UNIVERSITY OF CALIFORNIA
RIVERSIDE

Contextual Bandits in Imperfect Environments: Analysis and Applications

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Luting Yang

March 2021

Dissertation Committee:

Dr. Shaolei Ren, Chairperson
Dr. Daniel Wong
Dr. Hyoseung Kim

The Dissertation of Luting Yang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I would like to express my deepest and most sincere gratitude to my Ph.D. advisor, Professor Shaolei Ren, for his guidance, support, encouragement, and help during my Ph.D. study. I have acquired so many academic skills from him on how to find potential research directions, formulate and solve problems, write technical and tutorial articles, and give academic presentations. All the achievements in my graduate studies could not happen without his countless help. It is my honor to be his student.

I am also fortunate to have Professor Daniel Wong and Professor Hyoseung Kim as my committee members. I have learned a lot in discussions with and lectures from them. Their informative suggestions really help me deepen and sharpen my academic thinking. I would like to extend my gratitude to them. During my Ph.D. study, I did one impressive internships at the Walmart Lab in Sunnyvale, California. I would like to thank my mentor Dr. Yichuan Niu for his insightful suggestions in the research project as well as his help in my settlement in a new environment. I have been extremely fortunate to work with a wonderful group of colleagues and lab mates at UCR: Jianyi Yang, Bingqian Lu, Zhihui Shao, and Fangfang Yang who have made my time as a graduate student fun and memorable. In particular, I would like to thank Professor Mohammad Atiqul Islam for providing me with initial guidance when I started working on acoustic side channel problems. Special thanks to Jianyi Yang for his help in nearly every aspect of research at my early stages as a Ph.D. student. I spent wonderful times with Bingqian Lu, Zhihui Shao, and Fangfang Yang in Riverside. We can discuss anytime at various places: in offices, at restaurants, and on WeChat.

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

Finally, the financial support of my work came from National Science Foundation (NSF), UCR Dean's Distinguished Fellowship, and UCR TAship.

This dissertation is dedicated to my father Wantao Yang, my mother Jing Liu and my wife Yajie Yang. I could not achieve any success whatsoever in my life without their unconditional support, belief, and love

ABSTRACT OF THE DISSERTATION

Contextual Bandits in Imperfect Environments: Analysis and Applications

by

Luting Yang

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, March 2021
Dr. Shaolei Ren, Chairperson

The data explosion and development of artificial intelligence (AI) has fueled the demand for recommendation systems, information retrieval, personalization, among others. Consequently, the need of a solution to optimize these systems "on-the-fly" has also grown rapidly. Contextual bandit is a machine learning framework designed to tackle complex situations in an online manner, where the agent can select actions (i.e., arms) based on available context information. Based the feedback, the agent can learn the relations between context information and rewards for each arm, which further improves arm selection in the future. In practice, however, the learning environment may be far from being perfect. For example, the available context information may not be accurate, the reward feedback may be delayed or even missing, and data may not be centrally available due to user privacy concerns.

In this dissertation, we consider the practical scenario of contextual bandits in an imperfect environment. First, we focus on imperfect context and study learning with probabilistic contexts, where a bundle of contexts are revealed to the agent along with their corresponding probabilities instead of true context. Second, we study reward imperfectness by considering delayed or missing reward feedback. Third, we turn to an adversarial environment and study a novel combinatorial setting with arm removal and submodular utility where some selected arms can be removed adversarially. Finally, we consider a privacy-preserving federated bandit where a group of agents cooperate to solve the bandit problem, while ensuring that their communication remains private. For each of the settings, we propose new learning algorithms, analyze the cumulative regret, and conduct empirical evaluations based on real-world applications.

