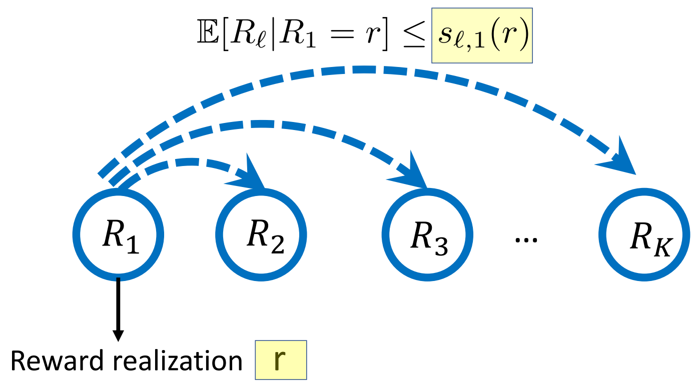


Figure 4.2: Upon observing a reward r from an arm k, pseudo-rewards $s_{\ell,k}(r)$, give us an upper bound on the conditional expectation of the reward from arm ℓ given that we observed reward r from arm k. These pseudo-rewards models the correlation in rewards corresponding to different arms.

is available a priori. In practice, the presence of such correlations may be known beforehand either through domain expertise or through controlled studies where each user is presented with multiple arms. For example, before starting ad campaign, partial information may be known about the *expected* reward we would receive from a user by showing that ad version ℓ, given their response to version k. A similar argument can be made in the application domain of clinical trials, namely in identifying the best drug for an unknown disease. There, the effect of different drugs on an individual may be correlated if the drugs share similar or contrasting components among them. In this context, the correlations would be expected to be known by the domain expertise of the physicians involved. The current best-arm identification algorithms cannot leverage these correlations to reduce the number of samples required in identifying the best arm. This work aims to fill this gap in the literature through the use correlated MAB model proposed in the previous chapter.

Proposed C-LUCB Algorithm and its Sample Complexity. After establishing a correlated bandit model in Chapter 3, we then focus on designing best-arm identification algorithms, that are able to make use of this correlation information to identify the best-arm in fewer samples than the classical best-arm identification algorithms. In particular, we propose an approach that makes use of the pseudo-reward information and extends the LUCB approach to the correlated bandit setting. Our sample complexity analysis shows that the proposed C-LUCB approach is able to explore certain arms without explicitly sampling them. Due to this, we see that these arms, termed as non-competitive contribute only an $\mathrm{O}(1)$ term in the sample complexity as to the typical $\mathrm{O}\left(\log \frac{1}{\delta}\right)$ contribution by each arm. As a result of this, we are able to provide better sample complexity results than LUCB in the correlated bandit setting. In particular,

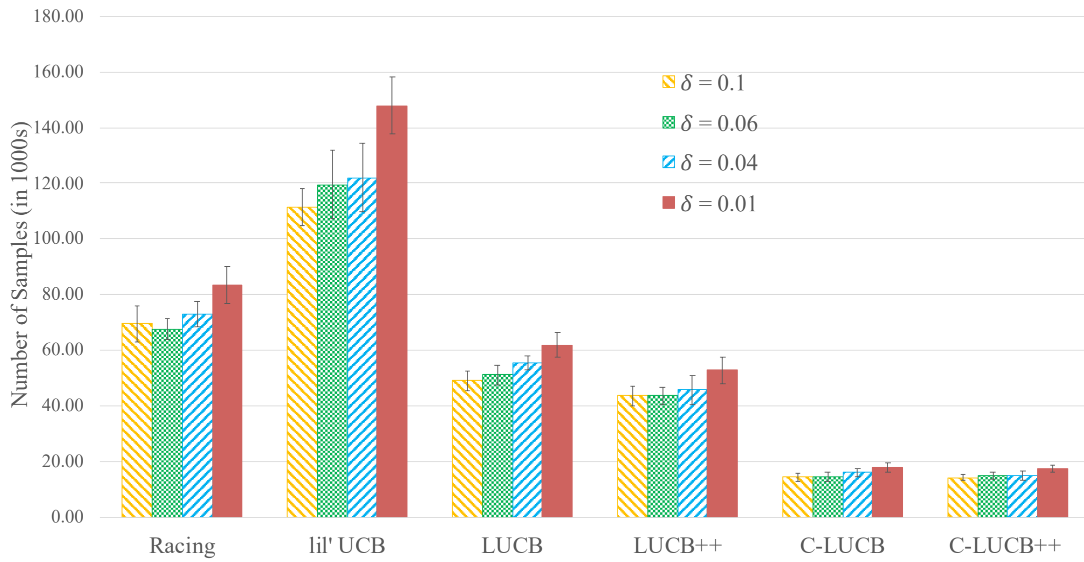


Figure 4.3: This plot illustrates the number of samples required by different algorithms to identify the best movie genre out of the 18 possible movie genres in the Movielens dataset with confidence $1-\delta$. As δ decreases, the algorithms need more samples to identify the best arm. As our proposed C-LUCB and C-LUCB++ algorithms utilize correlation information, they identify the best arm in fewer samples relative to Racing, lil'UCB, LUCB and LUCB++.

the LUCB algorithm stops with probability $1-\delta$ after obtaining at most $\sum_{k \in \mathcal{K}} \frac{2\zeta}{\Delta_k^2}\left(\log\left(\frac{K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$ samples, where $\Delta_k = \mu_{k^*} - \mu_k$, i.e., the difference in mean reward of optimal arm k^* and mean reward of arm k and $\Delta_{k^*} = \min_{k \neq k^*} \Delta_k$, i.e., the gap between best and second best arm and $\zeta > 0$ is a constant. The C-LUCB stops after at most $\sum_{k \in \mathcal{C}} \frac{2\zeta}{\Delta_k^2}\left(\log\left(\frac{2K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right) + O(1)$ samples with probability $1-\delta$. Here, $\mathcal{C} \subseteq \mathcal{K}$ with $2 \leq |\mathcal{C}| \leq K$ depending on the problem instance. As the size of the set $\mathcal{C}$ can be smaller than $\mathcal{K}$, we improve upon the sample complexity results of standard approaches of best-arm identification. This theoretical advantage gets reflected in our experiments on two real-world recommendation datasets, namely, Movielens and Goodreads. For instance, Figure 4.3 illustrates the performance of our proposed algorithms in a correlated bandit framework, where the goal is to identify the best movie genre from the set of 18 movie genres in the Movielens dataset. As our proposed approach utilizes the correlations in the problem, they draw fewer samples than the Racing, lil'UCB and the LUCB based approaches.

Organization of the rest of the chapter. In Section 4.2 of this chapter, we review the correlated multi-armed bandit framework, where correlation between arms is captured in the form of pseudo-rewards. In Section 4.3, we review state-of-the-art best-arm identification algorithms such as successive elimination (or racing), lil'UCB, and LUCB designed for the classical (independent arm) framework. We also discuss how our work contrasts with other best-arm identification works studied in structured and spectral bandit

r	$s_{2,1}(r)$
0	0.7
1	0.4

r	$s_{1,2}(r)$
0	0.8
1	0.5

	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.4
$R_2 = 1$	0.2	0.2

(a)

	$R_1 = 0$	$R_1 = 1$
$R_2 = 0$	0.2	0.3
$R_2 = 1$	0.4	0.1

(b)

Table 4.1: The top row shows the pseudo-rewards of arms 1 and 2, i.e., upper bounds on the conditional expected rewards (which are known to the player). The bottom row depicts two possible joint probability distribution (unknown to the player). Under distribution (a), Arm 1 is optimal whereas Arm 2 is optimal under distribution (b).

frameworks. In Section 4.4 we propose the C-LUCB algorithm, and compare it with state-of-the-art approaches. We discuss several variants of C-LUCB in Section 4.6. In Section 4.5 we analyze the sample complexity analysis of C-LUCB and discuss its proof technique and implications. This analysis reveals that utilizing correlations can lead to significant reduction in the number of samples required to identify the best-arm. Finally, in Section 4.7 we demonstrate the practical applicability our proposed model and algorithm via extensive experiments on real-world recommendation datasets.

4.2 The Correlated Multi-Armed Bandit Model

4.2.1 Problem formulation

Consider a Multi-Armed Bandit setting with K arms $\{1, 2, \dots K\}$. At each round t, we sample an arm $k_t \in \mathcal{K}$ and receive a random reward $R_{k_t} \in [0, b]$. Among the set of K arms, we denote the arm with the largest mean reward as the *best-arm* k^*, i.e., $k^* = \arg\max_{k \in \mathcal{K}} \mu_k$. In the *fixed-confidence* setting [31], the objective is to identify the best-arm in as few samples as possible. In particular, given $\delta > 0$, the goal is to devise a sampling strategy that stops at some round T (a random variable) and declares an arm k^{out} as the optimal arm, where,

$$\Pr(k^{\text{out}} = k^*) \geq 1 - \delta.$$

Put differently, we aim to find the best arm with probability at least $1 - \delta$ while minimizing the total *number of samples* drawn from the arms. We note that the number of samples can be different from the number of rounds T as some algorithms (e.g., LUCB, Racing) sample multiple arms in one round. Using the total number of samples drawn until round T allows us to compare them fairly against algorithms that draw only one sample at each round t (e.g., lil'UCB).

The classical multi-armed bandit setting implicitly assumes that the rewards $R_1, R_2, \dots, R_K$ are independent. That is, $\Pr(R_\ell = r_\ell | R_k = r) = \Pr(R_\ell = r_\ell) \quad \forall r_\ell, r$ and $\forall \ell, k$, which implies that, $\mathbb{E}[R_\ell | R_k = r] =$

$\mathbb{E}[R_\ell] \quad \forall r, \ell, k$. Motivated by the fact that rewards of a user corresponding to different arms might be correlated, we consider a setup where $f_{R_\ell|R_k}(r_\ell|r_k) \neq f_{R_\ell}(r_\ell)$, with $f_{R_\ell}(r_\ell)$ denoting the probability distribution function of the reward from arm ℓ. Consequently, due to such correlations, we have $\mathbb{E}[R_\ell|R_k] \neq \mathbb{E}[R_\ell]$.

In our problem setting, we consider that the player has partial knowledge about the joint distribution of correlated arms in the form of *pseudo-rewards*, as defined below:

Definition 11 (Pseudo-Reward)**.** *Suppose we sample arm k and observe reward r. Then the pseudo-reward of arm ℓ with respect to arm k, denoted by $s_{\ell,k}(r)$, is an upper bound on the conditional expected reward of arm ℓ, i.e.,*

$$\mathbb{E}[R_\ell|R_k = r] \leq s_{\ell,k}(r). \tag{4.1}$$

For convenience, we set $s_{\ell,\ell}(r) = r$.

Remark 10. *Note that the pseudo-rewards are upper bounds on the expected conditional reward and not hard bounds on the conditional reward itself. This makes our problem setup practical as upper bounds on expected conditional reward are easier to obtain, as illustrated below.*

The pseudo-reward information consists of a set of $K \times K$ functions $s_{\ell,k}(r)$ over $[0, b]$. This information can be obtained in practice through either domain and expert knowledge or from controlled surveys. For instance, in the context of medical testing, where the goal is to identify the best drug to treat an ailment from among a set of K possible options, the effectiveness of two drugs is correlated when the drugs share some common ingredients. Through domain knowledge of doctors, it is possible to answer questions such as "what are the chances that drug B would be effective given drug A was not effective?", through which we can infer the pseudo-rewards.

Remark 11 (Reduction to Classical Multi-Armed Bandits)**.** *When all pseudo-reward entries are unknown, then all pseudo-reward entries can be filled with maximum possible reward for each arm, that is, $s_{\ell,k}(r) = b \; \forall r, \ell, k$. In that case, the problem framework studied in this chapter reduces to the setting of the classical Multi-Armed Bandit problem.*

While the pseudo-rewards are known in our setup, the underlying joint probability distribution of rewards is unknown. For instance, Table 4.1(a) and Table 4.1(b) show two joint probability distributions of the rewards that are both possible given the pseudo-rewards at the top of Table 4.1. If the joint distribution is as given in Table 4.1(a), then Arm 1 is optimal, while Arm 2 is optimal if the joint distribution is as given in Table 4.1(b).

4.2.2 Application for best-arm identification in correlated multi-armed bandits

Consider a scenario where a company needs to run a display advertising campaign in a community for one of their products, and their design team has proposed several different designs. The traction (i.e., the number of clicks, time spent on the ad) that the company generates is likely to be dependent on the design that is used for publicity. In order to find the best design, the company can run a best-arm identification algorithm by viewing the problem as a multi-armed bandit problem. Here, at each round t, a new user of that community enters the system and they show one of the K designs (i.e., arms) to this user. The reward is received through the response of the user to the ad. A straightforward solution would be to treat this problem as a classical multi-armed bandit problem and use a well known best-arm identification algorithm such as lil'UCB, LUCB or successive elimination to identify the best design for the community. But, in practice, the rewards corresponding to different designs are likely to be correlated to one another. Consider the example shown in Figure 4.1, over there if a user reacts positively to the first design, the user is also likely to react positively to the second ad as both ads are related to tennis. Such correlations, when accounted for in the form of pseudo-rewards, can help us identify the best-arm in much fewer samples relative to algorithms such as lil'UCB, LUCB and Successive elimination that do not account for correlations in choices.

These correlations could be known from a controlled survey or a previous advertisement campaign performed in a different demographic. For instance, from these surveys one can interpret information such as "users who like ad 1 representing tennis tend to like ad 2 that also represents tennis but not ad K which represents soccer". If a company wants to identify the best ad in a new demographic, it can use this learned correlation information to identify the best-ad in a quick manner. Note that the population composition in the two demographics may be very different, i.e., the fraction of users liking tennis may be very different, but it is likely that the correlation in choices remain consistent across the two demographics. One can also consider the example of identifying best policy to publicize for a political campaign, where users preferences towards different policies (i.e., climate change, gun control, abortion laws) are often correlated in all demographics, but the marginal distribution of people advocating for a single policy is very different in different communities. In such scenarios, transferring correlation information from one demographic to another by modeling them through pseudo-reward in our correlated bandit framework can help reduce the number of samples needed to identify the best-arm.

These pseudo-rewards can also be known from domain knowledge. Consider the problem of identifying the best drug for the treatment of an unknown disease. The effectiveness of different drugs is likely to be correlated as they often contain similar components. In such a situation, the domain expertise of doctors can tell us "what are the chances that drug y will be effective given drug x was effective?". One can use a conservative upper bound on the answer to this question to model pseudo-rewards. Alternatively, such correlation information could also be obtained on how different people react to different drugs in a community. As the effectiveness of drugs depends on underlying medical conditions of the patients, their response would be correlated. This correlation knowledge can then be transferred to identify the best treatment in a different community, where the distribution of underlying medical conditions may be very different.

4.3 Related Prior Work

The design of best-arm identification algorithms in the fixed-confidence setting have three key design components: i) their sampling strategy, i.e., which arm to pick at round t; ii) their elimination criteria, i.e., when to declare an arm as sub-optimal and remove it from the rest of the sampling procedure; and iii) their stopping criteria, i.e., when to stop the algorithm and declare an arm as the best arm.

In order to accomplish the task of best-arm identification, algorithms use the empirical mean $\hat{\mu}_k(t)$ for arm k at round t. In addition to this, upper confidence bound and lower confidence bound on the mean of arm k are maintained based on the number of samples of arm k, $n_k(t)$, and the input confidence parameter δ. In particular, the upper confidence index $U_k(n_k, \delta) = \hat{\mu}_k(t) + B(n_k, \delta)$ and lower confidence index $L_k(n_k, \delta) = \hat{\mu}_k(t) - B(n_k, \delta)$ are maintained for each arm $k \in \mathcal{K}$. Here $B(n_k, \delta) \propto \sqrt{\frac{\log\left(\frac{\log(n_k)}{\delta}\right)}{n_k}}$ is an *anytime* confidence bound [40, 73] constructed such that

$$\Pr\left(\exists\, n_k \geq 1 : \quad \mu_k \notin [L_k(n_k, \delta),\ U_k(n_k, \delta)]\right) \leq \delta. \tag{4.2}$$

Note that the anytime confidence interval bound the probability of the mean lying outside the confidence interval uniformly for all $n_k \geq 1$, i.e., the probability that the mean lies outside the confidence interval $[L_k(n_k, \delta),\ U_k(n_k, \delta)]$ at *any* round t is upper bounded by δ. In contrast to the Hoeffding bound, which are only valid for a fixed and deterministic n_k, the anytime confidence bound holds true uniformly for all $t \geq 1$ and for random n_k as well. We refer the reader to [73] for a detailed discussion and developments in anytime confidence bounds $B(n_k, \delta)$.

4.3.1 Existing Best-Arm identification strategies

There are three well-known approaches to the best-arm identification problem: i) Successive Elimination (also called racing) [71, 74, 72]; ii) lil'UCB (Law of Iterated Logarithms Upper Confidence Bound) [40]; and iii) LUCB [70, 68] (Lower and Upper Confidence Bound). Below, we briefly introduce these algorithms, and

Algorithm	**Sampling Strategy**	**Eliminate Arm k if**	**Stopping Criteria**
Racing	Round Robin in $\mathcal{A}_t$	$U_k\left(\frac{\delta}{K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$
lil'UCB	Sample k_t, $k_t = \arg\max_k U_k(\delta)$	N/A	$n_{k_t} \geq \alpha \sum_{k \neq k_t} n_k$
LUCB	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} \hat{\mu}_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} U_k\left(\frac{\delta}{K}\right)$	$U_k\left(\frac{\delta}{K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{K}\right)$	$\lvert\mathcal{A}_t\rvert = 1^*$ or $L_{m_1}\left(\frac{\delta}{K}\right) > U_{m_2}\left(\frac{\delta}{K}\right)$
LUCB++	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{K}} \hat{\mu}_k(t)$ $m_2 = \arg\max_{k \in \mathcal{K} \setminus \{m_1\}} U_k\left(\frac{\delta}{2}\right)$	N/A	$L_{m_1}\left(\frac{\delta}{2K}\right) > U_{m_2}\left(\frac{\delta}{2}\right)$
C-LUCB (ours)	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} I_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min\left(\tilde{U}_{k,k}\left(\frac{\delta}{2K}\right), I_k(t)\right)$	$\tilde{U}_k\left(\frac{\delta}{2K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{2K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$
C-LUCB++ (ours)	Sample m_1, m_2, $m_1 = \arg\max_{k \in \mathcal{A}_t} I_k(t)$, $m_2 = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min\left(\tilde{U}_{k,k}\left(\frac{\delta}{2}\right), I_k(t)\right)$	$\tilde{U}_k\left(\frac{\delta}{3K}\right) < \max_{\ell \in \mathcal{A}_t} L_\ell\left(\frac{\delta}{3K}\right)$	$\lvert\mathcal{A}_t\rvert = 1$ or $L_{m_1}\left(\frac{\delta}{4K}\right) > \tilde{U}_{m_2,m_2}\left(\frac{\delta}{4}\right)$

Table 4.2: All best-arm identification algorithms have three key components, i) Sampling strategy at each round t, ii) elimination criteria for an arm and iii) the stopping criteria of the algorithm. We compare these for Racing, lil'UCB, LUCB and LUCB++ algorithms and see the differences in their operation. The indices used for our proposed C-LUCB and C-LUCB++ are defined in (4.6) and (4.8).

present a summary of their arm sampling strategies and elimination and stopping criteria in Table 4.2 [1]. For more details, we refer the reader to [31] that provides a comprehensive survey of best-arm identification in the fixed confidence setting.

Successive Elimination or Racing: The successive elimination (also called racing) strategy maintains a set of active arms $\mathcal{A}_t$ at each round. It samples arms in a round-robin fashion from the set of active arms and at the end of each round, it eliminates an arm k from the set of active arms if the lower confidence index of some other arm $\ell \neq k$, $L_\ell\left(n_\ell, \frac{\delta}{K}\right)$, is strictly larger than the upper confidence index of arm k, $U_k\left(n_k, \frac{\delta}{K}\right)$. It continues this until a single arm is left in the set $\mathcal{A}_t$ and returns that arm as the optimal arm. Two other algorithms, Exponential-gap elimination [75] and PRISM [76], build upon successive elimination to provide stronger theoretical guarantees. However, their empirical performance is not promising as noted in [31].

lil'UCB [40]: The lil'UCB algorithm samples the arm with the largest upper confidence index $U_k(n_k, \delta)$ at

[1] The confidence bound $C(n_k(t), \delta)$, and subsequently lower and upper confidence indices $L_k(n_k(t), \delta)$ and $U(n_k(t), \delta)$, depend on the number of rounds t, the number of samples of arm k till round t $n_k(t)$ and the confidence parameter δ. For brevity purposes, at times we represent the confidence bound as $C(n_k, \delta)$ or $C(\delta)$ and the LCB, UCB indices as $L_k(t, \delta), L_k(n_k, \delta)$ or $L_k(\delta)$ and $U_k(t, \delta), U_k(n_k, \delta)$ or $U_k(\delta)$ respectively.

round t and stops when an arm has been sampled more than $\frac{\alpha t}{\alpha+1}$ times till round t. In practice, the value of α is taken to be 9. It then declares the most sampled arm as the best-arm.

LUCB [70, 31]: The LUCB approach samples two arms $m_1(t), m_2(t)$ at each round t. Here, $m_1(t)$ is the arm with the largest empirical reward till round t, and $m_2(t)$ is the arm with the largest UCB index $U_k\left(n_k, \frac{\delta}{K}\right)$ among the rest. The LUCB algorithm stops if the lower confidence bound of the first arm $m_1(t)$ is larger than the upper confidence index of all other arms. [2] Subsequently, another algorithm LUCB++ [69, 35] was designed that operates in a similar manner to LUCB but constructs the upper confidence and lower confidence indices with different confidence parameters for $m_1(t), m_2(t)$. The details of the upper confidence and lower confidence indices for each of these algorithms are presented in Table 4.2. Note that our metric for comparison is the total number of samples collectively drawn from the arms. As LUCB algorithms sample two arms at each round, the total number of samples drawn from the LUCB algorithms is two times the number of rounds t. By comparing the total number of samples and not the number of rounds t, we draw a fair comparison between the performance of LUCB and lil'UCB algorithm.

All the approaches described above work well for the case where rewards are known to be either sub-Gaussian or bounded. Furthermore, if the class of distribution is known (e.g., it is known that rewards are Gaussian with known σ and unknown μ), then there are two more approaches known in the literature, namely Top Two Thompson Sampling (TTTS) [77] and Tracking [43]. In TTTS, the player computes a posterior distribution on the mean reward of each arm and then applies Thompson sampling on the posterior to obtain two samples. It stops when the posterior probability of an arm k being optimal exceeds a certain threshold $\tau_k(n_k, \delta)$. The TTTS algorithm can be computationally intensive as it involves the computation of posterior probability in each round of their algorithm. In [43], authors evaluate a lower bound for the Multi-Armed bandit problem in the form of an optimization problem. They propose a tracking based approach, that solves the optimization problem at each round to obtain an estimated rate at which each arm should be sampled at round t and sample arms in proportion to that rate. More recently, [78] proposed alternative approaches to the track-and-stop algorithm that do not require solving an optimization problem at each round. Instead, they view the optimization problem as an unknown game and have sampling rules based on iterative saddle point strategies. All of the approaches listed above require knowing the *class* of reward distribution. Since we only assume that the rewards are bounded and not the class of distribution, we do not focus on extending TTTS or Tracking based approaches to the correlated bandit setting in this chapter.

[2] Equivalently, one can eliminate an arm k from $\mathcal{A}_t$ at the end of each round if the upper confidence index of arm k is smaller than the lower confidence index of some other arm, and stop the algorithm when the set of active arms $|\mathcal{A}_t| = 1$. This implementation of the LUCB algorithm has the same guarantees as the one proposed in [70, 31] while obtaining similar empirical performance.

Algorithm	Confidence Bound $B(n_k, \delta)$	Type	Samples Drawn
Succ Elimination [72]	$\sqrt{\frac{\log\left(\frac{\pi^2 n_k^2}{3\delta}\right)}{2n_k}}$	Racing	577209.4
lil Succ Elimination [31]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	Racing	120498.5
KL-Racing [68]	$d(B) = 2\log\left(\frac{11.1t^{1.1}}{\delta}\right)$*	Racing	147780.4
Racing with [73]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	Racing	82504.7
LUCB with [68]	$\sqrt{\frac{\log\left(\frac{405t^{1.1}}{\delta}\log\left(\frac{405t^{1.1}}{\delta}\right)\right)}{2n_k}}$	LUCB	219510.2
lil LUCB [76]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	LUCB	90523.0
KL-LUCB [68]	$d(B) = 2\log\left(\frac{405.5t^{1.1}}{\delta}\right) + \log\log\left(\frac{405.5t^{1.1}}{\delta}\right)$	LUCB	81154.4
LUCB with [73]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	LUCB	62533.2
lil'UCB [40]	$0.85\sqrt{\frac{\log(\log(0.2585n_k))+0.96\log(67.59/\delta)}{n_k}}$	lil'UCB	140987.0
lil-KL-LUCB [35]	$d(B) = 1.86\log\left(\kappa\log_2\left(\frac{2n_k}{\delta}\right)\right)$	LUCB++	92000.0
LUCB++ with [73]	$0.85\sqrt{\frac{\log(\log(0.5n_k))+0.72\log(5.2/\delta)}{n_k}}$	LUCB++	55138.8

Table 4.3: Description of the well-known best-arm identification algorithms and the confidence bound $B(n_k, \delta)$ that they use for [0,1] bounded rewards. All the three types of algorithms have evolved with time due to the development of tighter $1-\delta$ anytime confidence intervals $B(n_k, \delta)$. We see that the algorithms perform best with the confidence bound suggested in [73], and hence we use that for all our implementations of Racing, LUCB, LUCB++ and our proposed algorithm in the rest of the chapter. The reported sample complexity is for the task of identifying best movie genre from the set of 18 movie genres in the Movielens dataset. Experimental setup is described in detail in Section 4.7.

4.3.2 Developments in Confidence sequence $B(n_k, \delta)$

It is important to note that the performance of the algorithms described above depends critically on the tightness of the confidence bound $B(n_k, \delta)$. For instance, initially the LUCB algorithm was proposed with the confidence interval $B(n_k, \delta) = \sqrt{\frac{\log\left(\frac{405n_k^{1.1}}{\delta}\log\left(\frac{405n_k^{1.1}}{\delta}\right)\right)}{2n_k}}$ (See [70]) for $[0, 1]$ bounded random variables. Subsequently tighter bounds as in [31], [68] were developed, which led to performance improvements in the LUCB algorithm. See Table 4.3 for a comparison different confidence bound developed over time and how they affect the empirical performance of the best-arm identification algorithms[3]. For a more detailed comparison of different confidence bounds $B_k(n_k, \delta)$, we refer the reader to Table 2 of [73]. To the best of our knowledge, the tightest $1-\delta$ anytime confidence interval for bounded and sub-Gaussian random variables is proposed in [73], which constructs

$$B(n_k, \delta) = 0.85\sqrt{\frac{\log(\log(0.5n_k)) + 0.72\log(5.2/\delta)}{n_k}}. \tag{4.3}$$

[3]The bound proposed in [68, 35] are KL based bounds that evaluate the indices $U_k(n_k, \delta), L_k(n_k, \delta)$ as $\inf\{j > \hat{\mu}_k : n_k(t)d_{kl}(\hat{\mu}_k, j) < d(B)\}$ and $\sup\{j < \hat{\mu}_k : n_k(t)d_{kl}(\hat{\mu}_k, j) < d(B)\}$. The distance $d_{kl}(x, y)$ is evaluated as $x\log(x/y) + (1-x)\log((1-x)/(1-y))$

Due to this observation, which is also supported by empirical evidence in Table 4.3, we use the bound suggested by [73] in all implementations of Successive Elimination, LUCB and our proposed algorithm. However, our algorithm and analysis extend to arbitrary $1-\delta$ anytime confidence interval $B(n_k, \delta)$.

We would also like to highlight the fact that lil'UCB is known to have the best known theoretical sample complexity (in terms of its dependency on the number of arms K). The LUCB algorithm stops with probability $1-\delta$ after obtaining at most $\sum_{k\in\mathcal{K}} \frac{2\zeta}{\Delta_k^2}\left(\log\left(\frac{K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$ samples, where $\Delta_k = \mu_{k^*} - \mu_k$, the difference in mean reward of optimal arm k^* and mean reward of arm k. And $\Delta_{k^*} = \min_{k\neq k^*}\Delta_k$, the gap between best and second best arm. It is known that lil'UCB algorithm has a sample complexity $O\left(\sum_{k\in\mathcal{K}} \frac{1}{\Delta_k^2}\log\left(\frac{\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$ i.e., it avoids the $\log(K)$ term in the numerator, and hence has the best known theoretical sample complexity. However, it has been observed (both in [31] and our experiments) that its empirical performance is inferior to that of the LUCB algorithm. Due to this reason, we focus on proposing an algorithm C-LUCB that extends the LUCB approach to the correlated bandit setting. We have included the performance of lil'UCB in all our experiments.