Contents

List of Figures **xii**

List of Tables **xiii**

1 Introduction **1**

1.0.1 Background and Motivation 1

1.0.2 Research Objectives and Contributions 2

1.0.3 Thesis Organization 6

2 Contextual Bandit with Probabilistic Contexts **8**

2.1 Introduction 8

2.2 Related Work 10

2.3 Problem Formulation 12

2.4 Algorithm 14

2.4.1 Feedback Prediction 14

2.4.2 Multi-Feedback Probabilistic Contextual UCB 17

2.5 Regret Analysis 18

2.5.1 Cumulative Regret Bound 18

2.5.2 Proof Sketch 19

2.5.3 Regret of Arm Selection by Most Probable Context 22

2.6 Simulation Results 23

2.6.1 Application to DNN Model Selection 23

2.6.2 Results 26

3 Contextual Bandit with Delayed Feedback **28**

3.1 Introduction 28

3.2 Related Work 30

3.3 Problem Formulation 31

3.4 Delayed Contextual UCB 33

3.4.1 Reward Estimation 33

3.4.2 Delayed Contextual UCB for Arm Selection 35

3.5 Regret Analysis 36

3.5.1 Finite Maximum Delay 37
3.5.2 Missing Feedback 40
3.6 UCB With Semi-supervised Learning 41
3.6.1 Minimizing Estimation Error by Similar Context 41
3.6.2 Regret Analysis 45
3.7 Experiments and Results 47
3.7.1 Yahoo! Today Module 48
3.7.2 Experimental Setup 48
3.7.3 Experimental Result 49

4 Contextual Combinatorial Bandits with Arm Removal and Submodular Utility 51
4.1 Introduction 51
4.2 Related Work 55
4.3 Problem Formulation 56
4.4 Robust Contextual Combinatorial Bandit with Arm Removal 59
4.4.1 Problem Hardness 59
4.4.2 Algorithm Design 62
4.5 Regret Analysis 64
4.5.1 Regret Definition 65
4.5.2 Analysis 66
4.6 Application to Sniffer Channel Assignment 72
4.6.1 Models 72
4.6.2 Applicability of R2C2-MAB 75
4.7 Numerical Evaluation 76
4.7.1 Settings 76
4.7.2 Stochastic Sniffer Removal 79
4.7.3 Adversarial Sniffer Removal 80

5 Federated Contextual Bandit with Differential Privacy 82
5.1 Introduction 82
5.2 Related Work 85
5.3 Preliminary 86
5.3.1 Federated Bandits 86
5.3.2 Differential Privacy 87
5.4 Problem Formulation 87
5.5 Algorithm Design 89
5.5.1 Non-parametric Contextual Bandits 89
5.5.2 Federated Bandits with Differential Privacy 91
5.6 Regret Analysis 94
5.7 Experiments and Results 102
5.7.1 Contextual COVID-19 Admission Decision 103
5.7.2 Simulation Settings 104
5.7.3 Simulation Results 105

6 Conclusions **108**

Bibliography **111**

A Supplemental Proof **120**

A.0.1 Proof of Lemma 8 120
A.0.2 Proof of Lemma 10 121
A.0.3 Proof of Lemma 11 121
A.0.4 Proof of Lemma 12 124
A.0.5 Proof of Theorem 13 125
A.0.6 Proof of Lemma 17 126
A.0.7 Proof of Proposition 20 130

List of Figures

2.1 Performance comparison. 27

3.1 Cumulative regrets for different algorithms. 47

4.1 Cumulative reward comparison among different algorithms. 77

5.1 Average reward for different privacy budget 107

List of Tables

2.1 Average Energy Consumption 25
2.2 Average Latency 25

5.1 Reward Matrix 104

Chapter 1

Introduction

1.0.1 Background and Motivation

From whom to meet with, to what to have for breakfast in the morning, to the decision to study in college or find a job, human nature is all about choices. From conversational agents to online recommendation to search and advertising, we are already interacting with increasingly sophisticated sequential decision making systems in daily life. Traditionally, sequential decision making has focused on balancing the exploration-exploitation trade-off, or casting the interaction paradigm under reinforcement / bandit learning dichotomy.