4.3.3 Algorithms outside the classical setting

Unlike the regret-minimization problem, the best-arm identification problem is relatively unexplored outside of the classical multi-armed bandit setting. A rare exception is the *structured* bandit setting, where mean rewards corresponding to different arms are related to one another through a hidden parameter θ. The underlying value of θ is fixed and unknown, but the mean reward mappings $\theta \to \mu_k(\theta)$ are known. The *linear* bandit setting is a special case of structured bandits, where mean reward mappings are of the form $x_k^\intercal\theta$ with x_k known to the player. The best-arm identification problem has been studied in [46, 48] for *linear* bandits and in [47] for the general structured bandit setting. Other special cases of structured bandits include global bandits [17], regional bandits [16] and the generalized linear bandits [22]; to the best of our knowledge the best arm identification problem has not been addressed in these special cases. Note that in the full generality, the structured bandit framework is simply a bandit problem with constraints on the joint probability distribution [67], but that setting has only been studied for the objective of regret minimization and not best-arm identification. To the best of our knowledge, the structured bandits work studying best-arm identification [46, 48, 47] assume the presence of a hidden parameter θ through which mean rewards of different arms are related to one another. Our correlated bandit framework focuses on structured bandit settings by modeling the correlations explicitly through the knowledge of pseudo-rewards.

Recently, best-arm identification was studied under the spectral bandit framework [79], which assumes that the arms are the nodes of known a weighted graph, with $w_{a,b}$ denoting the weight between arms a and arms b. The spectral bandit framework poses a restriction on the relationship between mean rewards of individual arms by assuming that $\sum_{a,b\in\mathcal{K}} w_{a,b}\frac{(\mu_a-\mu_b)^2}{2} \leq R$, where R is known to the player.

The correlated bandit model considered in this chapter is fundamentally different from the structured bandit framework as detailed below.

1. The model studied here explicitly models the correlations in the rewards of different arms *at any given round* t. In structured bandits, the mean rewards are related to each other, but the reward realizations at a given round are not necessarily correlated. Similar to structured bandits, the work on spectral bandits [79] considers a setup with constrains between mean rewards of different arms, but does not capture the correlations explicitly in their framework.

2. It is also possible to use the structured bandit framework for the objective of identify best global recommendation in an ad-campaign. However, there are two major challenges i) In deciding upon the hidden parameter θ that we need to use, through which the mean rewards are related to one another. ii) Secondly, in the structured bandits framework, the reward mappings from θ to $\mu_k(\theta)$ need to be *exact*. If they happen to be incorrect, then the algorithms for structured bandit cannot be used as they rely on the correctness of $\mu_k(\theta)$ to construct confidence intervals on the unknown parameter θ. In contrast, the model studied here only relies on the pseudo-rewards being upper bounds on the conditional expectations $\mathbb{E}\left[R_\ell | R_k = r\right]$. Our proposed algorithm works even when these bounds are not tight. The lack of hidden parameter θ and pseudo-rewards being upper bounds on conditional expectations make the model studied in this chapter more suitable for practical scenarios where the goal is to identify the best global recommendation.

4.4 Proposed Correlated-LUCB Best-arm Identification Algorithm

In the correlated MAB framework, the rewards observed from one arm can help estimate the rewards from other arms. Our key idea is to use this information to reduce the number of samples taken before stopping. We do so by maintaining the *empirical pseudo-rewards* of all pairs of distinct arms at each round t.

4.4.1 Empirical Pseudo-Rewards and New UCB indices

In our correlated MAB framework, pseudo-reward of arm ℓ with respect to arm k provides us an estimate on the reward of arm ℓ through the reward sample obtained from arm k. We now define the notion of empirical pseudo-reward which can be used to obtain an *optimistic estimate* of μ_ℓ through just reward samples of arm k.

Definition 12 (Empirical and Expected Pseudo-Reward)**.** *After t rounds, arm k is sampled $n_k(t)$ times. Using these $n_k(t)$ reward realizations, we can construct the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ for each arm ℓ with respect to arm k as follows.*

$$\hat{\phi}_{\ell,k}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{k_\tau = k}\, s_{\ell,k}(r_{k_\tau})}{n_k(t)}, \qquad \ell \in \{1,\ldots,K\} \setminus \{k\}. \tag{4.4}$$

The expected pseudo-reward of arm ℓ with respect to arm k is defined as

$$\phi_{\ell,k} \triangleq \mathbb{E}\left[s_{\ell,k}(R_k)\right]. \tag{4.5}$$

For convenience, we set $\hat{\phi}_{k,k}(t) = \hat{\mu}_k(t)$ and $\phi_{k,k} = \mu_k$. Note that the empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ is defined with respect to arm k and it is only a function of the rewards observed by sampling arm k.

Observe that $\mathbb{E}\left[s_{\ell,k}(R_k)\right] \geq \mathbb{E}\left[\mathbb{E}\left[R_\ell | R_k = r\right]\right] = \mu_\ell$. Due to this, empirical pseudo-reward $\hat{\phi}_{\ell,k}(t)$ can serve as an estimated upper bound on μ_ℓ. Using the definitions of empirical pseudo-reward, we now define auxiliary UCB indices, namely crossUCB and pseudoUCB indices, which are used in the selection and elimination strategy of the C-LUCB algorithm.

Definition 13 (CrossUCB Index $\tilde{U}_{\ell,k}(t,\delta)$)**.** *At the end of round t, we have $n_k(t)$ samples of arm k. Using these, we define the CrossUCB Index of arm ℓ with respect to arm k as*

$$\tilde{U}_{\ell,k}(t,\delta) \triangleq \hat{\phi}_{\ell,k}(t) + B(n_k,\delta). \tag{4.6}$$

Furthermore, we define

$$\tilde{U}_\ell(t,\delta) = \min_k \tilde{U}_{\ell,k}(t,\delta),$$

i.e., the tightest of the K upper bounds, $\tilde{U}_{\ell,k}(t,\delta)$, for arm ℓ.

Note that the CrossUCB index for arm ℓ with respect to arm k, $\tilde{U}_{\ell,k}(t,\delta)$ is constructed only through the samples obtained from arm k. Furthermore, we have $\tilde{U}_{k,k}(t,\delta) = \hat{\mu}_k(t) + B(n_k,\delta)$, which coincides with the standard upper confidence index used in the best-arm identification literature. We use the confidence bound suggested by [73] (see Section 4.3) for the construction of $B(n_k,\delta)$ for $[0,b]$ bounded random variables, i.e.,

$$B(n_k,\delta) = \frac{1.7b}{2}\sqrt{\frac{\log\left(\log\left(\frac{b^2 n_k}{2}\right)\right) + 0.72\log(5.2/\delta)}{n_k}}. \tag{4.7}$$

As pseudo-rewards are *upper bounds* on conditional expected reward, they can only be used to construct alternative upper bounds on the mean reward of other arms and not alternative lower bounds. Due to this reason, we keep the definition of lower confidence index $L_k(t,\delta)$ the same as that in the classical multi-armed bandit setting, i.e., $L_k(t,\delta) = \hat{\mu}_k(t) - B(n_k,\delta)$. In addition to the CrossUCB and the LCB index for each arm, we now define the PseudoUCB index of arm ℓ with respect to arm k. The PseudoUCB indices prove useful for the design and analysis of our proposed algorithm.

Definition 14 (PseudoUCB Index $I_{\ell,k}(t)$). *We define the PseudoUCB Index of arm ℓ with respect to arm k as follows.*

$$I_{\ell,k}(t) \triangleq \hat{\phi}_{\ell,k}(t) + b\sqrt{\frac{2\log t}{n_k(t)}} \tag{4.8}$$

Furthermore, we define $I_\ell(t) = \min_k I_{\ell,k}(t)$, the tightest of the K upper bounds for arm ℓ.

Note that the PseudoUCB Index uses a confidence bound, $b\sqrt{\frac{2\log t}{n_k(t)}}$, which is typically used in the UCB1 algorithm ([32]) for the objective of cumulative reward maximization. It has the property that $\Pr(I_\ell(t) < \mu_\ell) \leq Kt^{-3}$ [See lemma 20], i.e., the probability of mean lying outside the pseudoUCB index $I_\ell(t)$ at round t decays exponentially with the number of rounds t. This property allows us to show desirable sample complexity results for our proposed algorithm in Section 4.5. We now present the C-LUCB algorithm, that makes use of the PseudoUCB, CrossUCB and LCB indices in its strategy for sampling arms, eliminating arms and stopping the algorithm.

4.4.2 C-LUCB Algorithm

The C-LUCB algorithm maintains a set of active arms $\mathcal{A}_t$, which is initialized to the set of all arms $\mathcal{K} = \{1,\dots,K\}$. At each round t, it samples arms, eliminates arms and then decides whether to stop as described below.

1. **Sampling Strategy:** At each round t, the C-LUCB algorithm samples two arms $m_1(t)$ and $m_2(t)$, where
$$m_1(t) = \arg\max_{k\in\mathcal{A}_t} I_k(t), \quad m_2(t) = \arg\max_{k\in\mathcal{A}_t\setminus\{m_1(t)\}} \min\left(\tilde{U}_{k,k}\left(t,\frac{\delta}{2K}\right), I_k(t)\right).$$

2. **Elimination Criteria:** The C-LUCB algorithm removes an arm k from the set $\mathcal{A}_t$, if the CrossUCB index of arm k is smaller than the LCB index of some other arm in $\mathcal{A}_t$, i.e., if
$$\tilde{U}_k\left(t,\frac{\delta}{2K}\right) < \max_{\ell\in\mathcal{A}_t} L_\ell\left(t,\frac{\delta}{2K}\right).$$
Here, $\tilde{U}_\ell\left(t,\frac{\delta}{2K}\right) = \min_k \tilde{U}_{\ell,k}\left(t,\frac{\delta}{2K}\right)$.

3. **Stopping Criteria:** If $|\mathcal{A}_t| = 1$, stop the algorithm and declare the arm in $\mathcal{A}_t$ as the optimal arm with $1-\delta$ confidence.

Both LUCB and C-LUCB sample the top two arms at round t in $m_1(t)$ and $m_2(t)$ so as to resolve the ambiguity among them as fast as possible. However, C-LUCB uses the additional pseudo-reward information to modify its choice of $m_1(t)$ and $m_2(t)$. In particular, the use of $I_k(t)$ in definition of $m_2(t)$ avoids the sampling of an arm that appears sub-optimal from samples of other arms. Similarly, using the CrossUCB index $\tilde{U}_k\,(t, \delta/2K)$ instead of $\tilde{U}_{k,k}(t, \delta/2K)$, allows the C-LUCB to eliminate some arms earlier than the LUCB algorithm. A comparison of the operation of C-LUCB with LUCB and Racing based algorithms is presented in Table 4.2. We show that the proposed C-LUCB algorithm is $1-\delta$ correct and analyze its sample complexity in the next section. As the key difference between C-LUCB and LUCB is in its sampling strategy, we explore some other variants of C-LUCB in Section 4.6, where we study the effect of performance on altering the definitions of $m_1(t)$ and $m_2(t)$.

4.5 Sample Complexity Results

In this section, we analyze sample complexity of the proposed C-LUCB algorithm, that is, the number of samples required to identify the best arm with probability $1-\delta$. We show that some arms, referred to as *non-competitive* arms, are explored implicitly through the samples of the optimal arm k^* and contribute only an O(1) term in the sample complexity, while other arms called *competitive* arms have an $\mathrm{O}\left(\log(1/\delta)\right)$ contribution in the sample complexity of the C-LUCB algorithm. The correlation information enables us to identify the non-competitive arms using samples from other arms and eliminate them early. For the sample complexity analysis, we assume that the rewards are bounded between $[0,1] \forall k \in \mathcal{K}$. Note that the algorithms do not require this condition and the analysis can also be generalized to any bounded rewards.

4.5.1 Competitive and Non-competitive arms

We now define the notion of *competitive* and *non-competitive* arms, which are important to interpret our sample complexity results for the C-LUCB algorithm. Let k^* denote the arm with the largest mean and $k^{(2)}$ denote the arm with the second largest mean.

Definition 15 (Non-Competitive and Competitive arms)**.** *An arm ℓ is said to be non-competitive if the expected reward of the second best arm $k^{(2)}$ is strictly larger than the expected pseudo-reward of arm ℓ with respect to the optimal arm k^*, i.e, $\tilde{\Delta}_\ell \triangleq (\mu_{k^{(2)}} - \phi_{\ell,k^*}) > 0$. Similarly, an arm ℓ is said to be competitive if $\tilde{\Delta}_\ell = (\mu_{k^{(2)}} - \phi_{\ell,k^*}) \leq 0$. We refer to $\tilde{\Delta}_\ell$ as the pseudo-gap of arm ℓ in the rest of the chapter. We denote the set of the competitive arms as $\mathcal{C}$ and the total number of competitive arms as C in this chapter.*

The best arm k^* and second best arm $k^{(2)}$ have pseudo-gaps $\tilde{\Delta}_{k^*} = (\mu_{k^{(2)}} - \phi_{k^*,k^*}) < 0$ and $\tilde{\Delta}_{k^{(2)}} =$

$(\mu_{k^{(2)}} - \phi_{k^{(2)},k^*}) \leq 0$ respectively, and hence are counted in the set of competitive arms. As $\phi_{\ell,k^*} \geq \mu_\ell$, the pseudo-gap $\tilde{\Delta}_\ell \leq \Delta_\ell$. Due to this, we have $2 \leq C \leq K$.

The central idea behind our C-LUCB approach is that after sampling the optimal arm k^* sufficiently large number of times, the non-competitive (and thus sub-optimal) arms will not be selected as $m_1(t)$ or $m_2(t)$ by the C-LUCB algorithm, and thus will not be explored explicitly. Furthermore, the non-competitive arms can be eliminated from the information obtained through arm k^*. As a result, the non-competitive arms contribute only an $\mathrm{O}(1)$ term in the sample complexity, i.e., the contribution is independent of the confidence parameter δ. However, the competitive arms cannot be discerned as sub-optimal by just using the rewards observed from the optimal arm, and have to be explored $\mathrm{O}\left(\log\left(\frac{1}{\delta}\right)\right)$ times each. Thus, we are able to reduce a K-armed bandit to a C-armed bandit problem, where C is the number of competitive arms. [4]

4.5.2 Analysis of C-LUCB

We start by first proving the $(1-\delta)$-correctness of C-LUCB algorithm and then analyzing its sample complexity in terms of the number of samples obtained until the stopping criterion is satisfied.

Theorem 8 ($(1-\delta)$ correctness of C-LUCB)**.** *Upon stopping, the C-LUCB algorithm declares arm k^* as the best arm with probability* $1-\delta$.

Proof Sketch. To prove Theorem 8, we define three events $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$ below. Let $\mathcal{E}_1$ be the event that empirical mean of all arm lie within their confidence intervals uniformly for all $t \geq 1$

$$\mathcal{E}_1 = \left\{ \forall t \geq 1, \forall k \in \mathcal{K}, \quad \hat{\mu}_k(t) - B\left(n_k(t), \frac{\delta}{2K}\right) \leq \mu_k \leq \hat{\mu}_k + B\left(n_k(t), \frac{\delta}{2K}\right) \right\} \tag{4.9}$$

Define $\mathcal{E}_2$ to be the event that empirical pseudo-reward of optimal arm with respect to all other arms lie within their CrossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_2 = \left\{ \forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{k^*,\ell} \leq \hat{\phi}_{k^*,\ell}(t) + B\left(n_\ell(t), \frac{\delta}{2K}\right) \right\} \tag{4.10}$$

Similarly define $\mathcal{E}_3$ to be the event that the empirical pseudo-reward of the sub-optimal arms with respect to the optimal arm lies within their CrossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_3 = \left\{ \forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{\ell,k^*} \leq \hat{\phi}_{\ell,k^*}(t) + B\left(n_{k^*}(t), \frac{\delta}{2K}\right) \right\} \tag{4.11}$$

Furthermore, we define $\mathcal{E}$ to be the intersection of the three events, i.e.,

$$\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3. \tag{4.12}$$

[4]Observe that k^* and subsequently C are both unknown to the algorithm. Before the start of the algorithm, it is not known which arm is optimal/competitive/non-competitive.

Due to the nature of anytime confidence intervals (See (4.2)) and union bound over the set of arms, we have $\Pr(\mathcal{E}_1^c) \leq \frac{\delta}{2}$, $\Pr(\mathcal{E}_2^c) \leq \frac{\delta}{4}$ and $\Pr(\mathcal{E}_3^c) \leq \frac{\delta}{4}$ giving us $\Pr(\mathcal{E}^c) \leq \delta$. Furthermore, we show that, when event $\mathcal{E}$ occurs, the C-LUCB algorithm always declares k^* as the best arm. This gives us the desired result in Theorem 8. A detailed proof is given in the Section 4.9.6.

Theorem 9. *Given event $\mathcal{E}$ (defined in (4.12)), the expected number of samples drawn by C-LUCB until stopping, is bounded as*

$$\mathbb{E}\left[N^{\text{C-LUCB}} \mid \mathcal{E}\right] \leq \sum_{k \in \mathcal{C}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{2K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) + \frac{3K + 2Kt_0}{1-\delta} + \frac{2}{1-\delta}\left(\frac{(K+1)^3}{t_0} + \frac{2}{t_0^2}\right), \tag{4.13}$$

where $t_0 = \inf\left\{\tau \geq 2 : \Delta_{k^*} \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \; \forall k \notin \mathcal{C}\right\}$ and ζ is a universal constant that depends on the type of confidence bound used to construct $B(n_k, \delta)$ (Section 3b) – the tighter the bound, the smaller the ζ. The gap Δ_k is defined as $\Delta_k \triangleq \mu_{k^*} - \mu_k \quad for\ k \neq k^*$, i.e., the difference in mean reward of optimal arm k^* and mean reward of arm k and $\Delta_{k^*} \triangleq \min_{k \neq k^*} \Delta_k$, i.e., the gap between best and second best arm.

We present a brief proof outline below, while the detailed proof is available in the Section 4.9.5.

Proof Sketch. In order to bound the total number of samples drawn by C-LUCB, we bound the total number of rounds T taken by C-LUCB before stopping. As C-LUCB algorithm pulls two arms $m_1(t)$ and $m_2(t)$ in each round t, the number of samples $N^{\text{C-LUCB}} = 2T$. We obtain an upper bound on the total number of rounds T, considering the following four counts of the number of rounds and obtain an upper bound for each of them under the event $\mathcal{E}$:

1. $T^{(\mathcal{R})}$: Let $T^{(\mathcal{R})}$ denote the number of rounds in which $I_{k^*}(t) < \mu_{k^*}$, i.e., the count of events in which the pseudoUCB index of arm k^* is smaller than the mean of arm k^* at round t.

2. $T^{(C)}$: Define $T^{(C)}$ to be the number of rounds in which $m_1(t), m_2(t) \in \mathcal{C}$ and event $I_{k^*}(t) < \mu_{k^*}$ does not occur.

3. $T^{(NC)}$: Define $T^{(NC)}$ to be the number of rounds in which $m_1(t) \notin \mathcal{C}, m_2(t) \neq k^*$ or $m_2(t) \notin \mathcal{C}, m_1(t) \neq k^*$.

4. $T^{(*)}$: Define $T^{(*)}$ to be the number of rounds in which $m_1(t) = k^*, m_2(t) \notin \mathcal{C}$ or $m_2(t) = k^*, m_1(t) \notin \mathcal{C}$.

We can now see that $T \leq T^{(\mathcal{R})} + T^{(C)} + T^{(NC)} + T^{(*)}$. We show that

$$\Pr(I_{k^*}(t) < \mu_{k^*} | \mathcal{E}) = \frac{\Pr(I_{k^*} < \mu_{k^*}, \mathcal{E})}{\Pr(\mathcal{E})} \leq \frac{\Pr(I_{k^*} < \mu_{k^*}, \mathcal{E})}{1-\delta} \leq \frac{\Pr(I_{k^*} < \mu_{k^*})}{1-\delta} \leq \frac{Kt^{-3}}{1-\delta},$$

giving us $\mathbb{E}\left[T^{(\mathcal{R})}|\mathcal{E}\right] \leq \frac{1}{1-\delta}\sum_{t=1}^{\infty} Kt^{-3} \leq \frac{3K}{2(1-\delta)}$. Next we show that $\Pr\left(T^{(C)} + T^{(*)} \geq \sum_{k\in\mathcal{C}} \frac{\zeta}{\Delta_k^2}\log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\Big|\mathcal{E}\right) = 0$. Due to this,

$$T^{(C)} + T^{(*)} \leq \sum_{k\in\mathcal{C}} \frac{\zeta}{\Delta_k^2}\log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) \quad \text{w.p. } 1-\delta.$$

We then evaluate an upper bound on $\mathbb{E}\left[T^{(NC)}|\mathcal{E}\right]$ and show that it is upper bounded by a O(1) constant, i.e.,

$$\mathbb{E}\left[T^{(NC)}|\mathcal{E}\right] \leq \frac{Kt_0}{1-\delta} + \frac{1}{1-\delta}\left(\frac{(K+1)^3}{t_0} + \frac{2}{t_0^2}\right).$$

Putting these results together, we obtain the result of Theorem 9.

Furthermore, as $\mathbb{E}\left[T^{(NC)}|\mathcal{E}\right], \mathbb{E}\left[T^{(\mathcal{R})}|\mathcal{E}\right]$ is upper bounded by an O(1) constant as $\delta \to 0$, we have $\sum_{t=1}^{\infty} \Pr(\mathcal{E}_t^{\text{NC}}) < \infty$, where $\mathcal{E}_t^{\text{NC}}$ is the event that $m_1(t) \notin \mathcal{C}, m_2(t) \neq k^*$ or $m_2(t) \notin \mathcal{C}, m_1(t) \neq k^*$. By Borel-Cantelli Lemma 1, this implies that with probability 1, the event $\mathcal{E}_t^{\text{NC}}$ takes place only finitely many time steps t. As a result of this, $\exists d_1 : \Pr(T^{(NC)} > d_1|\mathcal{E}) = 0$ *almost surely*. Similarly $\exists d_2 : \Pr(T^{(\mathcal{R})} > d_2|\mathcal{E}) = 0$ *a.s.* As a consequence of this, we have the following result bounding the total number of samples drawn from the C-LUCB algorithm with probability $1-\delta$.

Corollary 2. *The number of samples obtained by C-LUCB is upper bounded as*

$$N^{\textit{C-LUCB}} \leq \sum_{k\in\mathcal{C}} \frac{2\zeta}{\Delta_k^2}\log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) + d \quad \textit{w.p. } 1-\delta, \tag{4.14}$$

where $d = \max(d_1, d_2)$. Note that the $\mathrm{O}\left(\log\left(\frac{1}{\delta}\right)\right)$ term is only summed for the set of competitive arms $\mathcal{C}$, in contrast to the LUCB algorithm where the sample complexity term involves summation of a $\mathrm{O}\left(\log\left(\frac{1}{\delta}\right)\right)$ for all arms $k \in \mathcal{K}$. In this sense, our proposed algorithm reduces a K-armed bandit problem to a C-armed bandit problem.

The key intuition behind our sample complexity result is that the sampling of $m_1(t) = \arg\max_{k\in\mathcal{A}_t} I_k(t)$ ensures that the optimal arm is sampled at least t/K times till round t with high-probability. This in turn ensures that the non-competitive arms are not selected as $m_1(t)$ or $m_2(t)$, due to which we see that their expected number of samples are bounded above by a O(1) constant.

4.5.3 Comparison with the LUCB algorithm

The LUCB algorithm is known to stop after obtaining at most $\left(\sum_{k \in \mathcal{K}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right)$ samples with probability at least $1-\delta$. More formally,

$$N^{\text{LUCB}} \leq \left(\sum_{k \in \mathcal{K}} \frac{2\zeta}{\Delta_k^2} \log\left(\frac{K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)\right), \quad \text{w.p. } 1-\delta.$$

We compare this result with the one that we prove for C-LUCB algorithm in Theorem 9.

Reduction to a C-Armed Bandit problem: As highlighted earlier, in the C-LUCB approach, the $\mathrm{O}\left(\log\left(\frac{1}{\delta}\right)\right)$ term only comes from the set of competitive arms, as opposed to the LUCB algorithm which has $\mathrm{O}(\left(\log\left(\frac{1}{\delta}\right)\right)$ contribution from all its arms. In this sense, C-LUCB algorithm reduces a K-armed bandit problem to a C-armed bandit problem. Depending on the problem instance, the value of C can vary between 2 and K.

Slightly larger number of samples from competitive arms: We see that the contribution coming from a competitive arm in C-LUCB algorithm is $\frac{2\zeta}{\Delta_k^2} \log\left(\frac{2K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$. This is slightly larger than the contribution coming from a sub-optimal arm in LUCB algorithm, where each arm contributes $\frac{2\zeta}{\Delta_k^2} \log\left(\frac{K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$ in the sample complexity. This is due to the fact that we construct slightly wider confidence intervals, $B\left(n_k, \frac{\delta}{2K}\right)$ instead of $B\left(n_k, \frac{\delta}{K}\right)$, in C-LUCB to take advantage of the correlations present in the problem. We see in Section 4.7 that this small increase in the width of confidence intervals does not have a significant impact on the empirical performance of the algorithm.

Theorem 9's result is in conditional expectation: While the sample complexity result of the LUCB algorithm bounds the total number of samples taken with probability $1-\delta$, our sample complexity result bounds the expected samples taken by C-LUCB algorithm under the event $\mathcal{E}$ (Theorem 9). This arises as the analysis of our algorithm requires a transient component, because it tries to avoid sampling non-competitive arm at each round with high probability. We have a result in Corollary 2 that evaluates an upper bound which holds with probability $1-\delta$, but we are unable to quantify the constant d in Corollary 2 and can only characterize d in expectation as done in Theorem 9. An open problem is to evaluate the expected sample complexity of our C-LUCB algorithm for the cases where the event $\mathcal{E}$ does not occur. While such results are hard to obtain theoretically, in all our experiments we observed that the variance in the number of samples drawn by C-LUCB is not much, and is in fact similar to that of the LUCB algorithm in all the experiments performed. This indicates that even when algorithm stops with an incorrect arm, the number

Algorithm	First arm $m_1(t)$	Second arm $m_2(t)$	Samples drawn
C-LUCB	$\arg\max_{k \in \mathcal{A}_t} I_k(t)$	$\arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min \left(\tilde{U}_{k,k}\left(\frac{\delta}{2K}\right), I_k(t) \right)$	39277.8
maxmin-LUCB	$\arg\max_{k \in \mathcal{A}_t} \min_\ell \hat{\phi}_{k,\ell}(t)$	$\arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \tilde{U}_k\left(\frac{\delta}{2K}\right)$	36314.2
2-LUCB	$\arg\max_{k \in \mathcal{A}_t} I_k(t)$	$\arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \tilde{U}_k\left(\frac{\delta}{2K}\right)$	39385.8

Table 4.4: We study two intuitive variants of C-LUCB which differ in their sampling strategy of $m_1(t)$ and $m_2(t)$. Both of them have same elimination and stopping criteria as the C-LUCB algorithm. We report the number of samples needed to identify the best genre from the set of 18 movie genres in the Movielens dataset. While all of these are smaller than the samples drawn by LUCB (which is 61175.4 in this case), the difference between the variants of C-LUCB is minimal. Experimental details are described in detail in Section 4.7, we set the value of $p = 0.2$ (i.e., the fraction of pseudo-reward entries that are replaced by 5) in this experiment. Such similarity in empirical performance has also been observed in our other experiments and we found no clear winner among the three when compared on their empirical performance.

of samples obtained are similar to the samples obtained under the good event $\mathcal{E}$.

The $\log(K)$ term in numerator: Just like the sample complexity result of the LUCB algorithm [31], our sample complexity result also has a $\log(K)$ in its sample complexity result. This is avoidable in the classical MAB framework if one uses the lil'UCB algorithm, which is known to have the optimal theoretical sample complexity in the classical bandit setting as it avoids the $\log(K)$ term in its sample complexity expression. However the use of lil'UCB algorithm leads to worse empirical performance as seen in our experiments and prior work [31]. Due to this reason, we focus only on the extension of LUCB to the correlated bandit setting. The LUCB++ algorithm has a sample complexity of the form of $\left(\sum_{k \in \mathcal{K} \setminus \{k^*\}} \frac{2\zeta_1}{\Delta_k^2} \log\left(\frac{\log\left(\frac{1}{\Delta_k^2}\right)}{\delta} \right) + \frac{2\zeta_2}{\Delta_{k^*}^2} \log\left(\frac{K \log\left(\frac{1}{\Delta_{k^*}^2}\right)}{\delta} \right) \right)$. The LUCB++ algorithm avoids the $\log(K)$ term in the sample complexity for the sub-optimal arms and has it only for the optimal arm k^*. Due to this, it is seen that LUCB++ slightly outperforms the LUCB algorithm empirically. In our next section, we propose the C-LUCB++ algorithm, which is a heuristic extension of LUCB++ to the correlated bandit setting and show that it finds the optimal arm with probability at least $1 - \delta$.

Dependency with K: In our sample complexity results, the dependence with respect to K is loose. For our theoretical results, we focus on studying the dependence of sample complexity on δ in this chapter. In Section 4.7, we show that even when $\delta = 0.1$ (i.e., a moderate confidence regime), our proposed algorithms outperform the classical bandit algorithms (See Figure 4.3).

4.6 Variants of C-LUCB

In our proposed C-LUCB algorithm, at each round we sample two arms $m_1(t), m_2(t)$, where $m_1(t) = \arg\max_{k \in \mathcal{A}_t} I_k(t)$ and $m_2(t) = \arg\max_{k \in \mathcal{A}_t \setminus \{m_1\}} \min(\tilde{U}_{k,k}(\delta/2K), I_k(t))$. A sampling such as this allowed us to show $1-\delta$ correctness of the algorithm (Theorem 8) and analyse its sample complexity (Theorem 9). In this section, we explore two other algorithms, that we call maxmin-LUCB and 2-LUCB, that sample different $m_1(t)$ and $m_2(t)$ at round t, but have the same elimination and stopping criteria as that of C-LUCB. In Table 4.4, we contrast their sampling strategy with respect to C-LUCB. While we are able to show that both maxmin-LUCB and 2-LUCB algorithm will stop with the best-arm with probability at least $1-\delta$, we are unable to provide a sample complexity result for them.

We also evaluated the empirical performance of maxmin-LUCB and 2-LUCB on a real-world recommendation dataset, and found their empirical performance to be similar to C-LUCB. We chose to use C-LUCB as our proposed algorithm as it is possible to provide theoretical guarantees as in Theorem 8 and Theorem 9. Moreover, we find its empirical performance to be superior than classical bandit algorithms in correlated bandit settings, as we illustrate through our experiments in the next section.

4.6.1 C-LUCB++: Heuristic extension of LUCB++

The LUCB++ algorithm as illustrated in Section 4.3, is able to improve upon LUCB, by modifying its stopping criteria and in its sampling of $m_1(t)$ and $m_2(t)$. We propose an extension, C-LUCB++, that extends the LUCB++ algorithm to the correlated bandit setting. The comparison of C-LUCB++ and LUCB++ in its sampling, elimination and stopping criteria is presented in Table 4.2. While we are able to show that the C-LUCB++ stops with the best arm with probability at least $1-\delta$ in Section 4.9.7, analysing its sample complexity remains an open problem. We compare the performance of C-LUCB++, with C-LUCB, LUCB, Racing and lil'UCB algorithms extensively through our experiments on Movielens and Goodreads datasets in the next section.

4.7 Experiments

We now evaluate the performance of our proposed C-LUCB and C-LUCB++ algorithms in a real-world setting. By comparing the performance against classical best-arm identification algorithms on the MOVIELENS and GOODREADS datasets, we show that our proposed algorithms are able to exploit correlation to identify the best-arm in fewer samples. All results reported in our chapter are presented after conducting 10 independent trials and computing their average. Additionally, in all our plots we show the error bars of

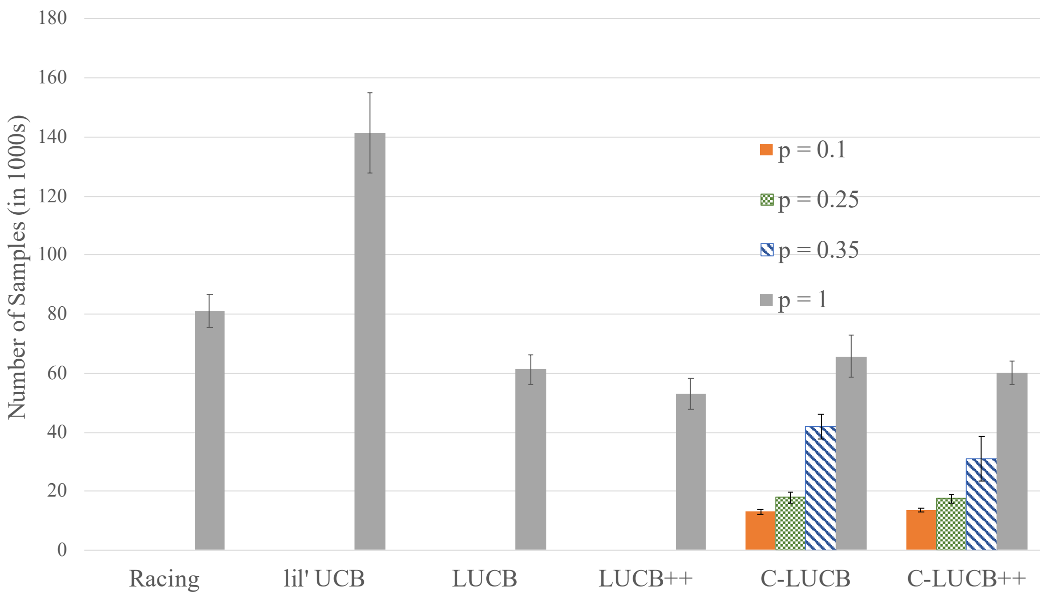


Figure 4.4: Number of samples drawn by Racing, lil'UCB, LUCB, LUCB++, C-LUCB and C-LUCB++ to identify the best movie genre out of 18 possible genres in the Movielens dataset. Here, p represents the fraction of pseudo-reward entries that are replaced by the maximum possible reward (i.e., 5). When p is small, there is more correlation information available that our proposed C-LUCB and C-LUCB++ algorithms exploit to reduce the number of samples needed to identify the best movie genre. When $p = 1$, there is no correlation information available, in which case our proposed C-LUCB and C-LUCB++ algorithms have a performance similar to LUCB and LUCB++ respectively.

width 2σ, where σ is the standard deviation in the number of samples drawn by an algorithm across the 10 independent trials.

4.7.1 Experiments on the MovieLens dataset

The MovieLens dataset [19] contains a total of 1M ratings for a total of 3883 Movies rated by 6040 Users. Each movie is rated on a scale of 1-5 by the users. Moreover, each movie is associated with one (and in some cases, multiple) genres. For our experiments, of the possibly several genres associated with each movie, one is picked uniformly at random. To perform our experiments, we split the data into two parts, with the first half containing ratings of the users who provided the most number of ratings. This half is used to learn the pseudo-reward entries, the other half is the test set which is used to evaluate the performance of the proposed algorithms. Doing such a split ensures that the rating distribution is different in the training and test data.

Best Genre identification. In this experiment, our goal is to identify the most preferred genre among the 18 different genre in the test population in fewest possible samples. The pseudo-reward entry $s_{\ell,k}(r)$ is evaluated by taking the empirical average of the ratings of genre ℓ that are rated by the users who rated genre k as r. As in practice, all such pseudo-reward entries might not be available, we randomly replace p-fraction of the pseudo-reward entries by maximum possible reward, i.e., 5. We then run our best-arm identification algorithms on the test data to identify the best-arm with 99% confidence. Figure 4.4 shows the average samples taken by C-LUCB and C-LUCB++ algorithm relative to the classical best-arm identification algorithms for different value of p (the fraction of pseudo-reward entries that are removed). We see that C-LUCB and C-LUCB++ algorithms significantly outperform all Racing, lil'UCB, LUCB and LUCB++ algorithms for $p = 0.1, 0.25, 0.35$ as they are able to exploit the correlations present in the problem to identify the best arm in a faster manner.

In the scenario where all pseudo-reward entries are unknown, i.e., $p = 1$, we see that the performance of C-LUCB is only slightly worse than that of LUCB algorithm. This is due to the construction of slightly wide confidence interval $B(n_k, \delta/2K)$ for the C-LUCB algorithm relative to LUCB algorithm that uses $B(n_k, \delta/K)$. We also see that in this scenario, LUCB++ and C-LUCB++ algorithm (which is an extension of LUCB++) outperform C-LUCB, which is due to the known superiority of LUCB++ over LUCB [69, 35].

Variation with δ. We then study the performance of the best-arm identification algorithms for different value of δ. In Figure 4.3, we plot the number of samples required by C-LUCB and C-LUCB++ to identify the best arm with $90\%, 94\%, 98\%$ and 99% confidence, with $p = 0.2$ (i.e., 20% of pseudo-reward entries are replaced by 5). As C-LUCB and C-LUCB++ are able to make use of the available correlation information, we see our proposed algorithms require fewer samples than the Racing, lil'UCB, LUCB and LUCB++ algorithms in each of the four settings.

4.7.2 Experiments on the Goodreads dataset

The Goodreads dataset [20] contains the ratings for 1,561,465 books by a total of 808,749 users. Each rating is on a scale of 1-5. For our experiments, we only consider the poetry section and focus on the goal of identify the most liked poem for the population. The poetry dataset has 36,182 different poems rated by 267,821 different users. We do the pre-processing of goodreads dataset in the same manner as that of the MovieLens dataset, by splitting the dataset into two halves, train and test. The train dataset contains the ratings of the users with most number of recommendations.

Best book identification. We consider the 25 most rated poetry books in the dataset and aim to identify the best book in fewest possible samples with 99% confidence. After obtaining the pseudo-reward entries from the training data, we replace p fraction of the entries with the highest possible reward (i.e., 5) as some pseudo-rewards may be unknown in practice. To account for the fact that these pseudo-reward entries may be noisy in practice, we add a safety buffer of 0.1 to each of the pseudo-reward entry $s_{\ell,k}(r)$; i.e., we set the pseudo-reward to be empirical conditional mean (obtained from training data) *plus* the safety

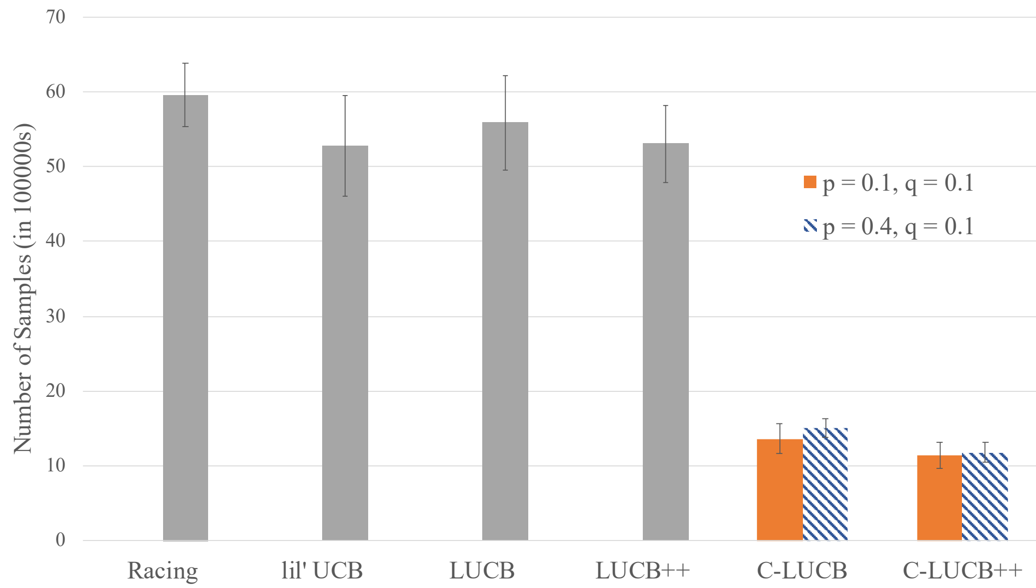


Figure 4.5: Number of samples needed by Racing, lil'UCB, LUCB, LUCB++, C-LUCB and C-LUCB++ to identify the best poem out of the set of 25 poem books in the Goodreads dataset. Here p represents the fraction of pseudo-rewards that are replaced by maximum possible reward and $q = 0.1$ is added to each pseudo-reward entry to account for the fact that pseudo-reward entries may be noisy. Our proposed C-LUCB and C-LUCB++ utilize correlation information and require significantly less samples than the classical best-arm identification algorithms.

buffer $q = 0.1$. We perform experiment on the test data and compare the number of samples obtained for different algorithms in Figure 4.5 for two different values of p. We see that in both the cases, our C-LUCB and C-LUCB++ algorithms outperform other algorithms as they are able to exploit the correlations in the rewards.

4.8 Concluding Remarks

In this work, we studied a new multi-armed bandit problem, where rewards corresponding to different arms are correlated to each other and this correlation is known and modeled through the knowledge of pseudo-rewards. These pseudo-rewards are *loose* upper bounds on conditional expected rewards and can be evaluated in practical scenarios through controlled surveys or from domain expertise. We then extended an LUCB based approach to perform best-arm identification in the correlated bandit setting. Our approach makes use of the pseudo-rewards to reduce the number of samples taken before stopping. In particular, our approach avoids the sampling of non-competitive arms leading to a stark reduction in sample complexity. The theoretical superiority of our proposed approach is reflected in practical scenarios. Our experimental results on Movielens and Goodreads recommendation dataset show that the presence of correlation, when exploited by our C-LUCB approach, can lead to significant reduction in the number of samples required to identify the best-arm with probability $1-\delta$.

This work opens up several interesting future directions, including but not limited to the following:

PAC-C-LUCB: In this work, we explored the problem of identifying the best-arm with probability $1-\delta$. A closely related problem is to find a PAC (probably approximately correct) algorithm, that identifies an arm which is within ϵ from μ_{k^*} with probability at least $1-\delta$. We believe such an algorithm can be constructed by modifying the elimination and stopping criteria of C-LUCB algorithm. More specifically, if one compares $U_k(n_k,\delta)+\epsilon$ v/s $\max_{k\in\mathcal{A}_t} L_k(n_k,\delta)$ in the C-LUCB's elimination criteria, it may be possible to design and analyse a PAC algorithm in the correlated multi-armed bandit setting.

Using Pseudo-Lower bounds: We assume in our work that only upper bounds on conditional expected rewards, in the form of pseudo-upper-bounds, are known to the player. In practical settings, it may also be possible to obtain pseudo-lower-bounds, that may allow us to know information about lower bound on conditional expected reward. In presence of such knowledge, we believe C-LUCB algorithm will need a modification in its definition of lower confidence bound $L_k(n_k,\delta)$. By defining a crossLCB index $L_{\ell,k}(n_k,\delta)$, equivalent to crossUCB index for upper bound, we can re-define $L_k = \max L_{\ell,k}$. This new definition of the lower confidence bound index can help us to incorporate cases where pseudo-lower bounds are also known.

Top m arms identification: Throughout this work, our focus was to identify just the optimal arm from the set of K arms. Another similar problem is to come up with an approach to find the best m arms from the set of K arms. It is an interesting direction to explore in the correlated-multi armed bandit setting. We believe such a problem would be even more interesting if the pseudo-lower bounds are known. An open problem is to extend a C-LUCB like approach to identify the best m arms from the set of K arms.

Lower bound and optimal solution: While our proposed approach shows promising empirical performance and has some theoretical guarantees, it may not be the optimal solution for the correlated bandit problem studied in this chapter. Studying a lower bound and correspondingly an optimal solution to this problem remains an open problem.

4.9 Full proofs

4.9.1 Standard Results from Previous Works

Fact 3 (Hoeffding's inequality)**.** *Let $Z_1, Z_2 \ldots Z_n$ be i.i.d random variables bounded between $[a,b] : a \leq Z_i \leq b$, then for any $\delta > 0$, we have*

$$\Pr\left(\left|\frac{\sum_{i=1}^{n} Z_i}{n} - \mathbb{E}\left[Z_i\right]\right| \geq \delta\right) \leq \exp\left(\frac{-2n\delta^2}{(b-a)^2}\right).$$

Lemma 18 (Standard result used in bandit literature)**.** *If $\hat{\mu}_{k,n_k(t)}$ denotes the empirical mean of arm k by sampling arm k $n_k(t)$ times through any algorithm and μ_k denotes the mean reward of arm k, then we have*

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) \leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right).$$

Proof. Let $Z_1, Z_2, ...Z_t$ be the reward samples of arm k drawn separately. If the algorithm chooses to sample arm k for m^{th} time, then it observes reward Z_m. Then the probability of observing the event $\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1$ can be upper bounded as follows,

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) = \Pr\left(\left(\frac{\sum_{i=1}^{n_k(t)} Z_i}{n_k(t)} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{4.15}$$

$$\leq \Pr\left(\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{4.16}$$

$$\leq \Pr\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right) \tag{4.17}$$

$$\leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right). \tag{4.18}$$

□

Lemma 19 (From Proof of Theorem 1 in [32])**.** *The probability that the mean reward of arm k, i.e., μ_k, is greater than the pseudoUCB index of arm k with respect to arm k, i.e., $I_{k,k} = \hat{\mu}_k + \sqrt{\frac{2\log t}{n_k(t)}}$ is upper bounded by t^{-3}.*

$$\Pr(\mu_k > I_{k,k}(t)) \leq t^{-3}.$$

Observe that this bound does not depend on the number $n_k(t)$ of times arm k is sampled and only depends on t.

Proof. This proof follows directly from [32]. We present the proof here for completeness as we use this frequently in the chapter.

$$\Pr(\mu_k > I_{k,k}(t)) = \Pr\left(\mu_k > \hat{\mu}_{k,n_k(t)} + \sqrt{\frac{2\log t}{n_k(t)}}\right) \tag{4.19}$$

$$\leq \sum_{m=1}^{t} \Pr\left(\mu_k > \hat{\mu}_{k,m} + \sqrt{\frac{2\log t}{m}}\right) \tag{4.20}$$

$$= \sum_{m=1}^{t} \Pr\left(\hat{\mu}_{k,m} - \mu_k < -\sqrt{\frac{2\log t}{m}}\right) \tag{4.21}$$

$$\leq \sum_{m=1}^{t} \exp\left(-2m\frac{2\log t}{m}\right) \tag{4.22}$$

$$= \sum_{m=1}^{t} t^{-4} \tag{4.23}$$

$$= t^{-3}. \tag{4.24}$$

where (4.20) follows from the union bound and is a standard approach (Lemma 18) to deal with random variable $n_k(t)$. We use this approach repeatedly in the proofs. We have (4.22) from the Hoeffding's inequality. Note that if the empirical mean μ_k is replaced by the empirical pseudo reward of arm k with respect to arm ℓ, i.e., $\phi_{k,\ell}$ and $I_{k,k}(t)$ by the expected pseudo reward of arm k with respect to arm ℓ, i.e., $I_{k,\ell}(t) = \hat{\phi}_{k,\ell} + \sqrt{\frac{2\log t}{n_\ell(t)}}$. Then we get that $\Pr(\phi_{k,\ell} > I_{k,\ell}(t)) \leq t^{-3}$ using the same steps as presented above. □

4.9.2 Intermediate lemmas for proving bounds on samples obtained through non-competitive arms

Lemma 20. *Let $I_k(t)$ denote the pseudoUCB index of arm k at round t, and μ_k denote the mean reward of that arm. Then, we have*

$$\Pr(\mu_k > I_k(t)) \leq Kt^{-3}.$$

Similar to Lemma 19, this bound does not depend on the number of times arm k is sampled till round t (i.e., $n_k(t)$) and only depends on the round t and the total number of arms K. Recall that $I_\ell(t) = \min_k I_{\ell,k}(t)$, where $I_{\ell,k}(t)$ is PseudoUCB index of arm ℓ with respect to arm k defined in (4.8).

Proof. This proof follows in the same way as that of Lemma 19.

$$\Pr(\mu_k > I_k(t)) = \Pr\left(\mu_k > \min_\ell \hat{\phi}_{k,\ell} + \sqrt{\frac{2\log t}{n_\ell(t)}}\right) \tag{4.25}$$

$$\leq \sum_{\ell \in \mathcal{K}} \Pr\left(\mu_k > \hat{\phi}_{k,\ell} + \sqrt{\frac{2\log t}{n_\ell(t)}}\right) \tag{4.26}$$

$$\leq \sum_{\ell \in \mathcal{K}} \Pr\left(\phi_{k,\ell} > \hat{\phi}_{k,\ell} + \sqrt{\frac{2\log t}{n_\ell(t)}}\right) \tag{4.27}$$

$$\leq \sum_{\ell \in \mathcal{K}} t^{-3} \tag{4.28}$$

$$= Kt^{-3}. \tag{4.29}$$

□

We have (4.25) from the definition of $I_k(t)$. Inequality (4.27) follows from the fact that $\phi_{k,\ell} \geq \mu_k$. We get (4.28) follows from the hoeffding's inequality combined with the union bound (Lemma 19).

Lemma 21. *If $k \neq k^*$ is a non-competitive arm i.e., $k \notin \mathcal{C}$ and has a pseudo-gap $\tilde{\Delta}_{k,k^*} > 0$, then,*

$$\Pr((m_1(t) = k \cup m_2(t) = k), n_{k^*}(t) \geq t/2K, \mathcal{W}, \mathcal{E}) \leq 2(K+1)t^{-3}. \quad \forall t > t_0,$$

where $t_0 = \inf\left\{\tau \geq 2 : \Delta_{min} \geq 4\sqrt{\frac{2K\log\tau}{\tau}}\right\}$ *and $\mathcal{W}$ denotes the event that $m_1(t), m_2(t) \neq k^*$.*

Proof. We now bound this probability as,

$$\Pr\left((m_1(t) = k \cup m_2(t) = k), n_{k^*}(t) \geq \frac{t}{2K}, \mathcal{E}, \mathcal{W}\right)$$

$$\leq \Pr\left(m_1(t) = k, n_{k^*}(t) \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) + \Pr\left(m_2(t) = k, n_{k^*}(t) \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.30}$$

$$\leq \Pr\left(k = \arg\max_{k \in \mathcal{A}_t} I_\ell(t), n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) + \Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.31}$$

$$\leq \Pr\left(\hat{\phi}_{k,k^*} + \sqrt{\frac{2\log t}{n_{k^*}(t)}} \geq I_{k^*}(t), n_{k^*} \geq \frac{t}{2K}\right) + \Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.32}$$

$$\leq \Pr\left(\hat{\phi}_{k,k^*} + \sqrt{\frac{2\log t}{n_{k^*}(t)}} \geq I_{k^*}(t), \mu_{k^*} < I_{k^*} n_{k^*} \geq \frac{t}{2K}\right) + \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) +$$
$$\Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.33}$$

$$\leq \Pr\left(\hat{\phi}_{k,k^*} + \sqrt{\frac{2\log t}{n_{k^*}(t)}} \geq \mu_{k^*}, n_{k^*} \geq \frac{t}{2K}\right) + Kt^{-3} + \Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.34}$$

$$= \Pr\left(\hat{\phi}_{k,k^*} - \phi_{k,k^*} \geq \mu_{k^*} - \sqrt{\frac{2\log t}{n_{k^*}(t)}}, n_{k^*} \geq \frac{t}{2K}\right) + Kt^{-3} +$$
$$\Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.35}$$

$$\leq t\exp\left(-2\frac{t}{2K}\left(\mu_{k^*} - \phi_{k,k^*} - \sqrt{\frac{4K\log t}{t}}\right)^2\right) + Kt^{-3} + \Pr\left(m_2(t) = k, \mathcal{E}\right) \tag{4.36}$$

$$\leq t^{-3}\exp\left(\Delta_{\min}^2 - 2\Delta_{\min}\sqrt{\frac{4K\log t}{t}}\right) + Kt^{-3} + \Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \tag{4.37}$$

$$\leq t^{-3} + Kt^{-3} + \Pr\left(m_2(t) = k, \mathcal{W}, \mathcal{E}\right) \quad \forall t > t_0 \tag{4.38}$$

Here (4.34) follows from Lemma 20. Inequality (4.36) follows as a result of hoeffding bound and the union bound, as n_{k^*} can take any value between $\frac{t}{2K}$ and t (Lemma 18). We get (4.37) as $\phi_{k,k^*} < \mu_{k(2)}$ as the arm k is non-competitive.

We now bound $\Pr((m_2(t) = k, \mathcal{E})$ separately. Under $\mathcal{E}$, the crossUCB index $\tilde{U}_{k^*,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right)$ is larger than μ_{k^*}. Using similar steps as done for the first term we now evaluate the upper bound on the probability that arm k to be selected as $m_2(t)$ at round t,

$$\Pr\left(m_2(t) = k, n_{k^*} \geq \frac{t}{2K}, \mathcal{W}, \mathcal{E}\right) \leq \tag{4.39}$$

$$\leq \Pr\left(\hat{\phi}_{k,k^*} + \sqrt{\frac{2\log t}{n_{k^*}(t)}} \geq \mu_{k^*}, I_{k^*}(t) > \mu_{k^*}, n_{k^*} \geq \frac{t}{2K}\right) + \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \tag{4.40}$$

$$\leq t^{-3} + Kt^{-3} \tag{4.41}$$

$$\tag{4.42}$$

Combining this with (4.38), we get the result of Lemma 21. □

Lemma 22. *If $\Delta_{min} \geq 4\sqrt{\frac{2K\log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\Pr(m_1(t) = k, n_k(t) \geq s, \mathcal{E}) \leq 2(K+1)t^{-3} \quad \textit{for } s > \frac{t}{2K}, \forall t > t_0.$$

Proof. By noting that $m_1(t) = k$ corresponds to arm k having the highest pseudoUCB index among the set of active arms at round t (denoted by $\mathcal{A}_t$), we have,

$$\Pr(m_1(t) = k, n_k(t) \geq s, \mathcal{E}) = \Pr(I_k(t) = \arg\max_{k' \in \mathcal{A}_t} I_{k'}(t), n_k(t) \geq s, \mathcal{E}) \tag{4.43}$$

$$\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s\right). \tag{4.44}$$

Here (4.44) follows from the fact that under $\mathcal{E}$, k^* is always in $\mathcal{A}_t$ (Section 4.9.6).

$$\Pr(I_k(t) > I_{k^*}(t), n_k(t) \geq s) =$$

$$\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) +$$

$$\Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s | \mu_{k^*} > I_{k^*}(t)\right) \times \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \tag{4.45}$$

$$\leq \Pr\left(I_k(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + \Pr\left(\mu_{k^*} > I_{k^*}(t)\right) \tag{4.46}$$

$$\leq \Pr\left(I_{k,k}(t) > I_{k^*}(t), n_k(t) \geq s, \mu_{k^*} \leq I_{k^*}(t)\right) + Kt^{-3} \tag{4.47}$$

$$= \Pr\left(I_{k,k}(t) > \mu_{k^*}, n_k(t) \geq s\right) + Kt^{-3} \tag{4.48}$$

$$= \Pr\left(\hat{\mu}_k(t) + \sqrt{\frac{2\log t}{n_k(t)}} > \mu_{k^*}, n_k(t) \geq s\right) + Kt^{-3} \tag{4.49}$$

$$= \Pr\left(\hat{\mu}_k(t) - \mu_k > \mu_{k^*} - \mu_k - \sqrt{\frac{2\log t}{n_k(t)}}, n_k(t) \geq s\right) + Kt^{-3} \tag{4.50}$$

$$= \Pr\left(\frac{\sum_{\tau=1}^{t} \mathbb{1}_{\{k_\tau = k\}} r_\tau}{n_k(t)} - \mu_k > \Delta_k - \sqrt{\frac{2\log t}{n_k(t)}}, n_k(t) \geq s\right) + Kt^{-3} \tag{4.51}$$

$$\leq t\exp\left(-2s\left(\Delta_k - \sqrt{\frac{2\log t}{s}}\right)^2\right) + Kt^{-3} \tag{4.52}$$

$$\leq t^{-3}\exp\left(-2s\left(\Delta_k^2 - 2\Delta_k\sqrt{\frac{2\log t}{s}}\right)\right) + Kt^{-3} \tag{4.53}$$

$$\leq 2(K+1)t^{-3} \quad \text{for all } t > t_0. \tag{4.54}$$

We have (4.45) holds because of the fact that $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$, Inequality (4.47) follows from Lemma 20 and from the fact that $I_k(t) = \min_\ell I_{k,\ell}(t)$. From the definition of $I_{k,k}(t)$ we have (4.49). Inequality (4.52) follows from Hoeffding's inequality and the term t before the exponent in (4.52) arises as the random variable $n_k(t)$ can take values from s to t (Lemma 18). Inequality (4.54) follows from the fact that $s > \frac{t}{2K}$ and $\Delta_k \geq 4\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$.

□

Lemma 23. *Let $n_k^{m_1}(t)$ denote the number of times arm k has been sampled as $m_1(t)$ till round t. If $\Delta_{min} \geq 4\sqrt{\frac{2K \log t_0}{t_0}}$ for some constant $t_0 > 0$, then,*

$$\Pr\left(n_k^{m_1}(t) > \frac{t}{K}, \mathcal{E}\right) \leq \frac{(K+1)^3}{t^2} \quad \forall t > Kt_0.$$

Proof. We expand $\Pr\left(n_k(t) > \frac{t}{K}\right)$ as,

$$\Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) = \Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E} \middle| n_k^{m_1}(t-1) \geq \frac{t}{K}, \mathcal{E}\right) \Pr\left(n_k^{m_1}(t-1) \geq \frac{t}{K}, \mathcal{E}\right) + \Pr\left(m_1(t) = k, n_k^{m_1}(t-1) = \frac{t}{K} - 1, \mathcal{E}\right) \tag{4.55}$$

$$\leq \Pr\left(n_k^{m_1}(t-1) \geq \frac{t}{K}, \mathcal{E}\right) + \Pr\left(m_1(t) = k, n_k^{m_1}(t-1) = \frac{t}{K} - 1, \mathcal{E}\right) \tag{4.56}$$

$$\leq \Pr\left(n_k^{m_1}(t-1) \geq \frac{t}{K}, \mathcal{E}\right) + (2K+2)(t-1)^{-3} \quad \forall (t-1) > t_0. \tag{4.57}$$

Here, (4.57) follows from Lemma 22. This gives us

$$\Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) - \Pr\left(n_k^{m_1}(t-1) \geq \frac{t}{K}, \mathcal{E}\right) \leq (2K+2)(t-1)^{-3}, \quad \forall (t-1) > t_0.$$

Now consider the summation

$$\sum_{\tau = \frac{t}{K}}^{t} \Pr\left(n_k^{m_1}(\tau) \geq \frac{t}{K}, \mathcal{E}\right) - \Pr\left(n_k^{m_1}(\tau - 1) \geq \frac{t}{K}, \mathcal{E}\right) \leq \sum_{\tau = \frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3}.$$

This gives us,

$$\Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) - \Pr\left(n_k^{m_1}\left(\frac{t}{K} - 1\right) \geq \frac{t}{K}, \mathcal{E}\right) \leq \sum_{\tau = \frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3}.$$

Since $\Pr\left(n_k^{m_1}\left(\frac{t}{K}-1\right) \geq \frac{t}{K}, \mathcal{E}\right) = 0$, we have,

$$\Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) \leq \sum_{\tau=\frac{t}{K}}^{t} (2K+2)(\tau-1)^{-3} \tag{4.58}$$

$$\leq (K+1)\left(\frac{t}{K}-2\right)^{-2} \quad \forall t > Kt_0. \tag{4.59}$$

The last step (4.59) follows from the fact that $\sum_{\tau=t/K}^{t}(\tau-1)^{-3} \leq \int_{\tau=t/K-1}^{\infty}(\tau-1)^{-3}$. □

4.9.3 Probability of sampling a non-competitive arm at round t

For ease of presentation we denote $\mathcal{W}$ to be the event that $m_1(t), m_2(t) \neq k^*$.