This dissertation focuses on the online learning and sequential decision-making problem under unknown environments. The objective in this class of problems is to learn, ”on-the-fly”, the most profitable actions (arms) among a number of selections. The problem is formulated and studied under the classic framework of multiarmed bandits (MAB) in this dissertation. MAB is a crucial online learning problem to discover optimal decisions (a.k.a. arms) based on received feedback signals over time [60]. Meanwhile, several efficient algorithms, such as Lin-UCB [65], contextual Thompson sampling (CTS) [5], EXP3 [9] and their variants, have been developed for different problem settings. Importantly, contextual bandit learning extends the standard MAB setting by allowing the learner/agent to access some side information (i.e., context) about the environment prior to arm selection [68]. For contextual bandit, the context and selected arm jointly determine the distribution of reward received by the agent, and the goal of the agent is to maximize its cumulative reward by gradually identifying the optimal mapping of context information into actions based on the history of context-action-feedback. Applications of contextual bandit have been increasingly expanding, including advertisement [45], personalization [65], adaptive rate and modulation based on channel condition in wireless networks [89] and service placement based on traffic demand in edge computing systems [31]. LinUCB [65] and Thompson sampling [4, 5] are two landmark algorithms for contextual bandits, which have subsequently been extended by many studies for various settings and applications [44, 93, 102, 33, 36, 19, 46, 8].

In this work, we point out several emerging challenges in applications under complex imperfect environment that call for new models and new learning strategies, and develop corresponding solutions with performance guarantees both theoretically and practically.

1.0.2 Research Objectives and Contributions

Despite the abundant theories and algorithms developed in the previous bandit literature, it is still challenging to utilize bandit solution in the real world due to imperfect environment. In this dissertation work, we consider four practical scenario of contextual bandits in an imperfect environment and develop algorithms which has the potential to be implemented in the real world, with performance guarantee, and numerical evaluation.

Probabilistic Contexts

A standard assumption in the literature is that exact contexts are perfectly known prior to arm selection. In the first part, we deviate from this assumption and focus on bandit learning with probabilistic contexts, where a bundle of contexts, instead of only the true one, are revealed to the agent along with their corresponding probabilities at each round (*e.g.* the agent needs to act upon the probabilistic output of a deep neural network classifier).

In order to design an efficient learning algorithm for probabilistic setting, the agent leverages the available probabilistic context information to learn multiple feedback functions that jointly determine its utility. To balance the exploration and exploitation, we consider the joint upper confidence bound for multiple functions when choose arm. Moreover, we consider a general setting where each individual feedback function can be nonlinear with respect to the selected arm and contexts, and apply the kernel method to transfer feedback function in the Reproducing kernel Hilbert Space (RKHS).

We propose a kernelized probabilistic contextual bandit learning algorithm, based on the principle of the maximum likelihood estimator, to learn the optimal arm in reproducing kernel Hilbert space for each context bundle. Moreover, we theoretically establish an upper bound on the cumulative regret, which shows that the regret growth of our algorithm is sublinear with time. Our result from contextual DNN model selection experiment validates the sub-linearity of regret bound and superiority of our algorithm compared to state of the art when probabilistic contexts are provided.

Delayed or Missing Reward

Contextual bandits model a wide range of applications such as online recommendation systems, where the agent can select actions (i.e., arms) based on available context information. In practice, however, the reward feedback provided to the agent for learning is often delayed or even missing. For example, in the customer rating system. it is common that customer only provides rating feedback several hours/days later or even leaves no feedback.

In this part, we consider this practical setting and propose an algorithm based on delayed contextual upper confidence bound (UCB) to balance exploitation and exploration. Our algorithm updates its reward function learning whenever new reward feedback is received, and selects arms based on UCB. Importantly, we derive an upper bound of the cumulative regret for our algorithm that grows sub-linearly with time.

Further, with the concept that similar context yields similar reward by the same action, we advance delayed UCB by using semi-supervised learning to produce fictitious estimates for rewards that are delayed or missed and have not been revealed to the agent. Therefore, by combining semi-supervised learning with online contextual bandit learning, we propose a novel extension and design another algorithm to minimize the estimation error. Finally, we apply our algorithms to the problem of online context-aware articles recommendation to viewers. Our result validates the regret analysis and demonstrates that the fictitious estimates for delayed or missing rewards can be useful for decreasing the regret.

Combinatorial Arm with Removal

Contextual combinatorial bandit is an important online learning problem, with applications to many networked systems, such as dynamic channel assignment in wireless networks and influence maximization in social networks. In practice (especially in tactical or adversarial environments), the selected arms may be deliberately or accidentally nullified, contributing zero to the learning agent's total utility and warranting robust arm selection strategies to account for the worst case.