Lemma 24. *The probability of sampling a non-competitive arm at round t, jointly with the event $\mathcal{E}$, is bounded as*

$$\Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}\right) \leq \frac{2(K+1)K}{t^3} + \frac{K(K+1)^3}{t^2} \quad \forall t > Kt_0.$$

Proof.

$$\begin{aligned}
&\Pr((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{E}) = \\
&\Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}, n_{k^*}(t) \geq \frac{t}{K}\right) + \\
&\qquad \Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}, n_{k^*}(t) < \frac{t}{K}\right) &(4.60)\\
&\leq \Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}, n_{k^*}(t) \geq \frac{t}{K}\right) + \Pr\left(n_{k^*}(t) < \frac{t}{K}, \mathcal{E}\right) &(4.61)\\
&\leq \Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}, n_{k^*}(t) \geq \frac{t}{K}\right) + \Pr\left(n_{k^*}^{m_1}(t) < \frac{t}{K}, \mathcal{E}\right) &(4.62)\\
&\leq \Pr\left((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), \mathcal{W}, \mathcal{E}, n_{k^*}(t) \geq \frac{t}{K}\right) + \sum_{k \neq k^*} \Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) &(4.63)\\
&\leq \sum_{k \notin \mathcal{C}} \Pr\left((m_1(t) = k \cup m_2(t) = k), \mathcal{W}, \mathcal{E}, n_{k^*}(t) \geq \frac{t}{K}\right) + \sum_{k \neq k^*} \Pr\left(n_k^{m_1}(t) \geq \frac{t}{K}, \mathcal{E}\right) &(4.64)\\
&\leq \frac{2(K+1)K}{t^3} + \frac{K(K+1)^3}{t^2} \quad \forall t > Kt_0 &(4.65)
\end{aligned}$$

In (4.62), $n_{k^*}^{m_1}(t)$ denotes the number of times arm k^* was samples as $m_1(t)$ till round t. As $n_{k^*}^{m_1}(t) < n_{k^*}(t)$, we have (4.62). The last step follows from Lemma 21 and Lemma 23. □

4.9.4 Intermediate steps to analyse samples obtained from competitive arms

For $k \neq k^*$, define τ_k to be the first integer such that $B\left(n_k, \frac{\delta}{2K}\right) < \frac{\Delta_k}{4}$ and define $\tau_{k^*} = \tau_{k^{(2)}}$. We call an arm k to be GOOD at round t, if $B\left(n_k, \frac{\delta}{2K}\right) \leq \frac{\Delta_k}{4}$, i.e., an arm is GOOD if it has been sampled *significant* number of times till round t, i.e., $n_k(t) \geq \tau_k$. Otherwise, the arm is called BAD. We denote μ^{ref} as $\frac{\mu_{k^*}+\mu_{k(2)}}{2}$, i.e., the average of the mean reward of best and second best arm. We will first show that an arm $k \neq k^*$ being GOOD implies that its psuedoUCB index is below μ^{ref}, i.e., $n_k > \tau_k \Rightarrow \tilde{U}_{k,k}\left(n_k, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$. Consider $\tilde{U}_{k,k}\left(n_k, \frac{\delta}{2K}\right)$ for $k \neq k^*, n_k > \tau_k$. Under $\mathcal{E}$, we have

$$\hat{\mu}_k + B\left(n_k, \frac{\delta}{2K}\right) \leq \mu_k + 2B\left(n_k, \frac{\delta}{2K}\right) \tag{4.66}$$

$$= \mu^{\text{ref}} + 2B\left(n_k, \frac{\delta}{2K}\right) + \frac{(\mu_k - \mu_{k^*}) + (\mu_k - \mu_{k(2)})}{2} \tag{4.67}$$

$$\leq \mu^{\text{ref}} + 2B\left(n_k, \frac{\delta}{2K}\right) - \frac{\Delta_k}{2} \tag{4.68}$$

$$\leq \mu^{\text{ref}} \tag{4.69}$$

Here (4.66) follows from the fact that, under $\mathcal{E}$, $\hat{\mu}_k \leq \mu_k + B_k\left(n_k, \frac{\delta}{2K}\right)$. The last step follows as arm k is GOOD, i.e., $B\left(n_k, \frac{\delta}{2K}\right) \leq \frac{\Delta_k}{4}$.

Using a similar argument for k^*, we can prove that Arm k^* being GOOD, i.e., $n_{k^*} > \tau_{k^*} \Rightarrow L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \mu^{\text{ref}}$,. In addition to this, $n_{k^*} > \tau_{k^*}$ (i.e., Arm k^* being GOOD), also implies that $\tilde{U}_{k,k^*} < \mu^{\text{ref}}$ for $k \notin \mathcal{C}$ as we present below. Under $\mathcal{E}$, we have the bound on $\tilde{U}_{k,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) = \hat{\phi}_{k,k^*} + B\left(n_{k^*}, \frac{\delta}{2K}\right)$, as follows,

$$\hat{\phi}_{k,k^*} + B\left(n_{k^*}, \frac{\delta}{2K}\right) \leq \phi_{k,k^*} + 2B\left(n_{k^*}, \frac{\delta}{2K}\right) \tag{4.70}$$

$$\leq \mu_{k(2)} + 2B\left(n_{k^*}, \frac{\delta}{2K}\right) \tag{4.71}$$

$$\leq \mu_{k(2)} + 2\frac{\Delta_{\min}}{4} \tag{4.72}$$

$$\leq \mu^{\text{ref}} \tag{4.73}$$

The inequality (4.71) follows from the fact that arm $k \notin \mathcal{C}$, i.e., $\phi_{k,k^*} < \mu_{k(2)}$. We now use this observation to list four possible scenarios under which algorithm does not stop and bound each individual term to prove the statement of Theorem 9.

Define $\mathcal{R}(t)$ to be the event that $I_{k^*}(t) < \mu_{k^*}$, i.e., $\mathcal{R}(t) = \{I_{k^*}(t) > \mu_{k^*}\}$. By Lemma 20, $\Pr(\mathcal{R}(t)) \leq Kt^{-3}$.

Lemma 25. *If the algorithm has not stopped at round t and the event $\mathcal{E}$ holds true, at least one of the following occurs*

1. *Event $\mathcal{R}(t)$ does not occur,*
2. *$m_1(t)$ or $m_2(t)$ is Non-Competitive and $m_1(t), m_2(t) \neq k^*$*
3. *($m_1(t) = k^*$ is BAD and $m_2(t) \notin \mathcal{C}$) or ($m_2(t) = k^*$ is BAD and $m_1(t) \notin \mathcal{C}$)*
4. *$m_1(t), m_2(t) \in \mathcal{C}$ and either $m_1(t)$ is BAD or $m_2(t)$ is BAD.*

Proof. We prove this by contradiction. We consider the event that all the four cases listed above do not occur jointly and show that such a situation cannot occur if algorithm has not stopped till round t under $\mathcal{E}$. The proof technique is inspired from the analysis done in [31] but needed some modification to prove the result for C-LUCB algorithm in a correlated bandit environment. Let's break down the scenario where all of the four events listed in Lemma 25 do not occur and look at each of them individually.

Case 1:

$\{m_1(t) = k^*, m_1(t) \text{ is GOOD }\} \cap \{m_2(t) \neq k^*, m_2(t) \in \mathcal{C}, m_2(t) \text{ is GOOD }\} \cap \mathcal{R}(t) \cap \{t < \mathcal{T}\}$.

We note the following two things in this case,

1. $m_1(t) = k^*$ is GOOD $\Rightarrow L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \mu^{\text{ref}}$.
2. $m_2(t) = \ell \neq k^*$ is GOOD $\Rightarrow \tilde{U}_{\ell,\ell}\left(n_\ell, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$.

As we have, $L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \tilde{U}_\ell\left(n_\ell, \frac{\delta}{2K}\right)$ at round t, arm ℓ cannot belong the the set of active arms $\mathcal{A}_t$ and hence cannot be selected as $m_2(t)$.

Case 2:

$\{m_1(t) \neq k^*, m_1(t) \in \mathcal{C}, m_1(t) \text{ is GOOD }\} \cap \{m_2(t) = k^*, m_2(t) \text{ is GOOD }\} \cap \mathcal{R}(t) \cap \{t < \mathcal{T}\}$.

In case 2, we make the following observations

1. $m_1(t) = \ell \neq k^*$ is GOOD $\Rightarrow \tilde{U}_{\ell,\ell}\left(n_\ell, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$.
2. $m_2(t) = k^*$ is GOOD $\Rightarrow L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \mu^{\text{ref}}$.

As we have, $L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \tilde{U}_\ell\left(n_\ell, \frac{\delta}{2K}\right)$ at round t, arm ℓ cannot belong the the set of active arms $\mathcal{A}_t$ and hence cannot be selected as $m_1(t)$.

Case 3:

$\{m_1(t) \neq k^*, m_1(t) \in \mathcal{C}, m_1(t) \text{ is GOOD}\} \cap \{m_2(t) \neq k^*, m_2(t) \in \mathcal{C}, m_2(t) \text{ is GOOD}\} \cap \mathcal{R}(t) \cap \{t < \mathcal{T}\}$.

For case 3, we see that

1. $m_1(t) = \ell_1 \neq k^*$ is GOOD $\Rightarrow \tilde{U}_{\ell_1,\ell_1}\left(n_{\ell_1}, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$.
2. $m_2(t) = \ell_2 \neq k^*$ is GOOD $\Rightarrow \tilde{U}_{\ell_2,\ell_2}\left(n_{\ell_2}, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$, it further implies that $\min\left(I_{\ell_2}(t), \tilde{U}_{\ell_2,\ell_2}\left(n_{\ell_2}, \frac{\delta}{2K}\right)\right) \leq \mu^{\text{ref}}$.

As arm k^* is not selected, it implies that either $I_{k^*}(t) \leq \mu^{\text{ref}}$ or $\tilde{U}_{k^*,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) \leq \mu^{\text{ref}}$. By $\mathcal{R}(t)$, $I_{k^*}(t) \geq \mu_{k^*} > \mu^{\text{ref}}$ and with event $\mathcal{E}$, $\tilde{U}_{k^*,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \mu_{k^*} > \mu^{\text{ref}}$. This shows that case 3 cannot occur and leads to a contradiction.

Case 4:

$\{(m_1(t) = k^* \text{ is GOOD}, m_2(t) = \ell \notin \mathcal{C}) \cup (m_2(t) = k^* \text{ is GOOD}, m_1(t) = \ell \notin \mathcal{C})\} \cap \mathcal{R}(t) \cap \{t < \mathcal{T}\}$.

For Case 4, we see from (4.69), (4.73) that

1. k^* is GOOD $\Rightarrow L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) > \mu^{\text{ref}}$.
2. k^* is GOOD $\Rightarrow \tilde{U}_{\ell,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) < \mu^{\text{ref}}$.

As $\tilde{U}_\ell\left(\frac{\delta}{2K}\right) < \tilde{U}_{\ell,k^*}\left(n_{k^*}, \frac{\delta}{2K}\right) < \mu^{\text{ref}} < L_{k^*}\left(n_{k^*}, \frac{\delta}{2K}\right)$, arm ℓ cannot be in the set of active arms at round t and hence cannot be sampled at round t. Therefore, all the four cases listed above cannot occur and we have a contradiction.

This proves the statement of Lemma 25, as at least one of the events listed in Lemma 25 must occur for the algorithm to proceed further. This analysis follows similar steps as that in [31, 70] but needed further modifications to prove statement for our C-LUCB algorithm. □

Lemma 26. *Let $T^{(B)}$ denote the total number of times that the events (3) or (4) of Lemma 25 occur. We have that $T^{(B)}$ is upper bounded by $\sum_{k\in\mathcal{C}} \frac{\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$ under the event $\mathcal{E}$.*

Proof. We now bound $T^{(B)}$ under the event $\mathcal{E}$,

$$T^{(B)} = \sum_{t=1}^{\infty} \mathbb{1}\Big(\{m_1(t) = k^* \text{ is BAD}, m_2(t) \notin \mathcal{C}\} \cup \{m_2(t) = k^* \text{is BAD}, m_1(t) \notin \mathcal{C}\} \bigcup$$

$$\{m_1(t) \in \mathcal{C} \text{ is BAD or } m_2(t) \in \mathcal{C} \text{ is BAD}\}\Big) \tag{4.74}$$

$$\leq \sum_{t=1}^{\infty} \mathbb{1}\Big((\{m_1(t) \text{ is } k^* \text{ or } m_2(t) \text{ is } k^*\} \cap \{k^* \text{ is BAD}\}) \bigcup$$

$$(\{m_1(t) \in \mathcal{C} \text{ is BAD or } m_2(t) \in \mathcal{C} \text{ is BAD}\})\Big) \tag{4.75}$$

$$= \sum_{t=1}^{\infty} \sum_{k\in\mathcal{C}} \mathbb{1}\Big((\{m_1(t) \text{ is } k^* \text{ or } m_2(t) \text{ is } k^*\} \cap \{k^* \text{ is BAD}\}) \bigcup$$

$$= (\{m_1(t) \text{ is k or } m_2(t) \text{ is k}\} \cap \{k \text{ is BAD }\})\Big) \tag{4.76}$$

$$= \sum_{t=1}^{\infty} \sum_{k\in\mathcal{C}} \mathbb{1}\left(\{m_1(t) \text{ is k or } m_2(t) \text{ is k}\} \cap \{k \text{ is BAD }\}\right) \tag{4.77}$$

$$= \sum_{t=1}^{\infty} \sum_{k\in\mathcal{C}} \mathbb{1}\left(\{m_1(t) \text{ is k or } m_2(t) \text{ is k}\} \cap \{n_k(t) \leq \tau_k\}\right) \tag{4.78}$$

$$\leq \sum_{k\in\mathcal{C}} \tau_k \tag{4.79}$$

The last (4.79) holds from the fact that if $n_k(t) \leq \tau_k$ and $m_1(t)$ is k or $m_2(t)$ is k, then arm k gets sampled and $n_k(t+1) = n_k(t) + 1$, this can only occur τ_k times before $n_k(t) > \tau_k$. For anytime confidence intervals $B\left(n_k, \frac{\delta}{2K}\right)$, first integer τ_k such that $B\left(n_k, \frac{\delta}{2K}\right) < \frac{\Delta_k}{4}$ is upper bounded by $\frac{\zeta}{\Delta_k^2} \log\left(\frac{2K\log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right)$ where

$\zeta > 0$ is a constant depending on the tightness of confidence interval $B(n_k, \delta)$ [69]. The tighter the confidence interval, smaller is the constant ζ. Due to this, we get a bound on $T^{(B)}$ under the event $\mathcal{E}$ as,

$$T^{(B)} \leq \sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log \left(\frac{2K \log \left(\frac{1}{\Delta_k^2} \right)}{\delta} \right).$$

As the probability of event $\mathcal{E}$ is at least $1 - \delta$, we get that $T^{(B)} \leq \sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log \left(\frac{2K \log \left(\frac{1}{\Delta_k^2} \right)}{\delta} \right)$ with probability $1 - \delta$. In Section 4.5, we denoted $T^{(C)}$ as the total number of rounds in which $m_1(t), m_2(t) \in \mathcal{C}$ and $I_{k^*}(t) > \mu_{k^*}$ and similarly $T^{(*)}$ as the total number of rounds in which $m_1(t) = k^*, m_2(t) \notin \mathcal{C}$ or $m_2(t) = k^*, m_1(t) \notin \mathcal{C}$. From Lemma 25, we note that $T^{(*)} + T^{(C)}$ is equivalent to $T^{(B)}$ on which we derived a bound above. Due to this, $T^{(C)} + T^{(*)} = T^{(B)} \leq \sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log \left(\frac{2K \log \left(\frac{1}{\Delta_k^2} \right)}{\delta} \right)$ under the event $\mathcal{E}$.

□

4.9.5 Proof of Theorem 9

We now bound the total number of rounds played by C-LUCB algorithm under the event $\mathcal{E}$. From Lemma 25, we note that if the algorithm has not stopped at round t under the event $\mathcal{E}$, it implies that at least one of the following events must be true at round t,

1. Event $\mathcal{R}(t)$ does not occur, i.e., $I_{k^*}(t) < \mu_k^*$
2. $m_1(t)$ or $m_2(t)$ is Non-Competitive and $m_1(t), m_2(t) \neq k^*$,
3. ($m_1(t) = k^*$ is BAD and $m_2(t) \notin \mathcal{C}$) or ($m_2(t) = k^*$ is BAD and $m_1(t) \notin \mathcal{C}$)
4. $m_1(t), m_2(t) \in \mathcal{C}$ and either $m_1(t)$ is BAD or $m_2(t)$ is BAD.

From Lemma 20 we see that $\Pr(\mathcal{R}(t)) \leq \frac{K}{t^3}$ and the result from Lemma 24 gives us a bound on $\Pr((m_1(t) \notin \mathcal{C} \cup m_2(t) \notin \mathcal{C}), (m_1(t), m_2(t) \neq k^*), \mathcal{E})$. The result from Lemma 26 shows that the third and fourth event occur at most $\sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log \left(\frac{2K \log \left(\frac{1}{\Delta_k^2} \right)}{\delta} \right)$ times. Combining these, we get our desired bound on the sample complexity result.

Proof.

$$\mathbb{E}\left[\mathcal{T} | \mathcal{E}\right] = \sum_{t=1}^{\infty} \mathbb{E}\left[\mathbb{1}(t = \mathcal{T} | E)\right] \tag{4.80}$$

$$\leq \sum_{t=1}^{\infty} \mathbb{1}(\mathcal{R}(t)|\mathcal{E}) + \mathbb{E}\left[T^{(B)}|\mathcal{E})\right] +$$

$$\sum_{t=1}^{\infty} \mathbb{E}\left[\mathbb{1}(m_1(t) \text{ or } m_2(t) \notin \mathcal{C}, m_1(t) \text{ and } m_2(t) \neq k^*|\mathcal{E})\right] \tag{4.81}$$

$$= \sum_{t=1}^{\infty} \Pr(\mathcal{R}(t), \mathcal{E}) \times \frac{1}{\Pr(\mathcal{E})} + \mathbb{E}\left[T^{(B)}|\mathcal{E})\right] +$$

$$\sum_{t=1}^{\infty} \Pr\left((m_1(t) \text{ or } m_2(t) \notin \mathcal{C}), m_1(t) \text{ and } m_2(t) \neq k^*, \mathcal{E}\right) \times \frac{1}{\Pr(\mathcal{E})} \tag{4.82}$$

$$\leq \sum_{t=1}^{\infty} \frac{1}{1-\delta} \times \frac{K}{t^3} + \sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log\left(\frac{2K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) +$$

$$\frac{Kt_0}{1-\delta} + \frac{1}{1-\delta} \sum_{t=Kt_0+1}^{\infty} \left(\frac{2(K+1)K}{t^3} + \frac{K(K+1)^3}{t^2}\right) \tag{4.83}$$

$$\leq \frac{3K}{2(1-\delta)} + \frac{Kt_0}{1-\delta} + \frac{1}{1-\delta} \times \left(\frac{2}{t_0^2} + \frac{(K+1)^3}{t_0}\right) + \sum_{k \in \mathcal{C}} \frac{\zeta}{\Delta_k^2} \log\left(\frac{2K \log\left(\frac{1}{\Delta_k^2}\right)}{\delta}\right) \tag{4.84}$$

□

By noting that the C-LUCB samples two arms at each round, we get the sample complexity result stated in Theorem 9.

4.9.6 Proof for Theorem 8

Proof. To prove Theorem 8, we define three events $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$ below. Let $\mathcal{E}_1$ be the event that empirical mean of all arm lie within their confidence intervals uniformly for all $t \geq 1$

$$\mathcal{E}_1 = \left\{\forall t \geq 1, \forall k \in \mathcal{K}, \quad \hat{\mu}_k(t) - B\left(n_k(t), \frac{\delta}{2K}\right) \leq \mu_k \leq \hat{\mu}_k + B\left(n_k(t), \frac{\delta}{2K}\right)\right\} \tag{4.85}$$

Define $\mathcal{E}_2$ to be the event that empirical pseudo-reward of optimal arm with respect to all other arms lie within their crossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_2 = \left\{\forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{k^*,\ell} \leq \hat{\phi}_{k^*,\ell}(t) + B\left(n_\ell(t), \frac{\delta}{2K}\right)\right\} \tag{4.86}$$

Similarly define $\mathcal{E}_3$ to be the event that the empirical pseudo-reward of the sub-optimal arms with respect to the optimal arm lies within their crossUCB indices uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_3 = \left\{\forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{\ell,k^*} \leq \hat{\phi}_{\ell,k^*}(t) + B\left(n_{k^*}(t), \frac{\delta}{2K}\right)\right\} \tag{4.87}$$

Furthermore, we define $\mathcal{E}$ to be the intersection of the three events, i.e.,

$$\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3. \tag{4.88}$$

Due to the nature of anytime confidence intervals (See (4.2)) and union bound over the set of arms, we have $\Pr(\mathcal{E}_1^c) \leq \frac{\delta}{2}$, $\Pr(\mathcal{E}_2^c) \leq \frac{\delta}{4}$ and $\Pr(\mathcal{E}_3^c) \leq \frac{\delta}{4}$ giving us $\Pr(\mathcal{E}^c) \leq \delta$. We now show that under the event $\mathcal{E}$, the C-LUCB algorithm cannot stop with an arm $k \neq k^*$. We do that through a proof by contradiction.

Suppose, the algorithm stops with arm $k \neq k^*$, i.e., arm k is the only arm in set $\mathcal{A}_t$. In such a scenario, $\exists \tau, k \neq k^* : \tilde{U}_{k^*}\left(\tau, \frac{\delta}{2K}\right) < L_k\left(\tau, \frac{\delta}{2K}\right)$. This can only occur if one of the following events occur,

1. $\tilde{U}_{k^*,k^*}\left(\tau, \frac{\delta}{2K}\right) < L_k\left(\tau, \frac{\delta}{2K}\right))$
2. $\tilde{U}_{k^*,\ell}\left(\tau, \frac{\delta}{2K}\right) < L_k\left(\tau, \frac{\delta}{2K}\right) \quad \ell \neq k^*$

See that under the event $\mathcal{E}$, $\tilde{U}_{k^*,\ell}\left(\tau, \frac{\delta}{2K}\right) > \mu_{k^*}$ and $L_k\left(\tau, \frac{\delta}{2K}\right) < \mu_k \quad \forall \tau, k$. This shows that under the event $\mathcal{E}$, $\tilde{U}_{k^*}\left(\tau, \frac{\delta}{2K}\right) > L_k\left(\tau, \frac{\delta}{2K}\right) \quad \forall k, \tau$ as $\mu_{k^*} > \mu_k \quad \forall k \neq k^*$. This implies that the algorithm returns the best arm with probability at least $1-\delta$ as $\Pr(\mathcal{E}^c) \leq \delta$. □

4.9.7 $1-\delta$ Correctness of C-LUCB++

We now show that the C-LUCB++ algorithm declares the arm k^* as the best arm with probability at least $1-\delta$.

Proof. To prove the correctness of C-LUCB++, we use similar arguments as done in the proof of Theorem 8 for the C-LUCB algorithm. In particular, we define an event $\mathcal{E}^+$ that holds true with at least $1-\delta$ probability and show that the C-LUCB++ algorithm always stops with the best arm under the event $\mathcal{E}^+$.

We define three events $\mathcal{E}_1^+, \mathcal{E}_2^+$ and $\mathcal{E}_3^+$ below. Let $\mathcal{E}_1^+$ be the event that empirical mean of all arm $k \neq k^*$ lie within their confidence intervals uniformly for all $t \geq 1$

$$\mathcal{E}_1^+ = \left\{\forall t \geq 1, \forall k \in \mathcal{K}, \quad \hat{\mu}_k(t) - B\left(n_k(t), \frac{\delta}{3K}\right) \leq \mu_k \leq \hat{\mu}_k + B\left(n_k(t), \frac{\delta}{3K}\right)\right\} \tag{4.89}$$

Define $\mathcal{E}_2^+$ to be the event that empirical pseudo-reward of optimal arm with respect to all other arms lie within their confidence intervals uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_2^+ = \left\{\forall t \geq 1, \forall \ell \in \mathcal{K}, \quad \phi_{k^*,\ell} \leq \hat{\phi}_{k^*,\ell}(t) + B\left(n_\ell(t), \frac{\delta}{3K}\right)\right\} \tag{4.90}$$

Additionally, define $\mathcal{E}_3^+$ as the event where empirical mean of arm k^* lies below the upper confidence index of arm k^* (constructed with width $\delta/4$) uniformly for all $t \geq 1$, i.e.,

$$\mathcal{E}_3^+ = \left\{\forall t \geq 1, \quad \mu_{k^*} \leq \hat{\mu}_{k^*}(t) + B\left(n_{k^*}(t), \frac{\delta}{4}\right)\right\} \tag{4.91}$$

Furthermore, we define $\mathcal{E}^+$ to be the intersection of the three events, i.e.,

$$\mathcal{E}^+ = \mathcal{E}_1^+ \cap \mathcal{E}_2^+ \cap \mathcal{E}_3^+ \tag{4.92}$$

Due to the nature of anytime confidence intervals (See (4.2)) and union bound over the set of arms, we have $\Pr(\mathcal{E}_1^+) \geq 1 - \frac{\delta}{3}$, $\Pr(\mathcal{E}_2^+) \geq 1 - \frac{\delta}{6}$ and $\Pr(\mathcal{E}_3^+) \geq 1 - \frac{\delta}{2}$, giving us $\Pr(\mathcal{E}^+) \geq 1 - \delta$. We now show that under the event $\mathcal{E}^+$, the C-LUCB++ algorithm cannot stop with an arm $k \neq k^*$. We do that through a proof by contradiction.

Suppose, the algorithm stops with arm $k \neq k^*$, i.e., arm k is the only arm in set $\mathcal{A}_t$ or $\max_{\ell \neq k} \tilde{U}_{\ell,\ell}\left(\tau, \frac{\delta}{4}\right) < L_k\left(\frac{\delta}{4K}\right)$. In such a scenario, $\exists \tau, k \neq k^* : \tilde{U}_{k^*}\left(\tau, \frac{\delta}{3K}\right) < L_k\left(\tau, \frac{\delta}{3K}\right)$ or $\tilde{U}_{k^*,k^*}\left(\tau, \frac{\delta}{4}\right) < L_k\left(\frac{\delta}{4K}\right)$ This can only occur if one of the following events occur,

1. $\tilde{U}_{k^*,k^*}\left(\tau, \frac{\delta}{4}\right) < L_k\left(\tau, \frac{\delta}{4K}\right) < L_k\left(\tau, \frac{\delta}{4K}\right)$
2. $\tilde{U}_{k^*,\ell}\left(\tau, \frac{\delta}{3K}\right) < L_k\left(\tau, \frac{\delta}{3K}\right) \quad \ell \neq k^*$

See that under the event $\mathcal{E}^+$, $\tilde{U}_{k^*,\ell}\left(\tau, \frac{\delta}{3K}\right) > \mu_{k^*}$, $\tilde{U}_{k^*,k^*}\left(\tau, \frac{\delta}{4}\right) < \mu_{k^*}$ and $L_k\left(\tau, \frac{\delta}{3K}\right) < \mu_k \quad \forall \tau, k$. This shows that under the event $\mathcal{E}^+$, $\tilde{U}_{k^*,k^*}\left(\tau, \frac{\delta}{4}\right) > L_k\left(\tau, \frac{\delta}{3}\right) \quad \forall k, \tau$ and $\tilde{U}_{k^*,\ell}\left(\tau, \frac{\delta}{3K}\right) > L_k\left(\tau, \frac{\delta}{3K}\right) \quad \forall \ell \neq k^*$ as $\mu_{k^*} > \mu_k \quad \forall k \neq k^*$. This implies that the algorithm returns the best arm with probability at least $1 - \delta$ as $\Pr(\mathcal{E}^+) \geq 1 - \delta$. $\square$

Chapter 5

Correlated Combinatorial Bandits for Online Resource Allocation

In the last couple of chapters, we have established our proposed correlated multi-armed bandit framework and demonstrated its application in the context of recommendation systems. In this chapter, we further demonstrate the utility of our propose correlated bandit framework by solving online resource allocation problems, which frequently arise in tasks such as power allocation in wireless systems, financial optimization and multi-server scheduling. This is done by extending our framework and algorithms to the setting of online resource allocation. We begin our discussion by first understanding the online resource allocation problem. We then establish connections of this problem with the correlated multi-armed bandit framework proposed in this thesis.

5.1 Introduction

5.1.1 Background and Motivation

Resource allocation is a fundamental challenge that arises in wide ranging applications, including wireless networks [80, 81], computer systems [82], multi-server scheduling [83] and financial optimization [84]. In the case of financial optimization, the company needs to decide the investment of its limited financial budget across different products with the goal of maximizing its overall revenue. In the context of power allocation in multi-channel wireless systems, the goal is to maximize the throughput of the system by allocating the power across different available channels. In such problems, the task is to distribute a limited *budget* (i.e., money, power, etc.) among available *entities* (i.e., product teams, channel etc.) with the objective of maximizing the *reward* attained (i.e., revenue, throughput, etc.). These budget allocation problem can be framed as the following optimization problem,

$$\begin{aligned} \underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad & \sum_{k=1}^{K} f_k(a_k) \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \leq Q, a_k \in \mathcal{A}, \end{aligned} \tag{5.1}$$

with S being the budget allocation vector $(a_1, a_2, \ldots a_K)$ and Q representing the total available budget. The function $f_k(a_k)$ represents the reward attained from entity k upon allocating a budget of a_k to entity k. This budget is selected from a set $\mathcal{A}$ which may or may not be countable. Depending on the problem setting, the reward functions f_k may or may not be known. For instance, under the financial optimization example, the company distributes its total budget of Q among K different products with the goal of maximizing the total revenue, which is the sum of revenue $f_k(a_k)$ from individual products. In this example, the reward function $f_k(a_k)$ may not be known. In the power allocation problem for wireless systems, a total power of Q needs to be distributed across K different channels, and the throughput at each channel depends on the power allocated to that channel and is typically known as a function of the power allocated to the channels.

Moreover, in these problems, the reward obtained upon allocating a budget of a_k to entity k may be random and may depend on the underlying randomness associated with entity k. For instance, the revenue of the product may depend on the underlying unknown demand/market factors. Similarly, in the power allocation problem, with allocated power a_k, the throughput at channel k is $\log\left(1 + \frac{a_k}{X_k}\right)$, where X_k is the background noise associated with channel k and is random. As a result, the problem of budget allocation would now be

$$\begin{aligned} \underset{S=(a_1,a_2,\ldots a_K)}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\ \text{subject to} \quad & \sum_{k=1}^{K} a_k \leq Q, a_k \in \mathcal{A}. \end{aligned} \tag{5.2}$$

In this scenario, the optimization problem can be solved if $\mathbb{E}\left[f_k(a_k, X_k)\right]$ is known for all (a_k, k) pairs, i.e., the mean reward of each entity k is known at all budget allocations a_k for entity k. In view of this, we refer to (5.2) as the *offline budget allocation problem*. In practice, the reward function may be unknown and the X_k's may be unknown parameters in the reward function. For instance, in the financial optimization, the reward obtained for a given budget allocation a_k for product k may depend on underlying market conditions X_k and one may not know the corresponding reward function f_k. As a result, $\mathbb{E}\left[f_k(a_k, X_k)\right]$ remains unknown. In the power allocation example, X_k corresponds to the background noise, which is a latent variable whose distribution is unknown, and correspondingly one does not know $\mathbb{E}\left[f_k(a_k, X_k)\right]$ a priori.

Motivated by this, we study the online resource allocation problem, where the goal is to sequentially decide a budget allocation $\boldsymbol{S_t} = (a_{1,t}, a_{2,t} \ldots a_{k,t}, \ldots a_{k,t})$ for each round t, so as to maximize the cumulative reward attained over a total of T rounds. To perform this allocation, there is a need to estimate $\mathbb{E}\left[f_k(a_k, X_k)\right]$ for each (a_k, k) pair and subsequently use these estimates to decide a budget allocation $\boldsymbol{S_t}$ that generates the maximum possible reward in round t. When deciding a budget allocation $\boldsymbol{S}_t$, the decision-maker has two conflicting goals. Firstly, the allocation $\boldsymbol{S}_t$ should try to gather as much information as possible about the unknown reward distributions (exploration), and secondly the allocation should try to maximize the reward in each round (exploitation).

Resource allocation as a combinatorial bandit problem. In order to balance this exploration-exploitation trade-off, we can view the online resource allocation problem as a combinatorial bandit problem, which is a variant of the classical multi-armed bandit (MAB) problem [1, 30]. Under the classical multi-armed bandit framework, the decision-maker is faced with M different base arms whose distributions are unknown and the goal is to maximize the long-term cumulative reward over a total of T rounds by selecting one amongst the available M base arms in each round t and observing its reward. Under the combinatorial bandit framework [85], the decision-maker can select multiple base arms in a given round from a given pre-defined set and observe the reward for each of the selected base arms. By viewing the allocation of budget a_k to entity k as a base arm (a_k, k), we can view the online resource allocation problem as a combinatorial bandit problem [85]. The underlying distribution of the reward of each base arm (a_k, k), i.e., $f_k(a_k, k)$, is unknown and the goal is to maximize the cumulative reward over a total of T rounds by selecting K different base arms in each round t, i.e., one corresponding to each entity k. Upon the budget allocation, we receive rewards for all the base arms selected in round t, which is then used to decide the budget allocation in round $t+1$. By modeling the resource allocation problem as a combinatorial multi-armed bandit problem, we can use the existing combinatorial bandit algorithms to solve the resource allocation. However, these algorithms do not exploit the structural correlations in reward functions $f_k(a_k, X_k)$. Taking advantage of these correlations is the main challenge of our work.

5.1.2 Main contributions

Novel correlated combinatorial bandit framework for online resource allocation. The combinatorial bandit framework described above considers the reward obtained for different base arms to be independent of each other. However, in the context of resource allocation, the rewards may be correlated in two ways. i) the rewards received for one entity k at budget i and for the same entity k at budget j are likely to be correlated. For instance, in the power allocation example, the throughput observed at channel k under power i gives some information on what the throughput would have been if power j were allocated to channel k. ii) the rewards received across two different entities may also be correlated. In the financial optimization example, the revenue obtained from product k under budget i may give some information on what the revenue would have been at product ℓ under budget j. This may occur if the sales of two products are related to one another. In this work, we model such correlations through *pseudo-rewards*, which are upper bounds on conditional expected reward of each base-arm (j, ℓ) given reward sample of base-arm (i, k). In the financial optimization example, this amounts to the knowledge of the form "what is the maximum revenue the company can expect from product ℓ at budget j given the observed revenue of product k under budget i?". The details of this framework are presented in Section 5.2.

Correlated and Combinatorial UCB. For this novel framework, we propose the correlated upper confidence bound algorithm for online resource allocation. It makes use of the correlations across base-arms to select an allocation $\boldsymbol{S_t}$ that balances the task of gaining information about the reward distributions of $f_k(a_k, X_k)$ for each (a_k, k) pair and maximizing the expected reward in round t based on the available information. More specifically, it computes an upper confidence bound on $\mathbb{E}\left[f_k(a_k, X_k)\right]$ for each (a_k, k) pair through the reward samples observed of $f_k(a_k, X_k)$ observed till round t. These reward samples may be obtained *directly* from the past reward samples of base arm (a_k, k) or *indirectly* through the pseudo-rewards of base arm (a_k, k) from the past reward samples of other base arms (j, ℓ). These upper confidence bounds on $\mathbb{E}\left[f_k(a_k, X_k)\right]$ are then used to select an allocation $\boldsymbol{S}_t$ to be played in round $t + 1$. The proposed algorithm is detailed in Section 5.3. As our proposed approach makes use of the correlation information in selection of $\boldsymbol{S}_t$, as opposed to prior work which were correlation-agnostic, we observe significant performance gains.

Reduction in cumulative regret through correlations. We evaluate our proposed algorithm in terms of the *cumulative regret*, which is defined as the difference between the total reward obtained by our online algorithm and the total reward obtained by the optimal offline solution, where the offline problem has complete knowledge about the joint distribution of X. We introduce novel proof techniques to analyze the regret, and show that the regret of our proposed algorithm is $C \cdot \mathrm{O}(\log T)$ where $0 \leq C \leq KA$, with A denoting the size of the set $\mathcal{A}$ from which budget a_k is allocated to each entity. This is a significant improvement over approaches that are agnostic to correlation [21], which have a regret of the form of $KA \cdot \mathrm{O}(\log T)$. In a lot of practical settings, $C = 0$, which implies that our proposed algorithm achieves a *bounded regret*. This is an order-wise improvement over correlation-agnostic approaches as shown in Section 5.4.

Synthetic experiments on real-world problems. We validate the performance of our algorithm by evaluating it on three practical problems in Section 5.6. We conduct experiments for i) the power allocation problem in wireless systems, ii) channel assignment in slotted ALOHA protocol and iii) scheduling of jobs in a multi-server system. For all the three problems, we see that using our correlated and combinatorial UCB algorithm achieves significant improvement in performance relative to correlation agnostic approaches.

5.1.3 Related works

The classical offline resource allocation problem, i.e., the setting where the distributions of $f_k(a_k, X_k)$ are known, has been extensively studied for decades [80, 86, 87] and has been applied in several application settings such as financial optimization [84], wireless systems [81, 80], scheduling in multi-server systems [88] etc. Recently, the online resource allocation problem has attracted much attention as the distribution of rewards $f_k(a_k, X_k)$ is typically unknown in practice [89, 90, 91, 21]. First, the online resource allocation problem was studied in a setting where the reward functions $f_k(a_k, X_k)$ were assumed to be linear [89, 92]. This was extended by [91], as they assume the reward functions to be concave. More recently, [21] studied this problem in the most general setting by placing no restriction on the type of reward functions $f_k(a_k, X_k)$.

In [21], the online resource allocation is modeled as a combinatorial multi-armed bandit problem by viewing the allocation of budget a_k to entity k as a base arm. Subsequently, they extend the UCB algorithm for combinatorial bandits [93] to the online resource allocation problem. The action space $\mathcal{A}$ in [21] is allowed to be countable, unlike [93] which restricted the action space to be binary. A drawback of the approach in [21] is that it considers the rewards corresponding to different base arms to be independent of each other, and does not make use of the fact that the reward obtained from one base arm may give some information on what the reward would have been for a different base arm.

In this work, we fill this gap by proposing our correlated combinatorial bandit framework to study the online resource allocation in the most general setting. To the best of our knowledge, this is the first work that models the correlation in a combinatorial bandit framework. The idea of capturing correlations in reward across different arms was previously studied in the context of classical multi-armed bandits, i.e., the setting where only one base-arm is played in each round t, in [54, 94]. We extend this idea to the combinatorial bandit framework, where multiple base-arms may be played in each round t, and propose the correlated UCB algorithm for online resource allocation. The extension is non-trivial as the classical multi-armed bandit and combinatorial bandit often require different design of algorithms and regret analysis due to selection of multiple base arms within provided constraints as opposed to the selection of single base arm in each round t. Upon doing so, we are able to exploit the correlations to obtain significant performance improvements as demonstrated in Section 5.4, 5.6. To the best of our knowledge, this is the first work to show that $O(1)$ regret can be achieved in certain online resource allocation problems.

5.2 Problem Setup

5.2.1 Offline Resource Allocation

Consider the offline resource allocation problem where a decision-maker splits the available budget among K different entities. For each entity $k \in [K]$, the decision-maker needs to decide a budget $a_k \in \mathcal{A}$, where $\mathcal{A}$ is the feasible budget space. Notice that the budget space $\mathcal{A}$ could be either discrete (e.g., $\mathbb{N}$) or continuous (e.g., $\mathbb{R}_{\geq 0}$). We focus on the discrete action space first and then consider the case of continuous action space separately in Section 5. We denote the overall budget allocation vector as $\boldsymbol{S} = (a_1, \cdots, a_K)$. We consider a general reward function $f_k(a_k, X_k)$ for each entity k, where X_k (which can be discrete or continuous) is a hidden random variable which reflects the random fluctuation of the obtained reward within entity k. We also consider m general constraints, denoted as $h_i(\boldsymbol{S}) \leq 0, i = 1, 2, \cdots, m$.

For the offline setting where the joint distribution $\boldsymbol{D} = (D_1, \ldots, D_K)$ of $\boldsymbol{X} = (X_1, X_2 \ldots X_K)$ is known, our goal is to maximize the expected total reward collected from all entities, which we denote by $r(\boldsymbol{S}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right]$. This can be formulated as the following optimization problem.

$$
\begin{aligned}
\underset{\boldsymbol{S}=(a_1,\cdots,a_K)}{\text{maximize}} \quad & \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_k, X_k)\right] \\
\text{subject to} \quad & h_i(\boldsymbol{S}) \leq 0, \quad i = 1, 2, \cdots, m \\
& a_k \in \mathcal{A}, \qquad \forall k \in [K]
\end{aligned}
\tag{5.3}
$$

The above formulation is a general version of (1) and (2) which contain just one constraint $h_1(\boldsymbol{S}) = \sum_k a_k - Q$. We could have more complex constraints on $\boldsymbol{S}$ through $h_i(\boldsymbol{S})$, e.g., $max_k a_k - W \leq 0$. These constraints on budgets are known to the decision-maker. For instance, if $f_k(a_k, X_k)$ is convex over a_k, $h_i(\boldsymbol{S})$ is convex over $\boldsymbol{S}$, and $\mathcal{A}$ is a convex set, it becomes a convex optimization problem that might be solved exactly; if $\mathcal{A}$ is a discrete set, it can be a NP-hard combinatorial optimization problem.

As the reward functions $f_k(\dot{})$ may not be known in practice (e.g., the financial optimization example in Section 5.1), we do not specify the exact form of the reward functions $f_k(a_k, X_k)$ and consider them to be unknown. We assume that there exists an offline approximation oracle $\mathcal{A}$, which outputs an allocation $\mathcal{S}^{\mathcal{O}}$ such that $r(\boldsymbol{S}^{\mathcal{O}}, \boldsymbol{D}) \geq \alpha \cdot \text{opt}(\boldsymbol{D})$, where α is the approximation ratio and $\text{opt}(\boldsymbol{D}) = \sup_{\boldsymbol{S}} r(\boldsymbol{S}, \boldsymbol{D})$ is the optimal solution to the budget allocation problem. The oracle can output such an allocation if $\mathbb{E}\left[f_k(a_k, X_k)\right]$ is known for all $(a_k, k) \in \mathcal{A} \times \mathcal{K}$.

5.2.2 Online Resource Allocation as a Combinatorial Bandit Problem

Now we introduce the online version of the resource allocation, which is a sequential decision making problem. In each round t, we allocate $a_{k,t}$ budget to each entity k, subject to the budget constraints, $h_i(\boldsymbol{S}) \leq 0, i = 1, 2, \cdots, m$. We then obtain $f_k(a_{k,t}, X_{k,t})$ reward from each entity k, where $X_{k,t}$ is sampled from an unknown distribution D_k. The total reward obtained in round t is $\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})$. Our goal is to accumulate as much total reward as possible through this sequential budget allocation.

We denote the overall budget allocation in round t as $\boldsymbol{S_t} = (a_{1,t}, \cdots, a_{K,t})$ and the joint distribution of all $X_{k,t}$'s as $\boldsymbol{D} = (D_1, \ldots, D_K)$. We define the expected total reward obtained in round t as $r(\boldsymbol{S_t}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right]$. We consider a learning algorithm π that makes the budget allocation $\boldsymbol{S_t^{\pi}}$ in round t. We can measure the performance of π by its (expected) regret, which is the difference in expected cumulative reward between always taking the best offline allocation and taking the budget allocation selected by algorithm π. The best offline allocation can be obtained through the offline oracle $\mathcal{O}$, which knows the underlying joint distribution $\mathcal{D}$, and attains $r(\boldsymbol{S}_t^{\mathcal{O}}, \boldsymbol{D}) \geq \alpha \cdot \mathrm{opt}(\boldsymbol{D})$. In view of that, we use the following approximation regret for T rounds:

$$Reg_{\alpha}^{\pi}(T; \boldsymbol{D}) = T \cdot \alpha \cdot \mathrm{opt}(\boldsymbol{D}) - \textstyle\sum_{t=1}^{T} r(\boldsymbol{S}_t^{\pi}, \boldsymbol{D}). \tag{5.4}$$

Since the obtained reward $f_k(a_k, X_k)$ of entity k is determined by the allocated budget a_k, following the combinatorial multi-armed bandit framework [85], we can view allocating budget i to entity k as a base arm and denote it as (i, k). The overall budget allocation $\boldsymbol{S_t}$ can be considered as a super arm that consists of multiple base arms. For each base arm (i, k), we denote the expected reward of playing it as $\mu_{i,k} = \mathbb{E}_{X_{k,t} \sim D_k}\left[f_k(i, X_{k,t})\right]$. We can rewrite the expected total reward obtained in round t:

$$r(\boldsymbol{S_t}, \boldsymbol{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K} \sum_{i \in \mathcal{A}} \mu_{i,k} \cdot \mathbb{1}\{a_{k,t} = i\}, \tag{5.5}$$

Note that the expected total reward depends only on the mean rewards of base arms (i, k), therefore we can re-write the expected total reward as

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{i \in \mathcal{A}} \mu_{i,k} \cdot \mathbb{1}\{a_{k,t} = i\}. \tag{5.6}$$

If the mean rewards $\mu_{i,k}$ of individual base arms (i, k) were known, then one can use the offline oracle to obtain the optimal budget allocation in each round. As the mean rewards of individual base arms are unknown, they need to be estimated from the historical observations until round t. The mean reward of the base arm (i, k) can be estimated either through the past samples in which budget i was allocated to entity k, or through the side information collected from other observations. We discuss the latter next.

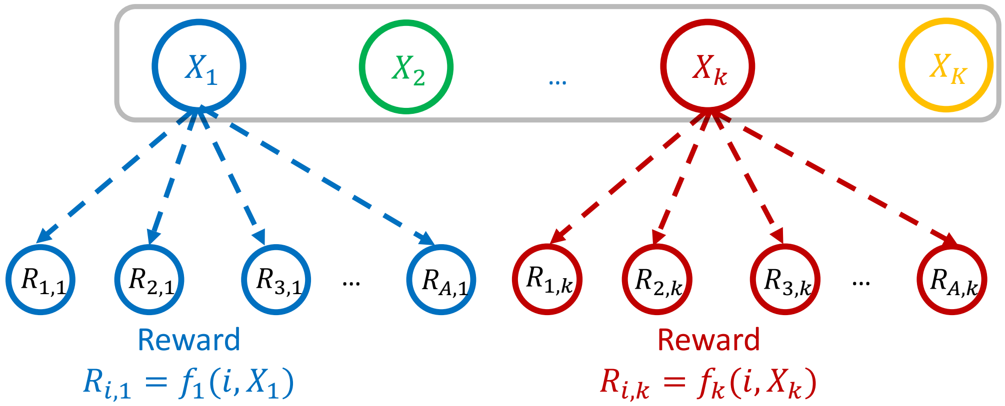


Figure 5.1: The rewards corresponding to a base arm (i, k), i.e., budget i to entity k, is a function of the allocated budget i and underlying randomness X_k associated with entity k. The rewards for base arms (i, k) and (j, k), i.e., different budget allocations within entity k, are correlated through X_k. There may be also correlation in the rewards across different entities if $X_1, X_2, \ldots X_K$ are correlated.

5.2.3 Proposed Correlated Combinatorial Bandit Framework

In several application settings, there may be some information on the knowledge of reward functions $f_k(a_k, X_k)$. As a result, the knowledge of the reward from one base arm (i, k) may provide some information on the reward that would have been obtained from entity k if budget j was allocated to entity k. This is illustrated in Figure 5.1. For instance, in the power allocation example, where the objective is to allocate the total power Q among K different channels to maximize the total throughput, the throughput at channel k is given by $\log\left(1 + \frac{a_{k,t}}{X_{k,t}}\right)$. Here, $a_{k,t}$ represents the power allocated in channel k and X_k denotes the hidden noise in channel k at round t. As the expression of throughput, i.e., the reward function $f(a_k, X_k)$, is known, the throughput in channel k at power i provides some information on what the reward would have been if power j was allocated to channel k. More generally, rewards obtained from one base arm (i, k) may provide some information on the reward of another base arm (j, ℓ). As a result, the rewards corresponding to different base arms are correlated. We capture the presence of such correlations in the form of *pseudo-rewards*, as defined below:

Definition 16 (Pseudo-Reward). *Suppose that we sample the base arm (i, k) and observe reward r. We call a quantity $s_{(j,\ell),(i,k)}(r)$ as the pseudo-reward of base arm (j, ℓ) with respect to base arm (i, k) if it is an upper bound on the conditional expected reward of base arm (j, ℓ), i.e.,*

$$\mathbb{E}[f_\ell(j, X_\ell) \mid f_k(i, X_k) = r\,] \leq s_{(j,\ell),(i,k)}(r). \tag{5.7}$$

For convenience, we set $s_{(j,\ell),(j,\ell)}(r) = r \quad \forall j, \ell$.

When no information is known, pseudo-rewards between two base arms are not known, then they can be set equal to the maximum possible reward. This makes our formulation quite general and in fact subsumes the correlation agnostic combinatorial framework studied in [21], the connection will be made explicit through Remark 12 in Section 5.3. Next, we show how the pseudo-rewards can be evaluated in practice.

Obtaining pseudo-rewards from reward correlations within the same entity. These pseudo-rewards can be evaluated easily in several different practical settings. For instance, if the form of the functions $f_k(a_k, X_k)$ is known, then the pseudo-reward of base arm (j,k) with respect to base arm (i,k) can be obtained as

$$s_{(j,k),(i,k)}(r) = \max_{X_k} f_k(j, X_k) \quad \text{s.t. } f_k(i, X_k) = r. \tag{5.8}$$

Note that pseudo-rewards can be obtained even in the scenario where only probabilistic upper and lower bounds on $f_k(a_k, X_k)$ are known, i.e., $\underline{f}_k(a_k, X_k) \leq f_k(a_k, X_k) \leq \bar{f}_k(a_k, X_k)$ w.p. $1-\kappa$. In this scenario, we can construct pseudo-rewards as follows:

$$s_{(j,k),(i,k)}(r) = (1-\kappa)^2 \times \left(\max_{\{X_k : \underline{f}_k(i,X_k) \leq r \leq \bar{f}_k(i,X_k)\}} \bar{f}_k(j, X_k) \right) + (1-(1-\kappa)^2) \times M, \tag{5.9}$$

where M is the maximum possible reward that a base arm can provide. We evaluate this pseudo-reward by first identifying the range of values within which X_k lies based on the reward with probability $1-\kappa$. The maximum possible reward of the base arm (j,k) within the identified range of X_k is then computed with probability $1-\kappa$. Due to this, with probability $(1-\kappa)^2$, conditional reward of base arm (j,k) is at most $\max_{X_k : \underline{f}_k(i,X_k) \leq r \leq \bar{f}_k(i,X_k)} \bar{f}_k(j, X_k)$. As the maximum possible reward is M otherwise, we get (5.9).

Obtaining pseudo-rewards from reward correlation across entities. In the most general scenario, there may be knowledge of reward correlations across entities as shown in Figure 5.2. This can occur if the random variables X_k and X_ℓ, i.e., the hidden random variables corresponding to two different entities k and ℓ, are correlated. These correlations can be incorporated in our framework through pseudo-rewards $s_{(j,\ell),(j,k)}$, which are an upper bound on the conditional expected reward. For instance, in the application of financial optimization, the company may invest its total budget among different products. As the performance of different products are likely to be correlated, the reward feedback under budget i for product k may inform something about the reward feedback for product ℓ under budget j. Such correlations can be modeled through pseudo-rewards, which may either be known from domain knowledge or from previously performed controlled experiments. For example, based on previously performed experiments, it may be known that the expected reward obtained from product ℓ under budget j is at most y whenever the reward obtained for product k under budget i is x. Note that in this modeling, one does not need to explicitly capture what the inherent randomness X_k represents and its corresponding values. This is a key strength of our proposed framework, as in a several applications X_k could be hard to interpret and model. For instance, in the financial optimization example, X_k may represent underlying market conditions which

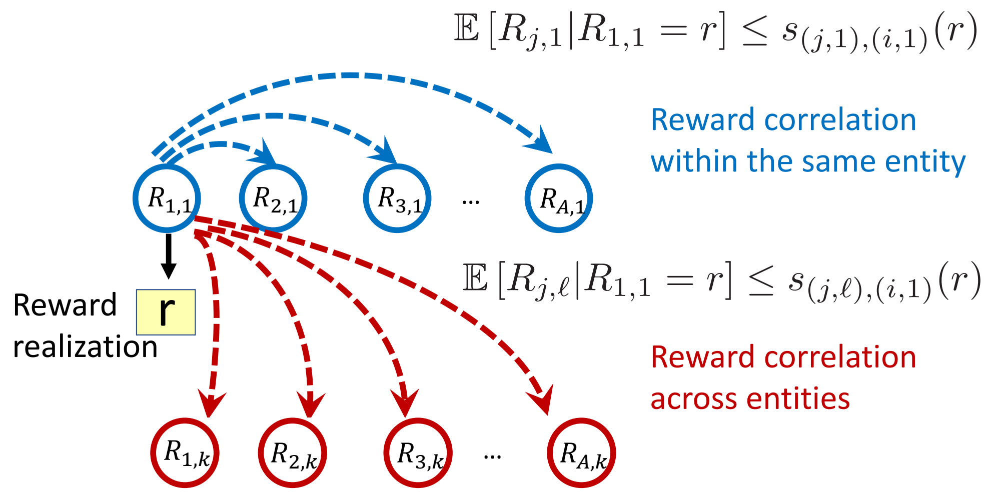


Figure 5.2: Upon observing a reward r from a base arm, pseudo-rewards $s_{(j,\ell),(i,k)}(r)$, give us an upper bound on the conditional expectation of the reward from base arm (j,ℓ) given that we observed reward r from arm (i,k). Reward received for entity k at a given budget i may provide some information on what the reward would have been if budget j were allocated to entity k, leading to correlations within entity. The rewards of different entities may also be correlated.

are complex and subsequently the reward functions $f_k(a_k, k)$ are also unknown. Even in such settings, the pseudo-reward based framework allows one to capture the correlation across different base arms.

5.3 Proposed Algorithm

We now propose the correlated-Upper Confidence Bound algorithm for resource allocation (corr-UCB-RA) that uses existing correlation in rewards across base arms to maximize the long-term cumulative reward. Before describing our algorithm, we first review the UCB algorithm for resource allocation (UCB-RA) proposed in [21].

5.3.1 The UCB algorithm for resource allocation

In order to solve the online resource allocation problem, the UCB-RA algorithm maintains a set of base arms $\{(k,a) \mid k \in [K], a \in \mathcal{A}\}$, where the total number of base arms is equal to KA, with A denoting the size of the discrete set $\mathcal{A}$. If the mean reward of each base arm were known, then the resource allocation problem can be easily solved by the use of the available offline oracle $\mathcal{O}$, which produces an allocation $S_t^{\mathcal{O}}$ such that $r(S_t^{\mathcal{O}}, \boldsymbol{\mu}) \geq \alpha \cdot \text{opt}(\boldsymbol{\mu})$. As the underlying mean rewards of the base arms are unknown, the UCB-RA algorithm maintains the empirical mean $\hat{\mu}_{i,k}(t)$ for each base arm (i,k) at round t. Using these empirical means, it then computes an upper confidence bound (UCB) index for each base arm (i,k) as

$$U_{i,k}(t) = \hat{\mu}_{(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}},$$

where $n_{(i,k)}(t)$ denotes the number of times budget i was allocated to entity k. UCB-RA algorithm then feeds these upper confidence indices of the base arms to the available offline oracle and obtains an allocation $\mathbf{S}_t = (a_{1,t}, a_{2,t}, \ldots a_{K,t})$. It then uses this allocation for round t and observes the feedback of $f_k(a_{k,t}, X_{k,t}) \forall k$. Note that the upper confidence indices are large if base arm (i,k) has a large empirical mean reward or if it has been sampled fewer times relative to other base arms. The algorithm description is presented in Algorithm 6.

5.3.2 The proposed correlated-UCB algorithm for resource allocation

Under the correlated combinatorial bandit framework, the pseudo-reward for base arm (j,ℓ) with respect to the base arm (i,k) provides an estimate on the reward of base arm (j,ℓ) based on the reward obtained from base arm (i,k). We now define the notion of empirical pseudo-reward, which can be used to obtain an *optimistic estimate* of $\mu_{(j,\ell)}$ through just reward samples of base arm (i,k).

Definition 17 (Empirical and Expected Pseudo-Reward)**.** *After t rounds, a base arm (i,k) is sampled $n_{(i,k)}(t)$ times. Using these $n_{(i,k)}(t)$ reward realizations, we can construct the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for each base arm (j,ℓ) with respect to base arm (i,k) as follows.*

$$\hat{\phi}_{(j,\ell),(i,k)}(t) \triangleq \frac{\sum_{\tau=1}^{t} \mathbb{1}_{(i,k)\in\mathcal{S}_t}\, s_{(j,\ell),(i,k)}(f_k(i, X_{k,t}))}{n_{(i,k)}(t)}, \tag{5.10}$$

$$(j,\ell) \in \mathcal{K} \times \mathcal{A} \setminus \{(i,k)\}. \tag{5.11}$$

The expected pseudo-reward of base arm (j,ℓ) with respect to base arm (i,k) is defined as

$$\phi_{(j,\ell),(i,k)} \triangleq \mathbb{E}\left[s_{(j,\ell),(i,k)}(f_k(i, X_k))\right]. \tag{5.12}$$

For convenience, we set $\hat{\phi}_{(i,k),(i,k)}(t) = \hat{\mu}_{(i,k)}(t)$ and $\phi_{(i,k),(i,k)} = \mu_{(i,k)}$. Note that the empirical pseudo-reward $\hat{\phi}_{(j,\ell),(i,k)}(t)$ is defined with respect to base arm (i,k) and it is only a function of the rewards observed by sampling base arm (i,k).

Definition 18 (PseudoUCB Index $U_{(j,\ell),(i,k)}(t)$)**.** *We define the PseudoUCB Index of base arm (j,ℓ) with respect to base arm (i,k) as follows.*

$$U_{(j,\ell),(i,k)}(t) \triangleq \hat{\phi}_{(j,\ell),(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}} \tag{5.13}$$

Furthermore, we define $U_{(j,\ell)}(t) = \min_{(i,k)} U_{(j,\ell),(i,k)}(t)$, the tightest of the KA upper bounds for base arm (j,ℓ).

At each round, the algorithm computes these pseudo-UCB indices $U_{(j,\ell)}$ for each base arm (j,ℓ). These indices are then fed to the oracle to obtain the budget allocation vector $\mathbf{S}_t$ at round t. At the end of each round we update the empirical pseudo-rewards $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for all (j,ℓ), the empirical reward for arm $(i,k) \in \mathcal{S}_t$, where $\mathcal{S}_t$ denotes the set of base arms played in round t. The description of this algorithm is given in Algorithm 7.

Remark 12 (Reduction to Combinatorial Multi-Armed Bandits)**.** *When all pseudo-reward entries are unknown, then all pseudo-reward entries can be filled with the maximum possible reward for each base arm, that is,* $s_{(i,k),(j,\ell)}(r) = M \; \forall r, \ell, k, i, j$. *In that case, the proposed Corr-UCB-RA algorithm reduces to the UCB-RA algorithm.*

Algorithm 6 UCB for resource allocation with offline oracle $\mathcal{O}$

1: **Input**: Constraints $g_i(\boldsymbol{S})$, Oracle $\mathcal{O}$.
2: For each base arm $(i,k) \in \mathcal{A} \times \mathcal{K}$, $n_{i,k}(t) \leftarrow 0$. {maintain the total number of times base arm (i,k) is played so far.}
3: **for** $t = 1, 2, 3, \ldots$ **do**
4: For each base arm $(i,k) \in \mathcal{A} \times \mathcal{K}$, compute the UCB index

$$U_{(i,k)}(t) = \hat{\mu}_{i,k}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}$$

5: $\boldsymbol{S_t} \leftarrow \mathcal{O}((U_{i,k}(t))_{(i,k)\in\mathcal{A}\times\mathcal{K}})$
6: Take allocation $\boldsymbol{S}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s
7: For each $k \in [K]$, update $n_{a_{k,t},k}$, empirical mean rewards $\hat{\mu}_{a_{k,t},k}$
8: **end for**

Algorithm 7 Correlated UCB for resource allocation with offline oracle $\mathcal{O}$

1: **Input**: Constraints $g_i(\boldsymbol{S})$, Oracle $\mathcal{O}$.
2: For each base arm $(i,k) \in \mathcal{A} \times \mathcal{K}$, $n_{i,k}(t) \leftarrow 0$. {maintain the total number of times base arm (i,k) is played so far.}
3: **for** $t = 1, 2, 3, \ldots$ **do**
4: For each base arm $(j,\ell) \in \mathcal{K} \times \mathcal{A}$, evaluate its KA pseudoUCB indices $U_{(j,\ell),(i,k)}(t) \triangleq \hat{\phi}_{(j,\ell),(i,k)}(t) + B\sqrt{\frac{2\log t}{n_{(i,k)}(t)}}$
5: For each $(j,\ell) \in \mathcal{A} \times \mathcal{K}$, $U_{(j,\ell)}(t) = \min_{(i,k)} U_{(j,\ell),(i,k)}(t)$
6: $\boldsymbol{S_t} \leftarrow \mathcal{O}((U_{i,k}(t))_{(i,k)\in\mathcal{A}\times\mathcal{K}})$
7: Take allocation $\boldsymbol{S}_t$, observe feedback $f_k(a_{k,t}, X_{k,t})$'s
8: Update $n_{(a_{k,t},k)}$, the empirical pseudo-rewards $\hat{\phi}_{(j,\ell),(i,k)}(t)$ for all (j,ℓ), the empirical reward for base arm $(i,k) \in \mathcal{S}_t$
9: **end for**

5.4 Regret bounds and analysis

5.4.1 Main results

We now characterize the performance of our proposed algorithm in terms of regret (See eq (5.4)).

$$Reg_{\alpha}^{\pi}(T;\boldsymbol{D}) = T \cdot \alpha \cdot \mathrm{opt}(\boldsymbol{D}) - \textstyle\sum_{t=1}^{T} r(\boldsymbol{S}_t^{\pi}, \boldsymbol{D}). \tag{5.14}$$

Here, $r(\boldsymbol{S}_t^{\pi}, \boldsymbol{D})$ represents the expected total reward obtained in round t, which can be written as,

$$r(\boldsymbol{S}_t, \boldsymbol{\mu}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K}\sum_{a\in\mathcal{A}} \mu_{i,k}\cdot \mathbb{1}\{i = a_{k,t}\}. \tag{5.15}$$

For the regret analysis, we assume without loss of generality that the rewards are between 0 and 1 for all base arms (i,k). Furthermore, we denote the oracle's optimal budget allocation vector as $\boldsymbol{S}^*$, i.e., the allocation vector that provides an α-optimal solution to the offline resource allocation problem, where $\mathbb{E}\left[f_k(a_k, X_k)\right]$ is known for all base arms. For simplicity, we assume that there is a unique solution $\boldsymbol{S}^*$ to the offline resource allocation problem. Correspondingly, we denote the set of base arms selected in $\boldsymbol{S}^*$ as the set of optimal base arms $\mathcal{S}^*$. To bound the regret, we rely on two properties of $r(\boldsymbol{S}, \mu)$.

Property 1. *(Monotonicity). The expected reward of playing any super arm* $\boldsymbol{S}_t$ *is monotonically increasing with respect to the expectation vector of base arms, i.e., if for all* $(i,k) \in \mathcal{A}\times\mathcal{K}$, *if* $\mu_{i,k} \le \mu'_{i,k}$, *then we have* $r(\boldsymbol{S}_t, \boldsymbol{\mu}) \le r(\boldsymbol{S}_t, \boldsymbol{\mu'})\ \ \forall \boldsymbol{S}_t$.

Property 2. *(Bounded Smoothness).* $\exists$ *an increasing function* $g(.)$ *such that, if* $\boldsymbol{S}_t$ *is the super-arm selected in round* t *and* $||\boldsymbol{\mu}_{\boldsymbol{S}_t} - \boldsymbol{\mu}'_{\boldsymbol{S}_t}||_\infty < \lambda$, *then*

$$|r(\boldsymbol{S}_t, \boldsymbol{\mu}) - r(\boldsymbol{S}_t, \boldsymbol{\mu'})| < g(\lambda).$$

Here, the infinity norm between $\boldsymbol{\mu}_{\boldsymbol{S}_t}$ *and* $\boldsymbol{\mu}'_{\boldsymbol{S}_t}$ *is defined as*
$\max_{(i,k)\in\mathcal{S}_t} |\mu_{(i,k)} - \mu'_{(i,k)}|$ *with* $\mathcal{S}_t$ *denoting the set of base arms played in round* t.

It is easy to see that both properties hold from the definition of $r(\boldsymbol{S}_t, \boldsymbol{\mu})$ in Eq. (5.5). Before stating our main result for the correlated UCB algorithm, we first review the regret bound under the UCB-RA algorithm [21].

Lemma 27. *The regret for UCB-RA algorithm is upper bounded as*

$$\begin{aligned} Reg_\alpha(T, \boldsymbol{D}) &\le \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \Delta_{min}^{(i,k)} \left(\frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2}\right) + 4KA\Delta_{max} \\ &= KA\cdot \mathrm{O}(\log T) + \mathrm{O}(1), \end{aligned} \tag{5.16}$$

$$\begin{aligned} \textit{with}\quad \Delta_{min}^{(i,k)} &= r(\boldsymbol{S}^*, \boldsymbol{\mu}) - \max(r(\boldsymbol{S}, \boldsymbol{\mu}) | \boldsymbol{S} \in \mathcal{S}_B, (i,k)\in\mathcal{A}\times\mathcal{K}), \\ \Delta_{max}^{(i,k)} &= r(\boldsymbol{S}^*, \boldsymbol{\mu}) - \min(r(\boldsymbol{S}, \boldsymbol{\mu}) | \boldsymbol{S} \in \mathcal{S}_B, (i,k)\in\mathcal{A}\times\mathcal{K}), \\ \Delta_{max} &= \max_{(i,k)\in\mathcal{A}\times\mathcal{K}} \Delta_{max}^{(i,k)}, \end{aligned}$$

where $\mathcal{S}_B$ *is the set of all sub-optimal super arms and* $\boldsymbol{S}^*$ *is the oracle's optimal allocation.*

The result follows from the intuition that after the UCB indices of all the base arms are relatively *close* to their true mean rewards, the algorithm selects the budget allocation $\boldsymbol{S}^*$ with high probability. Under the UCB-RA algorithm, each base arm needs to be sampled $\mathrm{O}(\log T)$ times to ensure that the UCB indices are *close* to their true means. Due to which, the regret of UCB-RA algorithm is of the form of $KA \cdot \mathrm{O}(\log T)$. We formalize this intuition for both the UCB-RA and our proposed Corr-UCB-RA algorithms through the following claim. This claim is a novel contribution of our work and it provides an alternative methodology to analyse the generic combinatorial bandit formulation [85] as well.

Claim 1.

$$\textit{If} \quad U_{(i,k)} \geq \mu_{(i,k)} \quad \forall (i,k) \in \mathcal{K} \times \mathcal{A}$$

and the UCB-RA and Corr-UCB-RA algorithms select a budget allocation $\boldsymbol{S_t}$ at round t where,

$$\mu_{(i,k)} \leq U_{(i,k)} < \bar{\mu}_{(i,k)} \quad \forall (i,k) \in \mathcal{S}_t,$$

then $\boldsymbol{S_t}$ is equal to the oracle's optimal allocation $\boldsymbol{S}^*$.
Here, the thresholds $\bar{\mu}_{(i,k)}$ are defined as

$$\bar{\mu}_{(i,k)} = \mu_{(i,k)} + g^{-1}(\Delta_{min}^{(i,k)}).$$

Using this claim, we will show regret bounds for our proposed Corr-UCB-RA algorithm. To state our results, we first define the notion of competitive and non-competitive base arms.

Definition 19 (Competitive and Non-Competitive base arms)**.** *If $\phi_{(j,\ell),(i,k)} \leq \bar{\mu}_{(j,\ell)}$ for some $(i,k) \in \mathcal{S}^*$ then base arm (j,ℓ) is called Non-competitive, otherwise it is called Competitive. Here, $\mathcal{S}^*$ denotes the set of base arms played in the oracle's optimal budget allocation vector $\boldsymbol{S^*}$. Furthermore, we define pseudo-gap of a base arm (j,ℓ) as $\bar{\Delta}_{(j,\ell)} = \bar{\mu}_{(j,\ell)} - \max_{(i,k)\in\mathcal{S}^*} \phi_{(j,\ell),(i,k)}$.*

Note that the pseudo-gap is greater than zero for non-competitive base arms and is less than or equal to zero for competitive base arms. The definition of pseudo-gap is useful to state our regret bounds. Intuitively, a base arm (j,ℓ) is non-competitive if it can be inferred that the mean reward of (j,ℓ) is smaller than the threshold $\bar{\mu}_{(j,\ell)}$ through just the samples of a base arm belonging to the oracle's optimal budget allocation $\mathcal{S}^*$. In what follows, we refer to the total number of competitive base arms as C and the set of competitive base arms as $\mathcal{C}$. As mentioned earlier, the Corr-UCB-RA algorithm selects the budget allocation $\boldsymbol{S}^*$ with high probability if the indices of base arms $U_{(i,k)}$ are *close* to their true means. In the presence of correlations, we show that this can be achieved by sampling competitive base arms $\mathrm{O}(\log T)$ times and non-competitive base arms only O(1) times. This occurs as the non-competitive base arms can be identified as sub-optimal based on samples of optimal base arms. We formalize this intuition to get the following regret bound for our Corr-UCB-RA algorithm.

Theorem 10 (Upper Bound on Cumulative Regret)**.** *The expected cumulative regret of the Correlated-UCB algorithm for resource allocation is upper bounded as*

$$Reg_{\alpha}(T,\boldsymbol{D}) \leq \sum_{(i,k)\in\mathcal{C}} \Delta_{max}^{(i,k)} \left(\frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2} + 2 \right) \quad + \sum_{(i',k')\in\mathcal{K}\times\mathcal{A}\backslash\{\mathcal{C}\}} \Delta_{max}^{(i',k')}(4KAt_0 + 6(KA)^3) \quad + \quad 2(KA)^2\Delta_{max}, \tag{5.17}$$

$$= C\cdot \mathrm{O}(\log T) + \mathrm{O}(1), \tag{5.18}$$

where $\mathcal{C} \subseteq \mathcal{K}\times\mathcal{A}$ is set of competitive base arms with cardinality C, where $t_0 = \inf\left\{\tau \geq 2 : g^{-1}\left(\Delta_{min}^{(i,k)}\right) \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall (i,k), \bar{\Delta}_{(i,k)} \geq 4\sqrt{\frac{2K\log\tau}{\tau}} \quad \forall (i,k) \in \mathcal{A}\times\mathcal{K}\setminus\mathcal{C}\right\}$.

We now present a proof of our Claim 1, which is then used to obtain the provide a proof sketch of Theorem 10. The proof of Claim 1 is of independent interest as well as these techniques can be used to analyse the regret of the UCB algorithm in generic combinatorial bandits as well (e.g., the combinatorial UCB algorithm in [85]).

5.4.2 Proof Sketch

Proof of Claim 1. In total there are $|K|\times|A|$ base arms. Index these base arms with indices z in the set $\{1,2,\ldots|K|\times|A|\}$ such that $\Delta_{\text{min}}^{(1)} \geq \Delta_{\text{min}}^{(2)} \geq \ldots\Delta_{\text{min}}^{(z)} \geq \ldots \geq \Delta_{\text{min}}^{(|K|\times|A|)}$.

We consider a case where, $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\text{min}}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$. Define y to be the smallest index such that base arm y is selected in $\boldsymbol{S}_t$. From definition of base arm y and through Property 2 we have,

$$||U_{\boldsymbol{S}_t}(t) - \mu_{\boldsymbol{S}_t}||_\infty < g^{-1}(\Delta_{\text{min}}^{(y)}) \tag{5.19}$$

$$\Rightarrow |r(\boldsymbol{S}_t,\boldsymbol{U(t)}) - r(\boldsymbol{S}_t,\boldsymbol{\mu})| < \Delta_{\text{min}}^{(y)}. \tag{5.20}$$

As $U_z(t) > \mu_z \ \forall z$, we have the following from the monotonicity condition (Property 1),

$$r(\boldsymbol{S}_t,\boldsymbol{\mu}) + \Delta_{\text{min}}^{(y)} > r(\boldsymbol{S}_t,\boldsymbol{U(t)}) \tag{5.21}$$

$$\geq r(\boldsymbol{S}^*,\boldsymbol{U(t)}) \tag{5.22}$$

$$\geq r(\boldsymbol{S}^*,\boldsymbol{\mu}) \tag{5.23}$$

Here, we have (5.22) as the allocation $\boldsymbol{S_t}$ is obtained from offline oracle and hence it is optimal for the UCB index vector, and its expected reward is larger than the allocation $\boldsymbol{S}^*$. (5.23) arises from the monontonicity condition as $U_z > \mu_z \forall z$. This shows that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(1)}) \quad \forall z \in S_t$ and $U_z > \mu_z \; \forall z$, then the expected reward for the budget allocation S_t,

$$r(\boldsymbol{S_t}, \boldsymbol{\mu}) > r(\boldsymbol{S}^*, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}. \tag{5.24}$$

As base arm y is selected in $\boldsymbol{S_t}$, then by definition of $\Delta_{min}^{(y)}$,

$$\max(r(\boldsymbol{S_t}, \boldsymbol{\mu}) | \boldsymbol{S_t} \in \mathcal{S}_B, (i,k) = y \in \mathcal{S}_t) \leq r(\boldsymbol{S}^*, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}, \tag{5.25}$$

which shows that the maximum reward that can be attained if the allocation $\boldsymbol{S_t}$ was sub-optimal and base arm y was selected is upper bounded by $r(\boldsymbol{S}^*, \boldsymbol{\mu}) - \Delta_{\min}^{(y)}$. Upon comparing (5.25) and (5.24), we conclude that if $\mu_z \leq U_z(t) < \mu_z + g^{-1}(\Delta_{\min}^{(z)}) \quad \forall z \in \mathcal{S}_t$ and $U_z > \mu_z \forall z$, then the budget allocation vector $\boldsymbol{S}_t$ is equal to $\boldsymbol{S}^*$, which is the oracle's unique optimal solution to the budget allocation problem.

Proof of Theorem 10. We now discuss the regret analysis of Theorem 10. In order to bound the regret, we first define the notion of a *responsible* base arm.

Definition 20 (Responsible)**.** *A base arm (i,k) is said to be responsible at round t, if*

1. *It was selected in round t and*
2. $U_{(i,k)}(t) \geq \bar{\mu}_{(i,k)}$

By Claim 1, if a sub-optimal budget allocation was selected in round t, it implies that either $U_{(i,k)}(t) < \mu_{(i,k)}$ for some $(i,k) \in \mathcal{K} \times \mathcal{A}$ or at least one of the selected base arms in $\boldsymbol{S_t}$ was responsible. Therefore, the expected number of rounds in which a sub-optimal allocation was played (referred to as bad rounds) can be upper bounded by

$$\begin{aligned} \mathbb{E}[\text{Bad rounds}(T)] \leq & \sum_{(i,k) \in \mathcal{K} \times \mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \\ & + \sum_{(i,k) \in \mathcal{K} \times \mathcal{A}} \mathbb{E}[n_{U_{(i,k)} < \mu_{(i,k)}}(T)], \end{aligned} \tag{5.26}$$

with $r_{(i,k)}(T)$ denoting the number of rounds for which base arm (i,k) is responsible up until round T and $n_{U_{(i,k)} < \mu_{(i,k)}}(T)$ representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) till round T. This inequality arises as a result of the union bound and linearity of expectation. Moreover, whenever arm (i,k) is responsible in round t, the regret incurred in that round can be upper bounded by by $\Delta_{\max}^{(i,k)}$ (by definition of $\Delta_{\max}^{(i,k)}$ in Lemma 27). In scenarios where, $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k), the regret incurred in that round can be upper bounded by $\Delta_{\max}$ (by definition of $\Delta_{\max}$ in Lemma 27). Using this observation, we can now bound the regret as

$$\begin{aligned}\mathbb{E}[\mathrm{Reg}(T)] \leq & \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} \\ & + \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)] \times \Delta_{\max}. \end{aligned} \tag{5.27}$$

Using Hoeffding's inequality, it can be shown that the second term is upper bounded by an O(1) constant, the details are presented in Lemma 33. To bound the regret in (5.27), we bound $\mathbb{E}\left[r_{(i,k)}(T)\right]$ separately for non-competitive and competitive base arms. More specifically, we show that $\mathbb{E}\left[r_{(i,k)}(T)\right]$ is upper bounded by an O(1) constant for non competitive base arms and is $\mathrm{O}(\log T)$ for competitive base arms. There are two key components to show upper bounds on $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for non-competitive base arm (i,k). Suppose the base arm is non-competitive with respect to (j,ℓ), i.e., $\phi_{(i,k),(j,\ell)} < \bar{\mu}_{(i,k)}$ and $(j,\ell) \in \mathcal{S}^*$.

1. The probability of base arm (i,k) being responsible in round t jointly with the event that $n_{j,\ell}(t) > \frac{2t}{3}$ is *small*.
$$\Pr\left((resp_{(i,k)}(t), n_{(j,\ell)}(t) \geq \frac{2t}{3} \right) \leq t^{-3} \quad \forall t > 3KAt_0.$$
This occurs as upon obtaining a *large* number of samples of base arm (j,ℓ), the expected pseudo-reward of base arm (i,k) is smaller than $\bar{\mu}_{(i,k)}$ with high probability. As a result, the probability that base arm (i,k) is responsible is *small*. The details of this can be seen in Lemma 31.

2. The probability that a sub-optimal budget allocation is made for more than $\frac{t}{3}$ times till round t is upper bounded as,
$$\Pr\left(T^{\text{sub-opt}}(t) \geq \frac{t}{3} \right) \leq 6(KA)^2 \left(\frac{t}{3KA}\right)^{-2} \quad \forall t > 3KAt_0,$$
We show this in Lemma 35 through Lemma 32, Lemma 34 by showing that $r_{(i,k)}(T)$, which is the number of rounds for which base arm (i,k) is responsible till round T, is smaller than $\frac{t}{3KA}$ with high probability. Additionally, $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$, representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ for some (i,k) till round T, is smaller than $\frac{t}{3}$ with high probability. Using these two arguments (1) and (2) above, we bound the expected times a non-competitive base arm (i,k) is responsible until round t in Lemma 36 as

$$\mathbb{E}\left[r_{(i,k)}(T)\right] \leq 3KAt_0 + \sum_{t=3KAt_0}^{T} t^{-3} + 6(KA)^2\left(\frac{t}{3KA}\right)^{-2} \tag{5.28}$$

$$= \mathrm{O}(1). \tag{5.29}$$

Next, we bound the term $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for competitive sub-optimal arms. We do so in Lemma 37, by showing that after base arm (i,k) has been sampled $\mathrm{O}(\log T)$ times, the probability of base arm being responsible at round t decays as t^{-2} and as a result $\mathbb{E}\left[r_{(i,k)}(T)\right]$ is $\mathrm{O}(\log T)$. This combined with (5.29), leads to Theorem 10.

5.4.3 Discussion on the regret bound

Competitive and Non-competitive base arms. Recall that a base arm (i,k) is said to be non-competitive if the expected pseudo-reward of base arm (i,k) with respect to some base $(j,\ell) \in \mathcal{S}^*$ is smaller than $\bar{\mu}_{(i,k)}$. Note that the optimal set of arms $\mathcal{S}^*$, reward distribution of individual base arms is unknown at the beginning and as a result the Corr-UCB-RA initially does not know which base arms are competitive and non-competitive.

Reduction in the effective set of base arms. Upon comparison with the regret of the UCB-RA algorithm, from Lemma 27, we see that our proposed algorithm reduces the regret from $KA \times \mathrm{O}(\log T)$ to $C \times \mathrm{O}(\log T)$, since only C out of the total KA need to be sampled $\mathrm{O}(\log T)$ times before the condition in Claim 1 is met with high probability. As a result, the Corr-UCB-RA only explores C out of the KA base arms explicitly and effectively reduces the problem with KA base arms to one with C base arms.

Bounded regret in certain settings. Whenever the set $\mathcal{C}$ is empty, the proposed Corr-UCB-RA algorithm achieves bounded regret, which is an order-wise improvement over the regret of correlation agnostic UCB-RA algorithm. One scenario in which this can occur is if the functions $f_k(\cdot)$ are invertible with respect to X_k given a_k. More generally, whenever the sub-optimal base arms can be identified as sub-optimal through just the samples of optimal base arms, we get a bounded regret. Note that the algorithm initially has no knowledge about the optimality/sub-optimality of base arms and in such cases it identifies them by sampling the sub-optimal base arms only $\mathrm{O}(1)$ times.

5.5 Continuous budget setting

So far we have studied the resource allocation problem under the assumption that the set $\mathcal{A}$ from which budget a_k for each entity k is allocated is a countable set. In this section, we discuss settings where $\mathcal{A}$ is uncountable. One instance where this could occur is if $a_k \in \mathbb{R}$. In such scenarios, it is still possible to design an algorithm while achieving bounded regret in some cases.

Reward functions are invertible. Suppose the reward functions $f_k(a_k, X_k)$ are invertible in X_k and are known to the algorithm. In this case, it is possible to estimate X_k directly from the reward samples of entity k. Therefore, one can maintain an empirical mean $\hat{X}_k(t)$ for each entity. This empirical mean can then be used to evaluate the upper confidence bound indices for base arm (i, k) as

$$U_{(i,k)}(t) = f_k(i, \hat{X}_k(t)) + \sqrt{\frac{2 \log T}{n_k(t)}},$$

where $n_k(t) = \sum_j n_{j,k}(t)$ and $\hat{X}_k(t) = \frac{\sum_{\tau=1}^{t} g^{-1}_{a_k(\tau),k}(r_k(\tau))}{n_k(t)}$. Here $g_{a_k(\tau),k}(X_k(t)) = f_k(a_k(\tau), X_k(t))$ and $r_k(\tau)$ is the reward attained from entity k at round τ.

One can then use these UCB indices to obtain an allocation $\boldsymbol{S_t}$ from the offline oracle as done in Corr-UCB-RA algorithm, which will then be used to select the super arm in the next round. Using techniques in Section 4.2, it can be shown that this algorithm in cases where reward functions are invertible will lead to an O(1) regret. This occurs as the information about the sub-optimal base arms can be obtained through the samples of optimal super arm.

Non-invertible reward functions. In scenarios where reward functions are non-invertible, it is still possible to extend the Corr-UCB-RA algorithm. This can be done by discretizing the budget space and making assumptions about Lipschitz continuity as done in [21]. Specifically, the regret is affected by the discretization granularity and [21] provided an optimized value for it. After the discretization, we can use Corr-UCB-RA on the countable action set.

5.6 Experimental results

To validate the effectiveness of our algorithm, we conduct experiments on three applications with synthetic and real data. First, we consider a dynamic user allocation problem in wireless networks, where we need to allocate new incoming users to different wireless access points with unknown number of existing users. We evaluate our algorithm in the setting with non-invertible reward function. Next, we study an online server assignment problem, where the servers need to be assigned to different job streams with unknown job arrival rates. Different from the first application, the reward function of this problem is invertible, so it is possible to obtain O(1) regret. However, we also study a partial feedback setting for this application, which leads to sublinear regret. Finally, we apply our algorithm to an online water filling problem [95] that is essential to the power allocation in OFDM systems [96]. It is a continuous budget allocation problem with invertible reward functions, and we study its partial feedback setting as well.

5.6.1 Dynamic User Allocation

In this section, we apply our corr-UCB-RA algorithm to a dynamic user allocation problem in wireless networks. Our goal is maximize the total throughput of wireless access points (APs) by allocating new incoming users to them. The number of existing users associated to each AP is time-varying, which affects

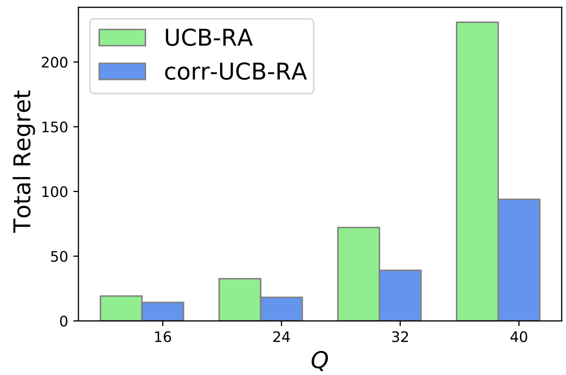
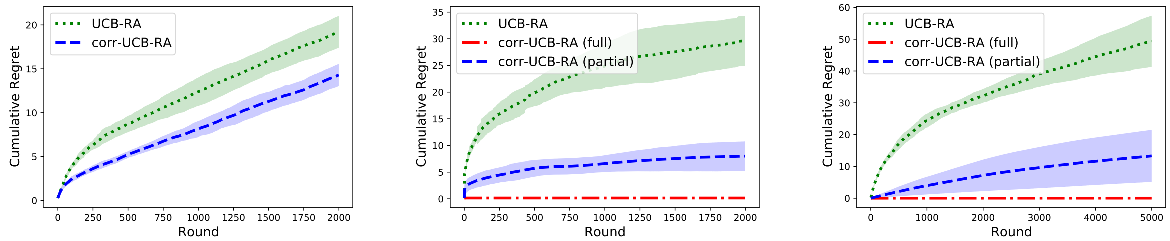


Figure 5.3: Comparison between regret of UCB-RA and Corr-UCB-RA as a function Q (new incoming users) for the application of dynamic user allocation.

the traffic load on the AP. We assume each user has a fixed traffic load of 0.2 and consider the well-known ALOHA protocol [97] for each AP. We consider K APs and Q new incoming users at each round. Let X_k denote the number of existing users in each AP k and a_k denote the number of new users allocated to it. Note that we assume all the users of an AP will leave when the round ends, so a_k in the current round will not affect X_k in the future rounds. Our goal is to maximize the total throughput of all APs:

$$\max_{a_k} \sum_{k=1}^{K} 0.2(X_k + a_k)e^{-0.2(X_k+a_k)}, \text{ s.t. } \sum_{i=k}^{K} a_k = Q, a_k \in \mathbb{N}.$$

We extract $\{X_k\}$, the number of existing users in each AP, from a real-world dataset [98]. We choose 4 APs (91, 92, 94, 95) on the 3rd floor of Building 3 on campus, and record their associated users from 13:00 to 16:00 on March 2, 2015. The detailed distribution of the number of existing users on different access points can be found in Figure 5.6. In our experiment, at each round, we first sample $\{X_k\}$ from the extracted distribution, then allocate $Q = 8$ new users to these four APs. Since the throughput function is non-invertible, our algorithm cannot directly infer X_k from the observed throughput of each AP and needs to maintain the pseudoUCB indices of base arms as explained in Section 5.3. We compare it with the UCB-RA algorithm. Figure 5.4a shows the average regrets with 95% confidence interval over 20 experiments. The result is consistent with our analysis in Section 5.4: corr-UCB-RA achieves 25% less regret than correlation agnostic UCB-RA algorithm. This occurs as the corr-UCB-RA algorithm is able to make use of the correlations between the reward of base arms to incur a regret of $C \cdot O(\log T)$ as opposed to $KA \cdot O(\log T)$. We also show the relationship between Q and the total regrets after 2000 rounds in Figure 5.3: with the increase of Q, the total regret of corr-UCB-RA increases much more slowly than that of UCB-RA.



(a) Dynamic User Allocation (b) Online server assignment. (c) Online water filling.

Figure 5.4: Performance comparison between the Corr-UCB-RA and the UCB-RA algorithm for different application problems.

5.6.2 Online Server Assignment

We consider 4 independent job streams (i.e., $K = 4$) with unknown expected job arrival rates $\boldsymbol{\lambda} = (0.2, 0.4, 0.6, 0.8)$. For each job stream k, the realized job arrival rate X_k follows a uniform distribution $U(\lambda_k - 0.1, \lambda_k + 0.1)$. We assume each job stream has one initial server to ensure it is a stable system with bounded expected waiting time. There are 8 additional servers (i.e., $Q = 8$) to be assigned and we denote the number of additional servers allocated to stream k as a_k. We assume the service rate of all servers as 1, and our goal is to minimize the average expected waiting time of all job streams:

$$\min_{a_k} \sum_{k=1}^{K} \frac{1}{1 - \frac{X_k}{a_k+1}} \cdot \frac{X_k}{\sum_{k=1}^{K} X_k}, \quad \text{s.t.} \sum_{i=k}^{K} a_k \leq Q, a_k \in \mathbb{N}.$$

We consider both the full feedback and the partial feedback settings. In the full feedback setting, we assume the waiting times of all job stream are always observable. Since the waiting time function is invertible, our algorithm can directly infer $\{X_k\}$ and update the pseudo-rewards of other base arms as per (5.8). Notice that our goal is to minimize the expected waiting time, so we need to maintain the lower confidence bound (LCB) indices of all base arms, instead of the UCB indices for reward maximization and correspondingly pseudo-rewards would be lower bounds on conditional expected reward. In the partial feedback setting, the waiting time can only be observed when $a_k \geq 1$, i.e., at least one additional server is assigned to stream K. When no server is assigned to stream K, the pseudo-reward of other assignments with respect to such an assignment is set to the minimum possible reward. We repeat the experiment 20 times and Figure 5.4b shows the average regrets with 95% confidence interval. In the full feedback setting, corr-UCB-RA obtains O(1) regret as there is no cost for inferring $\{X_k\}$. In the partial feedback setting, corr-UCB-RA has to balance between the actions of $a_k = 0$ and $a_k \geq 0$, which incurs a sublinear regret. It still outperforms UCB-RA due to the utilization of correlation information.

5.6.3 Online Water Filling

We finally consider the water filling problem where a total amount of one unit power has to be assigned to 4 communication channels, i.e., $Q = 1, K = 4$, with the objective of maximizing the total throughput. The throughput of the k^{th} channel is given by $\log(X_k + a_k)$, where a_k represents the power allocated to channel k and X_k represents the floor above the baseline at which power can be added to the channel. It can be written as a convex optimization problem:

$$\max_{a_k} \sum_{k=1}^{K} \log(X_k + a_k), \;\; \text{s.t.} \sum_{i=k}^{K} a_k \leq Q, a_k \geq 0.$$

For the online water filling problem, the $\{X_k\}$ are unknown and need to be learned. For each channel k, we assume the expectation $\mu_k = \mathbb{E}[X_k]$ is uniformly sampled from $[0.8, 1.2]$, and the realization of X_k follows a uniform distribution $U(\mu_k - 0.5, \mu_k + 0.5)$. As it is a online continuous resource allocation problem, we choose UCB-RA algorithm with discretization granularity 0.2 (i.e., $\mathcal{A} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$) as the baseline. Similar to the online server assignment problem, we consider both the full feedback and the partial feedback settings. In the full feedback setting, the throughput $\log(X_k + a_k)$ is always observable. Since the reward function is invertible, our algorithm can directly infer $\{X_k\}$ and update the pseudoUCB indices as described in Section 5.5. In the partial feedback setting, we assume $\log(X_k + a_k)$ can be observed only if $a_k \geq 0.2$. For channel k with $a_k < 0.2$, we update the pseudo-rewards of other base arms with the maximum possible rewards. We repeat the experiment 20 times and Figure 5.4c shows the average regrets with 95% confidence interval. We see that corr-UCB-RA algorithm achieves significantly reduced regret relative to UCB-RA in both the full feedback and the partial feedback settings. For the full feedback case, corr-UCB-RA obtains O(1) regret as the water filling reward function is invertible and there is no cost in inferring $\{\beta_k\}$. For the partial feedback case, since the minimal power needs to be 0.2 to observe the throughput, corr-UCB-RA needs to balance between the actions of $a_k < 0.2$ and $a_k \geq 0.2$, due to which it incurs a sublinear regret. The regret is still smaller than UCB-RA as it makes use of the available correlation information.

Figure 5.6 shows the distribution of the number of associated users on the access points considered in Section 5.6.1.

5.7 Concluding remarks

In this chapter, we study the problem of sequential resource allocation by modeling it through a combinatorial bandit framework, where the allocation of a budget to an entity is considered as a base arm. In several practical settings, rewards received under different budget allocations are often correlated. We

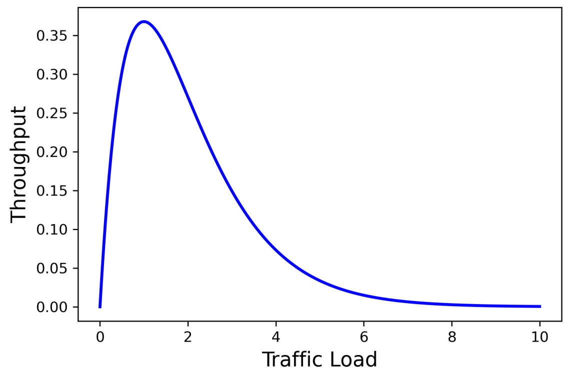


Figure 5.5: Relationship between the throughput and the traffic load.

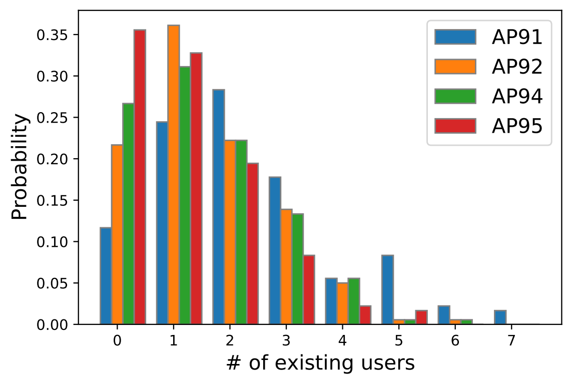


Figure 5.6: Distribution of the number of existing users on different access points for dynamic user allocation.

propose a novel correlated combinatorial bandit framework to tackle the online resource allocation problem. In particular, we model the correlations through *pseudo-rewards*, which represent an upper bound on the conditional expected reward of a budget-entity pair. Using the knowledge of these pseudo-rewards, we propose the correlated UCB algorithm for resource allocation (Corr-UCB-RA) which incurs a regret of $C \cdot \mathrm{O}(\log T)$ as opposed to $KA \cdot \log T$ regret attained by prior correlation agnostic approach in [21]. The value of C can be much smaller than KA and can even be 0 in certain settings, under which our proposed Corr-UCB-RA algorithm attains $\mathrm{O}(1)$ regret. These results are validated by our experimental results on multiple different application settings. While we study this problem in the context of online resource allocation, the algorithm and analysis could be easily extended to the general combinatorial bandit framework [85] as well. An interesting future direction is to learn correlations in an online manner. As multiple base arms are sampled in each round, it is possible to learn correlation information on the go and subsequently use them for budget allocation in the future rounds. We believe this is a challenging open problem which will require non-trivial extensions of the ideas proposed in this work.

5.8 Full proofs

5.8.1 Standard Results from Previous Works

Fact 4 (Hoeffding's inequality). *Let $Z_1, Z_2 \dots Z_n$ be i.i.d random variables bounded between $[a, b] : a \leq Z_i \leq b$, then for any $\delta > 0$, we have*

$$\Pr\left(\left|\frac{\sum_{i=1}^{n} Z_i}{n} - \mathbb{E}\left[Z_i\right]\right| \geq \delta\right) \leq \exp\left(\frac{-2n\delta^2}{(b-a)^2}\right).$$

Lemma 28 (Standard result used in bandit literature). *If $\hat{\mu}_{k,n_k(t)}$ denotes the empirical mean of arm k by sampling arm k $n_k(t)$ times through any algorithm and μ_k denotes the mean reward of arm k, then we have*

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) \leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right).$$

Proof. Let $Z_1, Z_2, ...Z_t$ be the reward samples of arm k drawn separately. If the algorithm chooses to select arm k for m^{th} time, then it observes reward Z_m. Then the probability of observing the event $\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1$ can be upper bounded as follows,

$$\Pr\left(\hat{\mu}_{k,n_k(t)} - \mu_k \geq \epsilon, \tau_2 \geq n_k(t) \geq \tau_1\right) = \Pr\left(\left(\frac{\sum_{i=1}^{n_k(t)} Z_i}{n_k(t)} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{5.30}$$

$$\leq \Pr\left(\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right), \tau_2 \geq n_k(t) \geq \tau_1\right) \tag{5.31}$$

$$\leq \Pr\left(\bigcup_{m=\tau_1}^{\tau_2} \frac{\sum_{i=1}^{m} Z_i}{m} - \mu_k \geq \epsilon\right) \tag{5.32}$$

$$\leq \sum_{s=\tau_1}^{\tau_2} \exp\left(-2s\epsilon^2\right). \tag{5.33}$$

□

Lemma 29 (From Proof of Theorem 1 in [32]). *Let $U_k(t)$ denote the UCB index of arm k at round t, and $\mu_k = \mathbb{E}\left[g_k(X)\right]$ denote the mean reward of that arm. Then, we have*

$$\Pr(\mu_k > I_k(t)) \leq t^{-3}.$$

Observe that this bound does not depend on the number $n_k(t)$ of times arm k is pulled. UCB index is defined as $U_k(t) = \hat{\mu}_{k,n_k(t)} + \sqrt{\frac{2\log t}{n_k(t)}}$.

Proof. This proof follows directly from [32]. We present the proof here for completeness as we use this frequently in the chapter.

$$\Pr(\mu_k > I_k(t)) = \Pr\left(\mu_k > \hat{\mu}_{k,n_k(t)} + \sqrt{\frac{2\log t}{n_k(t)}}\right) \tag{5.34}$$
$$\leq \sum_{m=1}^{t} \Pr\left(\mu_k > \hat{\mu}_{k,m} + \sqrt{\frac{2\log t}{m}}\right) \tag{5.35}$$
$$= \sum_{m=1}^{t} \Pr\left(\hat{\mu}_{k,m} - \mu_k < -\sqrt{\frac{2\log t}{m}}\right) \tag{5.36}$$
$$\leq \sum_{m=1}^{t} \exp\left(-2m\frac{2\log t}{m}\right) \tag{5.37}$$
$$= \sum_{m=1}^{t} t^{-4} \tag{5.38}$$
$$= t^{-3}. \tag{5.39}$$

where (5.35) follows from the union bound and is a standard trick (Lemma 28) to deal with random variable $n_k(t)$. We use this trick repeatedly in the proofs. We have (5.37) from the Hoeffding's inequality. □

5.8.2 Bounding contribution of Non-competitive base arms

Lemma 30. *If the base arm (i,k) is non-competitive with respect to (j,ℓ), i.e., $\phi_{(i,k),(j,\ell)} < \bar{\mu}_{(i,k)}$, where $(j,\ell) = \arg\min_{(j,\ell)\in\mathcal{S}^*} \phi_{(i,k),(j,\ell)}$ then, the probability that base arm (i,k) is* responsible *at round t jointly with $n_{(j,\ell)}$ being larger than $\frac{t}{KA}$ is upper bounded as,*

$$\Pr\left(resp_{(k,a)}(t), n_{(j,\ell)}(t) \geq \frac{t}{4KA}\right) \leq t\exp\left(\frac{-2t\tilde{\Delta}^2_{(i,k)}}{AK}\right).$$

Moreover, if the pseudo-gap $\tilde{\Delta}_{i,k} = \bar{\mu}_{(i,k)} - \phi_{(i,k),(j,\ell)} \geq \sqrt{4\frac{2KA\log t_0}{t_0}}$ for some constant $t_0 > 0$. Then,

$$\Pr\left(resp_{(k,a)}(t), n_{(j,\ell)}(t) \geq \frac{t}{4KA}\right) \leq t^{-3} \quad \forall t > t_0.$$

Proof. We now bound this probability as,

$$\Pr\left(\text{resp}_{(i,k)}(t), n_{(i,k)}(t) \geq s\right) \leq \Pr\left(U_{(i,k)}(t) > \bar{\mu}_{(i,k)}, n_{(j,\ell)}(t) \geq s\right) \tag{5.40}$$
$$\leq \Pr\left(\hat{\phi}_{(i,k),(j,\ell)}(t) + \sqrt{\frac{2\log t}{n_{(j,\ell)}(t)}} > \bar{\mu}_{(i,k)}, n_{(j,\ell)} \geq s\right) \tag{5.41}$$

$$= \Pr\left(\hat{\phi}_{(i,k),(j,\ell)}(t) - \phi_{(i,k),(j,\ell)} > \bar{\mu}_{(i,k)} - \phi_{(i,k),(j,\ell)} - \sqrt{\frac{2\log t}{n_{(j,\ell)}(t)}}, n_{(j,\ell)}(t) \geq s\right) \tag{5.42}$$

$$= \Pr\left(\hat{\phi}_{(i,k),(j,\ell)}(t) - \phi_{(i,k),(j,\ell)} > \bar{\Delta}_{(i,k)} - \sqrt{\frac{2\log t}{n_{(j,\ell)}(t)}}, n_{(j,\ell)}(t) \geq s\right) \tag{5.43}$$

$$\leq t \exp\left(-2s\left(\bar{\Delta}_{(i,k)} - \sqrt{\frac{2\log t}{s}}\right)^2\right) \tag{5.44}$$

$$\leq t^{-3} \exp\left(-2s\left(\bar{\Delta}^2_{(i,k)} - 2\bar{\Delta}_{(i,k)}\sqrt{\frac{2\log t}{s}}\right)\right) \tag{5.45}$$

$$\leq t^{-3} \quad \text{for all } t > t_0. \tag{5.46}$$

From the definition of a responsible arm, we have (5.40) as this condition needs to be satisfied in order for base arm (i,k) to be responsible at round t. By definition of index $U_(i,k)(t) = \min_{(j,\ell)} U_{(i,k),(j,\ell)}$, we get (5.41). Inequality (5.44) follows from Hoeffding's inequality and the term t before the exponent in (5.44) arises as the random variable $n_{(j,\ell)}(t)$ can take values from s to t (Lemma 28). Inequality (5.46) follows from the fact that $s > \frac{t}{4KA}$ and $\bar{\Delta}_{(i,k)} \geq 4\sqrt{\frac{2KA\log t_0}{t_0}}$ for some constant $t_0 > 0$. □

Lemma 31. *The probability that any base arm (i,k) is responsible at round t jointly with the event that it has been sampled for at least $s > \frac{t}{KA}$ rounds till round t is upper bounded as*

$$\Pr\left(resp_{(i,k)}(t), n_{(i,k)}(t) \geq s\right) \leq t^{-3} \quad \textit{for} \ \ s > \frac{t}{4KA} \quad \forall t > t_0,$$

where t_0 is the least integer larger than 2 that satisfies $g^{-1}(\Delta^{(i,k)}_{min}) \geq 4\sqrt{\frac{2KA\log t_0}{t_0}}$.

Proof.

$$\Pr\left(\text{resp}_{(i,k)}(t), n_{(i,k)}(t) \geq s\right) \leq \Pr\left(U_{(i,k)}(t) > \bar{\mu}_{(i,k)}, n_{(i,k)}(t) \geq s\right) \tag{5.47}$$

$$\leq \Pr\left(\hat{\mu}_{(i,k)}(t) + \sqrt{\frac{2\log t}{n_{(i,k)}(t)}} > \bar{\mu}_{(i,k)}, n_{(i,k)}(t) \geq s\right) \tag{5.48}$$

$$= \Pr\left(\hat{\mu}_{(i,k)}(t) - \mu_{(i,k)} > \bar{\mu}_{(i,k)} - \mu_{(i,k)} - \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}, n_{(i,k)}(t) \geq s\right) \tag{5.49}$$

$$= \Pr\left(\hat{\mu}_{(i,k)}(t) - \mu_{(i,k)} > g^{-1}(\Delta_{(i,k)}) - \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}, n_{(i,k)}(t) \geq s\right) \tag{5.50}$$

$$\leq t \exp\left(-2s\left(g^{-1}(\Delta^{(i,k)}_{\min}) - \sqrt{\frac{2\log t}{s}}\right)^2\right) \tag{5.51}$$

$$\leq t^{-3} \exp\left(-2s\left(\left(g^{-1}(\Delta^{(i,k)}_{\min})\right)^2 - 2g^{-1}(\Delta^{(i,k)}_{\min})\sqrt{\frac{2\log t}{s}}\right)\right) \tag{5.52}$$

$$\leq t^{-3} \quad \text{for all } t > t_0. \tag{5.53}$$

From the definition of a responsible arm, we have (5.47) as this condition needs to be satisfied in order for base arm (i,k) to be responsible at round t. By definition of index $U_{(i,k)}(t) = \min_{(j,\ell)} U_{(i,k),(j,\ell)}$, we get (5.48). Inequality (5.51) follows from Hoeffding's inequality and the term t before the exponent in (5.51) arises as the random variable $n_{(i,k)}(t)$ can take values from s to t (Lemma 28). Inequality (5.53) follows from the fact that $s > \frac{t}{4KA}$ and $g^{-1}\left(\Delta_{\min}^{(i,k)}\right) \geq 4\sqrt{\frac{2KA \log t_0}{t_0}}$ for some constant $t_0 > 0$. □

Lemma 32. *The probability that any base arm $(i,k) \in \mathcal{A} \times \mathcal{K}$ is* responsible *for more than $\frac{t}{3KA}$ rounds up until round t is upper bounded as,*

$$\Pr\left(r_{(i,k)}(t) > \frac{t}{3KA}\right) \leq 3KA\left(\frac{t}{3KA}\right)^{-2} \quad \forall t > 3KAt_0,$$

where t_0 is the least integer larger than 2 that satisfies $g^{-1}(\Delta_{(i,k)}) \geq 4\sqrt{\frac{2KA \log t_0}{t_0}}$.

Proof.

$$\Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}\right) = \Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}, r_{(i,k)}(t-1) \geq \frac{t}{3KA}\right) + \Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}, r_{(i,k)}(t-1) = \frac{t}{3KA} - 1\right) \tag{5.54}$$

$$\leq \Pr\left(r_{(i,k)}(t-1) \geq \frac{t}{3KA}\right) + \Pr\left(\mathrm{resp}_{(i,k)}(t), r_{(i,k)}(t-1) = \frac{t}{3KA} - 1\right) \tag{5.55}$$

$$\leq \Pr\left(r_{(i,k)}(t-1) \geq \frac{t}{3KA}\right) + \Pr\left(\mathrm{resp}_{(i,k)}(t), r_{(i,k)}(t-1) \geq \frac{t}{3KA} - 1\right) \tag{5.56}$$

$$\leq \Pr\left(r_{(i,k)}(t-1) \geq \frac{t}{3KA}\right) + (t-1)^{-3}, \tag{5.57}$$

with (5.57) coming from Lemma 31. This gives us

$$\Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}\right) - \Pr\left(r_{(i,k)}(t-1) \geq \frac{t}{3KA}\right) \leq (t-1)^{-3}, \quad \forall (t-1) > t_0.$$

Now consider the summation,

$$\sum_{\tau=\frac{t}{3KA}}^{t} \Pr\left(r_{(i,k)}(\tau) \geq \frac{t}{3KA}\right) - \Pr\left(r_{(i,k)}(\tau-1) \geq \frac{t}{3KA}\right) \leq \sum_{\tau=\frac{t}{3KA}}^{t} (\tau-1)^{-3}.$$

This gives us,

$$\Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}\right) - \Pr\left(r_{(i,k)}\left(\frac{t}{3KA} - 1\right) \geq \frac{t}{3KA}\right) \leq \sum_{\tau=\frac{t}{3KA}}^{t} (\tau-1)^{-3}.$$

Since $\Pr\left(r_{(i,k)}\left(\frac{t}{3KA} - 1\right) \geq \frac{t}{3KA}\right) = 0$, we have,

$$\Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}\right) \leq \sum_{\tau=\frac{t}{3KA}}^{t} (\tau-1)^{-3} \tag{5.58}$$

$$\leq 3KA\left(\frac{t}{3KA}\right)^{-2} \quad \forall t > 3KAt_0. \tag{5.59}$$

□

Lemma 33. *The probability that the index $U_{(i,k)}(t)$ for a base arm (i,k) is smaller than the mean reward of base arm (i,k) is upper bounded as*

$$\Pr(U_{(i,k)}(t) < \mu_{(i,k)}) \leq KA \times t^{-3}.$$

Moreover, the expected number of times $U_{(i,k)}(t) < \mu_{(i,k)}$ till round T is upper bounded as

$$\mathbb{E}\left[n_{\mu_{(i,k)} > U_{(i,k)}(t)}\right] \leq \sum_{t=1}^{T} KAt^{-3},$$

which is $O(1)$.

Proof.

$$\Pr(U_{(i,k)}(t) < \mu_{(i,k)}) = \Pr\left(\bigcup_{(j,\ell)} U_{(i,k),(j,\ell)} < \mu_{(i,k)}\right) \tag{5.60}$$

$$\leq \sum_{(j,\ell)} \Pr\left(\hat{\phi}_{(i,k),(j,\ell)}(t) + \sqrt{\frac{2\log t}{n_{(j,\ell)}(t-1)}} \leq \mu_{(i,k)}\right) \tag{5.61}$$

$$\leq \sum_{(j,\ell)} \Pr\left(\hat{\phi}_{(i,k),(j,\ell)}(t) + \sqrt{\frac{2\log t}{n_{(j,\ell)}(t)}} \leq \phi_{(i,k),(j,\ell)}\right) \tag{5.62}$$

$$\leq (KA)t^{-3}. \tag{5.63}$$

Here, (5.60) comes from the definition of $U_{(i,k)} = \min_{(j,\ell)} U_{(i,k),(j,\ell)}(t)$. We get (5.61) from union bound and the definition of $U_{(i,k),(j,\ell)}(t)$. Inequality (5.62) arises as $\phi_{(i,k),(j,\ell)}$ upper bounds on conditional expectation. The last inequality arises due to Lemma 29.

Subsequently, $\mathbb{E}\left[n_{\mu_{(i,k)} > U_{(i,k)}(t)}\right] = \sum_{t=1}^{T} \Pr(U_{(i,k)}(t) < \mu_{(i,k)})$, which provides the desired bound on $\mathbb{E}\left[n_{\mu_{(i,k)} > U_{(i,k)}(t)}\right]$.

□

Lemma 34. *Let $n_{\mu > U}(t)$ denote the number of rounds in which $\mu_{(i,k)} > U_{(i,k)}$ for some $(i,k) \in \mathcal{K} \times \mathcal{A}$. The probability of such rounds being more than $\frac{t}{3}$ till round t is upper bounded as,*

$$\Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}\right) \leq 3(KA)^2 \left(\frac{t}{3}\right)^{-2} \forall t.$$

Proof.

$$\Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}\right) = \Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}, n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}, n_{\mu>U}(t-1) = \frac{t}{3} - 1\right) \tag{5.64}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \Pr\left(\mu_{(i,k)} \geq U_{(i,k)}(t) \text{ for some } (i,k), n_{\mu>U}(t-1) = \frac{t}{3} - 1\right) \tag{5.65}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \sum_{(i,k)} \Pr\left(\mu_{(i,k)} \geq U_{(i,k)}(t), n_{\mu>U}(t-1) = \frac{t}{3} - 1\right) \tag{5.66}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \sum_{(i,k)} \Pr\left(\mu_{(i,k)} \geq U_{(i,k)}(t)\right) \tag{5.67}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \sum_{(i,k)} \sum_{(j,\ell)} \Pr\left(\mu_{(i,k)} \geq U_{(i,k),(j,\ell)}(t)\right) \tag{5.68}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \sum_{(i,k)} \sum_{(j,\ell)} \Pr(\left(\hat{\phi}_{(i,k),(j,\ell)}(t) + \sqrt{\frac{2\log t}{n_{(j,\ell)}(t-1)}} \leq \mu_{(i,k)}\right) \tag{5.69}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + \sum_{(i,k)} \sum_{(j,\ell)} \Pr(\left(\hat{\phi}_{(i,k),(j,\ell)}(t) + \sqrt{\frac{2\log t}{n_{(j,\ell)}(t-1)}} \leq \phi_{(i,k),(j,\ell)}\right) \tag{5.70}$$

$$\leq \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) + (KA)^2(t-1)^{-3}. \tag{5.71}$$

The steps (5.67) to (5.71) follow through the arguments made in Lemma 33.

This gives us

$$\Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}\right) - \Pr\left(n_{\mu>U}(t-1) \geq \frac{t}{3}\right) \leq (KA)^2(t-1)^{-3}, \quad \forall (t-1).$$

Now consider the summation,

$$\sum_{\tau=\frac{t}{3}}^{t} \Pr\left(n_{\mu>U}(\tau) \geq \frac{t}{3}\right) - \Pr\left(n_{\mu>U}(\tau-1) \geq \frac{t}{3}\right) \leq \sum_{\tau=\frac{t}{3}}^{t} (KA)^2(\tau-1)^{-3}.$$

This gives us,

$$\Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}\right) - \Pr\left(n_{\mu>U}\left(\frac{t}{3}-1\right) \geq \frac{t}{3}\right) \leq \sum_{\tau=\frac{t}{3}}^{t} (KA)^2(\tau-1)^{-3}.$$

Since $\Pr\left(n_{\mu>U}\left(\frac{t}{3}-1\right) \geq \frac{t}{3}\right) = 0$, we have,

$$\Pr\left(n_{\mu>U}(t) \geq \frac{t}{3}\right) \leq \sum_{\tau=\frac{t}{3}}^{t} (KA)^2(\tau-1)^{-3} \tag{5.72}$$

$$\leq 3(KA)^2 \left(\frac{t}{3}\right)^{-2} \qquad \forall t. \tag{5.73}$$

□

Lemma 35. *The probability that a sub-optimal budget allocation is pulled for more than $\frac{t}{3}$ times till round t is upper bounded as,*

$$\Pr\left(T^{sub\text{-}opt}(t) \geq \frac{t}{3}\right) \leq 6(KA)^2 \left(\frac{t}{3KA}\right)^{-2} \qquad \forall t > 3KAt_0,$$

with $T^{sub\text{-}opt}(t)$ denoting the number of sub-optimal budget allocations made till round t.

Proof. From Claim 1, the number of sub-optimal rounds, denoted by $T^{\text{sub-opt}}$ can be written as the union of the rounds with at-least one responsible arm and the rounds in which $\mu_{(i,k)} > U_{(i,k)}$ for some (i,k). This can be be upper bounded as follows using the union bound,

$$T^{\text{sub-opt}} \leq \sum_j r_{(k,a)}(t) + n_{\mu>U} \tag{5.74}$$

Furthermore, the probability that $T^{\text{sub-opt}}(t) \geq \frac{t}{3}$ can be upper bounded as

$$\Pr\left(T^{\text{sub-opt}} \geq \frac{t}{3}\right) \leq \sum_{(i,k)} \Pr\left(r_{(i,k)}(t) \geq \frac{t}{3KA}\right) + \Pr\left(n_{\mu>U} \geq \frac{t}{3}\right) \tag{5.75}$$

This follows through the argument that at least one of these events need to occur for the number of sub-optimal rounds to be at-least $T/3$. The two probabilities are upper bounded by $3KA\left(\frac{t}{3KA}\right)^{-2}$ and $3(KA)^2\left(\frac{t}{3}\right)^{-2} \forall t \geq 3KAt_0$ through Lemma 33, Lemma 34 respectively. As a result,

$$\Pr\left(T^{\text{sub-opt}} \geq \frac{t}{3}\right) \leq 3(KA)^2\left(\frac{t}{3KA}\right)^{-2} + 3(KA)^2\left(\frac{t}{3}\right)^{-2} \forall t \geq 3KAt_0. \tag{5.76}$$

□

Lemma 36. *The expected number of times a non-competitive base arm (i,k) is responsible up until round T is upper bounded as*

$$\mathbb{E}\left[r_{(k,a)}(T)\right] \leq 3KAt_0 + \sum_{t=3KAt_0}^{T} t^{-3} + 6(KA)^2\left(\frac{t}{3KA}\right)^{-2} \tag{5.77}$$

$$= O(1). \tag{5.78}$$

Proof.

$$\mathbb{E}\left[r_{(i,k)}(t)\right] = \sum_{t=1}^{T} \Pr\left(\text{resp}_{(i,k)}(t), T^{\text{sub-opt}}(t) \leq \frac{t}{3}\right) + \Pr\left(\text{resp}_{(i,k)}(t), T^{\text{sub-opt}}(t) \geq \frac{t}{3}\right) \tag{5.79}$$

$$\leq \sum_{t=1}^{T} \Pr\left(\text{resp}_{(i,k)}(t), T^{\text{opt}}(t) \geq \frac{2t}{3}\right) + \Pr\left(T^{\text{sub-opt}}(t) \geq \frac{t}{3}\right) \tag{5.80}$$

$$\leq \sum_{t=1}^{T} \Pr\left(\text{resp}_{(i,k)}(t), n_{(j,\ell)} \geq \frac{2t}{3}\right) + \Pr\left(T^{\text{sub-opt}}(t) \geq \frac{t}{3}\right), \quad (j,\ell) \in \mathcal{S}^* \tag{5.81}$$

$$\leq 3KAt_0 + \sum_{t=3KAt_0}^{T} t^{-3} + 6(KA)^2\left(\frac{t}{3KA}\right)^{-2} \tag{5.82}$$

$$\leq 3KAt_0 + 2 + 6(KA)^3 \tag{5.83}$$

$$= O(1) \tag{5.84}$$

Here, (5.82) follows from Lemma 35 and Lemma 30.

□

5.8.3 Bounding contribution of Competitive base arms

Lemma 37 (Contribution of competitive base arms)**.** *The expected number of times a competitive base arm (i,k) is responsible up until round T is upper bounded as*

$$\mathbb{E}\left[r_{(i,k)}(T)\right] \leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{min}^{(i,k)}\right)\right)^2} + 2 \tag{5.85}$$

$$= \mathrm{O}(\log T). \tag{5.86}$$

Proof.

$$\mathbb{E}\left[r_{(i,k)}(T)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\text{base arm (i,k) is responsible at round t}}\right] \tag{5.87}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2} + \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}_{\text{base arm (i,k) is responsible at round t}, \;\; r_{(i,k)}(t) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2}}\right] \tag{5.88}$$

$$= \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2} + \sum_{t=1}^{T} \Pr\left(\text{resp}_{(i,k)}(t), r_{(k,a)}(t-1) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2}\right) \tag{5.89}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2} + \sum_{t=1}^{T} \Pr\left(U_{(i,k)}(t) \geq \bar{\mu}_{(i,k)}, r_{(k,a)}(t-1) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2}\right) \tag{5.90}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2} + \sum_{t=1}^{T} \Pr\left(\hat{\mu}_{(i,k)}(t) \geq \bar{\mu}_{(i,k)} - \sqrt{\frac{2\log t}{n_{(i,k)}(t)}}, r_{(i,k)}(t-1) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2}\right) \tag{5.91}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2} + \sum_{t=1}^{T} \Pr\left(\hat{\mu}_{(i,k)}(t) \geq \mu_{(i,k)} + \frac{g^{-1}(\Delta_{\text{min}}^{(i,k)})}{2}, r_{(i,k)}(t-1) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(i,k)}\right)\right)^2}\right) \tag{5.92}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(k,a)}\right)\right)^2} + \sum_{t=1}^{T} t^{-2} \tag{5.93}$$

$$\leq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\text{min}}^{(k,a)}\right)\right)^2} + 2 \tag{5.94}$$

$$= \mathrm{O}(\log T) \tag{5.95}$$

□

This proof closely follows the analysis of $O(\log T)$ regret bound under the UCB algorithm for classical multi-armed bandits [32]. Here, (5.92) follows from the fact that $r_{(i,k)}(t-1) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\min}^{(i,k)}\right)\right)^2}$ implies that $n_{(i,k)}(t) \geq \frac{8\log T}{\left(g^{-1}\left(\Delta_{\min}^{(i,k)}\right)\right)^2}$. The inequality (5.93) then follows them the Hoeffding's inequality yielding the desired bound on $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for competitive base arms.

5.8.4 Proof of Theorem 10

By Claim 1, if a sub-optimal super arm allocation was played, it implies that either $U_{(i,k)}(t) < \mu_{(i,k)}$ for some $(i,k) \in \mathcal{K} \times \mathcal{A}$ or at least one of the base arms in $\boldsymbol{S_t}$ was responsible. Therefore the expected number of rounds in which a sub-optimal allocation was played (referred to as bad rounds) can be upper bounded by

$$\begin{aligned}\mathbb{E}[\text{Bad rounds}(T)] \leq &\sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \\ &+ \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)], \end{aligned} \tag{5.96}$$

with $r_{(i,k)}(T)$ denoting the number of times base arm (i,k) is responsible up until round T and $n_{U_{(i,k)}<\mu_{(i,k)}}(T)$ representing the number of rounds in which $U_{(i,k)}(t) < \mu_{(i,k)}$ till round T. This inequality arises as a result of union bound and linearity of expectation. Moreover, whenever arm (i,k) is responsible in round t, the regret incurred in that round can be upper bounded by by $\Delta_{\max}^{(i,k)}$ (by definition of $\Delta_{\max}^{(i,k)}$ in Lemma 27). In scenarios where, $U_{(i,k)}(t) < \mu_{(i,k)}$, the regret incurred in that round can be upper bounded by $\Delta_{\max}$ (by definition of $\Delta_{\max}$ in Lemma 27). Using this observation, we can now bound regret as

$$\begin{aligned}\mathbb{E}[\text{Reg}(T)] \leq &\sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} \\ &+ \sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)] \times \Delta_{\max}. \end{aligned} \tag{5.97}$$

This can further be broken down by separating the $\mathbb{E}\left[r_{(i,k)}(T)\right]$ term for competitive and non-competitive base arms. As a result,

$$\begin{aligned}\mathbb{E}[\text{Reg}(T)] \leq &\sum_{(i,k)\in\mathcal{C}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} + \sum_{(i,k)\in(\mathcal{K}\times\mathcal{A})\setminus\mathcal{C}} \mathbb{E}[r_{(i,k)}(T)] \times \Delta_{\max}^{(i,k)} + \\ &\sum_{(i,k)\in\mathcal{K}\times\mathcal{A}} \mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)] \times \Delta_{\max}. \end{aligned} \tag{5.98}$$

Using results from Lemma 36, Lemma 37 we have $\mathbb{E}\left[r_{(i,k)}(T)\right]$ for competitive and non-competitive base arms. Moreover from Lemma 34, we have the bound on $\mathbb{E}[n_{U_{(i,k)}<\mu_{(i,k)}}(T)]$. Putting these together gives us the desired regret upper bounds for our proposed algorithm.

Chapter 6

Future open directions

This thesis focuses on two key variants of the classical multi-armed bandit framework, namely, structured and correlated multi-armed bandits. We first study the structured bandit framework where the mean rewards corresponding to different actions are a known function of a hidden parameter θ^*, thereby inducing a structure in the mean reward of different arms. While in this setup, the mean rewards of different arms are related to one another, the reward realizations are not necessarily correlated. This motivated us to design novel correlated multi-armed bandit framework which explicitly captures the correlation in reward corresponding to different arms. For both these frameworks, we design algorithms that extend classical bandit algorithms to the structured and correlated bandit settings. Subsequently, we show the utility of our proposed algorithms in the context of recommendation systems. We further demonstrate the applicability of our proposed novel correlated bandit framework to solve online resource allocation problems which frequently arise in tasks such as power allocation in wireless systems, financial optimization and multi-server scheduling.

In this chapter, we discuss potential extensions based on the work presented in this thesis. We first look at potential application areas to which our correlated bandit framework and algorithms could be applied. Next, we discuss challenges to develop a correlated bandit framework in which the correlations are not known a priori but learned during the sequential decision making process.

6.1 Further applications of correlated multi-armed bandits

Hyperparameter optimization and bandits Recently, there has been focus in trying to automate the process of hyperparameter tuning for Machine Learning problems. In this setup, the different hyperparameters (such as learning rate of the algorithm, regularization parameter) can be viewed as the arms in the bandit problem and the validation loss corresponding to an hyperparameter could be viewed as the reward for an arm. The work in [99] formulates this as a best-arm identification problem, where each hyperparameter is trained partially and then the half of the worst-performing hyperparameters are removed and the remaining half are trained partially again. This approach was inspired from the successive halving [72] algorithm from bandit literature and led to the design of hyperband [34]. It will be interesting to explore the use of other bandit algorithms in this setting and investigate if we can model the correlations in performance due to different hyperparameters. For instance, performance of closer hyperparameters (in their L2 distance) is likely to be similar. This problem can also be looked in the context of federated learning, where different local devices need to tune their hyperparameters simultaneously. In this case, similar devices are likely to have similar hyperparameter settings, which is where structure or correlation in the bandit framework could prove to be useful.

Best-algorithm detection Aside from hyperparameter optimization, another aspect of automated machine learning is to find the best-performing algorithm in an application setting from a set of candidate algorithms. For example, for the product recommendation, a company may have several different recommendation engines with them. For each incoming user, they may choose to decide which engine to use to recommend product to the incoming user. This can now be viewed as a Multi-Armed bandit problem where different ML algorithms correspond to the arms in the bandit problem and user-action may generate reward. A key aspect of such applications is that the customer behavior, with regards to preference for different products, keeps changing with time. As a result, the prior data could be useful to learn about the correlations in performance for different ML algorithms, which could then be useful to decide the best algorithm to be used in an upcoming time period. While we mention product recommendation, this could be done for a wide variety of tasks such as workload optimization on a cluster, portfolio optimization etc. By making use of one of the open source machine learning data-drift datasets, one can obtain necessary data and subsequently make use of the correlated bandit framework to improve existing performance of these systems.

6.2 Dealing with non-stationary reward distributions

In several application settings, the reward distribution is non-stationary. For instance, the popularity of a particular ad may decay over time. While this non-stationarity has been studied in the context of classical multi-armed bandits [100], it is yet to be looked at for structured and correlated multi-armed bandits. An initial approach would be to design sliding-window based C-UCB/C-TS algorithms, which evaluate the upper confidence bound (and equivalently posterior distribution) using just the samples obtained between $t - \tau$ and t, with τ being the hyperparameter denoting the length of the sliding window. One could also extend the discounted UCB algorithm in [100] by giving less weight to old samples in the construction of upper confidence bound index for the C-UCB algorithm. It will be interesting to implement and evaluate these ideas on real-world datasets that exhibit this non-stationarity in rewards.

6.3 Learning correlations on the go

Throughout our work, we have assumed that the correlation information may be know either from domain knowledge or through previously performed experiments. An interesting, but challenging, problem would be to learn the correlations in the sequential decision making process itself. In order to do so, it is necessary that multiple arms are selected in each round. While this is not possible in the classical MAB framework, this may be feasible under the combinatorial bandit framework [85], where in each round the user is allowed to select multiple arms (see Chapter 5 for more details). In this particular setting, it may be possible to obtain *joint* samples of two arms ℓ and k. Suppose, there are $n_{\ell,k}(t)$ such samples till round t. These $n_{\ell,k}(t)$ samples could be used to construct an empirical upper bound on $\mathbb{E}\left[R_k|R_\ell = r\right]$, which is the pseudo-reward $s_{k,\ell}(r)$. This can be done by first constructing a high probability upper bound u on $R_k|R_\ell = r$ w.p. $1-\eta$, which can then be used to construct pseudo-reward $s_{k,\ell}(r) = (1-\eta) * u + \eta * M$, with M denoting the maximum possible reward. These empirically constructed pseudo-rewards can then be used to implement our correlated UCB/TS algorithms. As these algorithms are using more information than typical combinatorial bandit algorithms [85], one can expect to see some empirical performance improvements. It will be interesting to conduct this experiment in the context of recommendation systems.

This thesis shows how to model the structure/correlation in the rewards across different arms and discusses several approaches to use the knowledge of the structure and correlation to design sample efficient multi-armed bandit algorithms. This is an important step towards modeling the real-world sequential decision making problems. We believe there is a lot of potential to build upon structured and correlated bandit frameworks to capture even more intricacies that arise in modern-day applications. As the use of multi-armed bandits in real-world application continues to grow, we have an exciting opportunity to implement structured and correlated bandits in a wide variety of application settings, which have the potential to drastically improve the performance metrics.

Bibliography

[1] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985. 1, 11, 13, 16, 26, 38, 68, 80, 136

[2] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory*, 2012, pp. 39–1. 1, 17

[3] A. Garivier and O. Cappé, "The kl-ucb algorithm for bounded stochastic bandits and beyond," in *Proceedings of the 24th annual Conference On Learning Theory*, 2011, pp. 359–376. 1

[4] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 30, no. 2, p. 199, 2015. 1, 96

[5] D. Agarwal, B.-C. Chen, and P. Elango, "Explore/exploit schemes for web content optimization," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE, 2009, pp. 1–10. 1

[6] C. Tekin and E. Turgay, "Multi-objective contextual multi-armed bandit problem with a dominant objective," *arXiv preprint arXiv:1708.05655*, 2017. 1

[7] J. Nino-Mora, *Stochastic scheduling. In Encyclopedia of Optimization*, 2nd ed. New York: Springer, 2009, pp. 3818–3824. 1

[8] S. Krishnasamy, R. Sen, R. Johari, and S. Shakkottai, "Regret of queueing bandits," *CoRR*, vol. abs/1604.06377, 2016. [Online]. Available: http://arxiv.org/abs/1604.06377 1

[9] G. Joshi, "Efficient Redundancy Techniques to Reduce Delay in Cloud Systems," Ph.D. dissertation, Massachusetts Institute of Technology, Jun. 2016. 1

[10] J. White, *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.", 2012. 1, 96

[11] L. Zhou, "A survey on contextual multi-armed bandits," *CoRR*, vol. abs/1508.03326, 2015. [Online]. Available: http://arxiv.org/abs/1508.03326 2, 52

[12] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2014, pp. 1638–1646. 2, 52

[13] T. Lattimore and R. Munos, "Bounded regret for finite-armed structured bandits," in *Advances in Neural Information Processing Systems*, 2014, pp. 550–558. 3, 10, 12, 14, 17, 18, 24, 34, 52, 59

[14] R. Combes, S. Magureanu, and A. Proutière, "Minimal exploration in structured stochastic bandits," in *NIPS*, 2017. 3, 10, 14, 26, 27, 34, 37, 38, 52, 59, 93

[15] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014. 3, 14, 15, 29, 34

[16] Z. Wang, R. Zhou, and C. Shen, "Regional multi-armed bandits," in *AISTATS*, 2018. 3, 8, 9, 10, 13, 37, 59, 106

[17] O. Atan, C. Tekin, and M. van der Schaar, "Global multi-armed bandits with Hölder continuity," in *AISTATS*, 2015. 3, 8, 9, 10, 12, 13, 37, 59, 106

[18] A. J. Mersereau, P. Rusmevichientong, and J. N. Tsitsiklis, "A structured multi-armed bandit problem and the greedy policy," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2787–2802, Dec 2009. 3, 10, 13, 37

[19] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, 4, Article 19, 2015. 3, 6, 32, 73, 117

[20] M. Wan and J. McAuley, "Item recommendation on monotonic behavior chains," in *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018, pp. 86–94. 6, 75, 118

[21] J. Zuo and C. Joe-Wong, "Combinatorial multi-armed bandits for resource allocation," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2021, pp. 1–4. 7, 137, 138, 142, 143, 146, 152, 156

[22] C. Shen, R. Zhou, C. Tekin, and M. van der Schaar, "Generalized global bandit and its application in cellular coverage optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 218–232, 2018. 8, 9, 106

[23] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 661–670. 9

[24] R. Sen, K. Shanmugam, and S. Shakkottai, “Contextual bandits with stochastic experts,” in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 84, 2018, pp. 852–861. [Online]. Available: http://proceedings.mlr.press/v84/sen18a.html 9

[25] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, “Parametric bandits: The generalized linear case,” in *Advances in Neural Information Processing Systems*, 2010, pp. 586–594. 10, 12, 14, 37, 59

[26] T. Lattimore and C. Szepesvari, “The end of optimism? an asymptotic analysis of finite-armed linear bandits,” *arXiv preprint arXiv:1610.04491*, 2016. 10, 13, 14, 37

[27] T. L. Graves and T. L. Lai, “Asymptotically efficient adaptive choice of control laws incontrolled markov chains,” *SIAM journal on control and optimization*, vol. 35, no. 3, pp. 715–743, 1997. 10, 93

[28] W. R. Thompson, “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples,” *Biometrika*, vol. 25, no. 3-4, pp. 285–294, Dec. 1933. 10, 16, 17

[29] S. Agrawal and N. Goyal, “Thompson sampling for contextual bandits with linear payoffs,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043073 10

[30] S. Bubeck, N. Cesa-Bianchi *et al.*, “Regret analysis of stochastic and nonstochastic multi-armed bandit problems,” *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012. 12, 46, 136

[31] K. Jamieson and R. Nowak, “Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting,” in *Proceedings on the Annual Conference on Information Sciences and Systems (CISS)*, March 2014, pp. 1–6. 12, 15, 17, 78, 95, 99, 103, 104, 105, 106, 115, 128, 129

[32] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002. 13, 16, 21, 39, 45, 46, 60, 79, 80, 109, 121, 157, 158, 165

[33] A. Gopalan, S. Mannor, and Y. Mansour, “Thompson sampling for complex online problems,” in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 32. Bejing, China: PMLR, 22–24 Jun 2014, pp. 100–108. [Online]. Available: http://proceedings.mlr.press/v32/gopalan14.html 14, 15

[34] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A novel bandit-based approach to hyperparameter optimization,” *Journal of Machine Learning Research*, vol. 18, no. 1,

pp. 6765–6816, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3122009.3242042 15, 96, 167

[35] E. Tánczos, R. Nowak, and B. Mankoff, "A kl-lucb algorithm for large-scale crowdsourcing," in *Advances in Neural Information Processing Systems*, 2017, pp. 5894–5903. 15, 95, 96, 104, 105, 118

[36] K. G. Jamieson, L. Jain, C. Fernandez, N. J. Glattard, and R. Nowak, "Next: A system for real-world development, evaluation, and application of active learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 2656–2664. 15

[37] R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright, "Active ranking from pairwise comparisons and the futility of parametric assumptions," *arXiv*, vol. abs/1606.08842, 2016. [Online]. Available: http://arxiv.org/abs/1606.08842 15

[38] S. Bubeck, R. Munos, and G. Stoltz, "Pure exploration in multi-armed bandits problems," in *Algorithmic Learning Theory*, R. Gavaldà, G. Lugosi, T. Zeugmann, and S. Zilles, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 23–37. 15, 17, 95

[39] J.-Y. Audibert, S. Bubeck, and R. Munosr, "Best arm identification in multi-armed bandits," in *Proceedings of the annual Conference On Learning Theory (COLT)*, 2010, pp. 359–376. 15, 17

[40] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "lil'ucb: An optimal exploration algorithm for multi-armed bandits," in *Conference on Learning Theory*, 2014, pp. 423–439. 15, 17, 78, 95, 102, 103, 105

[41] E. Kaufmann, O. Cappé, and A. Garivier, "On the complexity of best-arm identification in multi-armed bandit models," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1–42, 2016. 15

[42] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, no. Jun, pp. 623–648, 2004. 15

[43] A. Garivier and E. Kaufmann, "Optimal best arm identification with fixed confidence," in *Annual Conference on Learning Theory (COLT)*, ser. Proceedings of Machine Learning Research, vol. 49. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 998–1027. [Online]. Available: http://proceedings.mlr.press/v49/garivier16a.html 15, 104

[44] V. Gabillon, M. Ghavamzadeh, and A. Lazaric, "Best arm identification: A unified approach to fixed budget and fixed confidence," in *Advances in Neural Information Processing Systems*, 2012, pp. 3212–3220. 15

[45] R. Sen, K. Shanmugam, A. G. Dimakis, and S. Shakkottai, "Identifying best interventions through online importance sampling," *stat*, vol. 1050, p. 9, 2017. 15

[46] M. Soare, A. Lazaric, and R. Munos, "Best-arm identification in linear bandits," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 828–836. [Online]. Available: http://papers.nips.cc/paper/5460-best-arm-identification-in-linear-bandits.pdf 15, 106

[47] R. Huang, M. M. Ajallooeian, C. Szepesvári, and M. Müller, "Structured best arm identification with fixed confidence," in *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, ser. Proceedings of Machine Learning Research, vol. 76, Kyoto University, Kyoto, Japan, Oct. 2017, pp. 593–616. [Online]. Available: http://proceedings.mlr.press/v76/huang17a.html 15, 106

[48] C. Tao, S. Blanco, and Y. Zhou, "Best arm identification in linear bandits with linear dimension dependency," in *Proceedings of the International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80, Jul. 2018, pp. 4877–4886. [Online]. Available: http://proceedings.mlr.press/v80/tao18a.html 15, 106

[49] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 2249–2257. 17

[50] D. Russo, B. V. Roy, A. Kazerouni, and I. Osband, "A tutorial on thompson sampling," *CoRR*, vol. abs/1707.02038, 2017. [Online]. Available: http://arxiv.org/abs/1707.02038 17

[51] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, 2011, pp. 2312–2320. 37, 52, 59

[52] S. Agrawal and N. Goyal, "Further optimal regret bounds for thompson sampling," in *Artificial Intelligence and Statistics*, 2013, pp. 99–107. 46, 47, 49, 60, 88, 89, 90, 91, 92

[53] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proceedings on the Conference on Learning Theory (COLT)*, 2008. 52

[54] S. Gupta, G. Joshi, and O. Yağan, "Correlated multi-armed bandits with a latent random source," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3572–3576. 58, 138

[55] S. Gupta, S. Chaudhari, S. Mukherjee, G. Joshi, and O. Yağan, "A unified approach to translate classical bandit algorithms to the structured bandit setting," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 3, pp. 840–853, 2020. 59

[56] S. Magureanu, R. Combes, and A. Proutiere, "Lipschitz bandits: Regret lower bounds and optimal algorithms," *arXiv preprint arXiv:1405.4758*, 2014. 59

[57] S. Pandey, D. Chakrabarti, and D. Agarwal, "Multi-armed bandit problems with dependent arms," in *Proceedings of the International Conference on Machine Learning*, 2007, pp. 721–728. 59

[58] R. Combes and A. Proutiere, "Unimodal bandits: Regret lower bounds and optimal algorithms," in *International Conference on Machine Learning*, 2014, pp. 521–529. 60

[59] C. Trinh, E. Kaufmann, C. Vernade, and R. Combes, "Solving bernoulli rank-one bandits with unimodal thompson sampling," in *Algorithmic Learning Theory*. PMLR, 2020, pp. 862–889. 60

[60] M. A. Qureshi and C. Tekin, "Fast learning for dynamic resource allocation in ai-enabled radio networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 95–110, 2019. 60

[61] ——, "Online bayesian learning for rate selection in millimeter wave cognitive radio networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1449–1458. 60

[62] H. Gupta, A. Eryilmaz, and R. Srikant, "Link rate selection using constrained thompson sampling," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 739–747. 60

[63] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi, "Bandits with heavy tail," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7711–7717, 2013. 78

[64] T. Lattimore, "Instance dependent lower bounds," http://banditalgs.com/2016/09/30/instance-dependent-lower-bounds/. 80, 81

[65] A. B. Tsybakov, *Introduction to Nonparametric Estimation*, 1st ed. Springer Publishing Company, Incorporated, 2008. 81

[66] J. Bretagnolle and C. Huber, "Estimation des densitiés: risque minimax. z. für wahrscheinlichkeitstheorie und verw," *Geb.*, vol. 47, pp. 199–137, 1979. 81

[67] B. Van Parys and N. Golrezaei, "Optimal learning for structured bandits," *Available at SSRN 3651397*, 2020. 93, 106

[68] E. Kaufmann and S. Kalyanakrishnan, "Information complexity in bandit subset selection," in *Conference on Learning Theory*, 2013, pp. 228–251. 95, 102, 105

[69] M. Simchowitz, K. Jamieson, and B. Recht, "The simulator: Understanding adaptive sampling in the moderate-confidence regime," *arXiv preprint arXiv:1702.05186*, 2017. 95, 104, 118, 130

[70] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "Pac subset selection in stochastic multi-armed bandits." in *ICML*, vol. 12, 2012, pp. 655–662. 95, 102, 104, 105, 129

[71] R. E. Bechhofer, "A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs," *Biometrics*, vol. 14, no. 3, pp. 408–429, 1958. 95, 102

[72] E. Even-Dar, S. Mannor, and Y. Mansour, "Pac bounds for multi-armed bandit and markov decision processes," in *International Conference on Computational Learning Theory*. Springer, 2002, pp. 255–270. 95, 102, 105, 167

[73] S. R. Howard, A. Ramdas, J. McAuliffe, and J. Sekhon, "Time-uniform, nonparametric, nonasymptotic confidence sequences," 2018. 102, 105, 106, 108

[74] E. Paulson *et al.*, "A sequential procedure for selecting the population with the largest mean from k normal populations," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 174–180, 1964. 102

[75] Z. Karnin, T. Koren, and O. Somekh, "Almost optimal exploration in multi-armed bandits," in *International Conference on Machine Learning*, 2013, pp. 1238–1246. 103

[76] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "On finding the largest mean among many," *arXiv preprint arXiv:1306.3917*, 2013. 103, 105

[77] X. Shang, R. Heide, P. Menard, E. Kaufmann, and M. Valko, "Fixed-confidence guarantees for bayesian best-arm identification," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1823–1832. 104

[78] R. Degenne, W. M. Koolen, and P. Ménard, "Non-asymptotic pure exploration by solving games," *arXiv preprint arXiv:1906.10431*, 2019. 104

[79] T. Kocák and A. Garivier, "Best arm identification in spectral bandits," *arXiv preprint arXiv:2005.09841*, 2020. 106, 107

[80] D. Julian, M. Chiang, D. O'Neill, and S. Boyd, "Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2002, pp. 477–486. 134, 138

[81] A. Lozano, A. M. Tulino, and S. Verdú, "Optimum power allocation for parallel gaussian channels with arbitrary input distributions," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 3033–3051, 2006. 134, 138

[82] W. W. Chu, "Optimal file allocation in a multiple computer system," *IEEE Transactions on Computers*, vol. 100, no. 10, pp. 885–889, 1969. 134

[83] K. S. Reddy, S. Moharir, and N. Karamchandani, "Resource pooling in large-scale content delivery systems," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1617–1630, 2019. 134

[84] D. N. Kleinmuntz, *20 Resource Allocation Decisions*. Citeseer, 2007. 134, 138

[85] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International conference on machine learning*. PMLR, 2013, pp. 151–159. 136, 140, 147, 148, 156, 168

[86] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1785–1798, 2013. 138

[87] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, "Near optimal online algorithms and fast approximation algorithms for resource allocation problems," *Journal of the ACM (JACM)*, vol. 66, no. 1, pp. 1–41, 2019. 138

[88] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, "A unified approach to approximating resource allocation and scheduling," *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1069–1090, 2001. 138

[89] T. Lattimore, K. Crammer, and C. Szepesvári, "Linear multi-resource allocation with semi-bandit feedback." in *NIPS*, 2015, pp. 964–972. 138

[90] A. Verma, M. K. Hanawal, A. Rajkumar, and R. Sankaran, "Censored semi-bandits: A framework for resource allocation with censored feedback." in *NeurIPS*, 2019. 138

[91] X. Fontaine, S. Mannor, and V. Perchet, "An adaptive stochastic optimization algorithm for resource allocation," in *ALT*. PMLR, 2020, pp. 319–363. 138

[92] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 1466–1478, 2012. 138

[93] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1746–1778, 2016. 138

[94] S. Gupta, S. Chaudhari, G. Joshi, and O. Yağan, "Multi-armed bandits with correlated arms," *IEEE Transactions on Information Theory*, 2021. 138

[95] Y. Gai and B. Krishnamachari, "Online learning algorithms for stochastic water-filling," in *2012 Information Theory and Applications Workshop*, 2012, pp. 352–356. 152

[96] A. Narasimhamurthy, M. Banavar, and C. Tepedelenliouglu, *OFDM Systems for Wireless Communications*. Morgan & Claypool, 2010. 152

[97] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, 1970, pp. 281–285. 153

[98] L. Pajevic, G. Karlsson, and V. Fodor, "CRAWDAD dataset kth/campus (v. 2019-07-01)," Downloaded from https://crawdad.org/kth/campus/20190701/eduroam, Jul. 2019, traceset: eduroam. 153

[99] K. Jamieson and A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 240–248. 167

[100] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *International Conference on Algorithmic Learning Theory*. Springer, 2011, pp. 174–188. 167, 168