In the third part, we study a novel contextual combinatorial bandit setting with arm removal and submodular utility: the agent can select multiple arms (subject to a cardinality constraint), but some selected arms can be removed and the overall utility is jointly determined through a monotone submodular utility function in terms of the remaining selected arms. Even with perfect knowledge regarding the feedback signals for each arm and context, robust submodular maximization with arm removal is a challenging NP-hard problem, let alone in our online learning setting.

We propose a novel online bandit algorithm, called R2C2-MAB, to robustly select arms to maximize the worst-case submodular utility while balancing exploration and exploitation. Importantly, we prove that R2C2-MAB achieves a sublinear regret in time compared to an efficient baseline algorithm that has a provable constant approximation ratio. To empirically evaluate R2C2-MAB, we consider the wireless sniffer channel assignment problem as a concrete example and run simulations. Under both stochastic and adversarial arm removals, our results show that R2C2-MAB achieves a total reward close to that of the baseline, while outperforming other existing bandit algorithms that either do not exploit the submodularity structure of the utility function or neglect the presence of arm removal.

Federated Bandits

Standard bandit learning approaches require centralized platform that the contextual information and reward feedback on one machine or server. However, the rapid proliferation of decentralized learning systems mandates the need for cooperative bandit learning. However, cooperative setting brings new challenge in the data privacy, security, heterogeneity. Federated learning enables multiple agents to collaboratively learn the environment without sharing data, thus allowing to address critical privacy issues.

In this last part, we study the problem, which combines contextual bandits and federated learning with differential privacy: we consider a collection of agents cooperating to solve a common contextual bandit, while ensuring that their communication remains private. For this problem, we devise centralized federated bandit algorithm, a multiagent private algorithm for both centralized federated setting. We provide a rigorous technical analysis of its utility in terms of regret, improving several results in cooperative bandit learning, and provide rigorous privacy guarantees as well. Our algorithms provide competitive performance in terms of empirical benchmark performance in various multi-agent settings.

1.0.3 Thesis Organization

The remainder of the dissertation is organized as follows: we give the detailed presentations and analysis of contextual bandit with imperfect environment from Chapter 2-5. In Chapter 2, we present the probabilistic framework and the proposed kernelized probabilistic contextual bandit learning algorithm to learn the optimal arm in reproducing kernel Hilbert space for each context bundle. In Chapter 3, we detail the problem formulation and the technical methods for reward feedback provided to the agent is often delayed or even missing. In Chapter 4. we discuss the novel contextual combinatorial bandit setting with arm removal and submodular utility In Chapter 5, we develop federated contextual bandit with differential privacy, given heterogeneous reward information from multiple agents. Finally, Chapter 6 concludes this dissertation and points out future research directions.

Chapter 2

Contextual Bandit with Probabilistic Contexts

2.1 Introduction

Contextual bandits have found success in many applications, including online recommendation [76], commercial advertising [98] and medical experiment design [105]. Subsequently, efficient learning algorithms like Lin-UCB [65], EXP4 [10] and their variations have drawn great attention. Nonetheless, most of the prior studies assume that the context information acquired by the agent before arm selection is perfect. While this assumption facilitates performance analysis of the proposed algorithms, it may fail in certain practical scenarios, where there is randomness and uncertainty about the context information.

To alleviate the uncertainty and randomness from environment, one can apply classification techniques, such as support vector machine (SVM) [85] and deep neural networks (DNN) [106], which yield a probability distribution over possible candidate category of contexts. For example, a recommendation system commonly recommends personalized items given user features (i.e., contexts) predicted by a neural network classifier. Thus, all the possible candidate contexts together form a context bundle with a probability distribution of different contexts, and the exact context is included in the bundle (possibly not having the greatest probability) but unknown to the agent. In this chapter, we also use “context” and “context candidate” exchangeably.

In addition to the lack of exact context information, another practical consideration for contextual bandit learning is that the agent can receive multiple feedbacks instead of a single one. In this case, the goal of the agent is to maximize a (possibly time-varying) utility function jointly determined by multiple feedbacks rather than any of the individual feedback. For example, when selecting an app for a mobile device, both energy and latency can be measured and reported to the learner/agent, and these metrics jointly affect the performance of the selected app.

Motivated by the aforementioned practical considerations, the focus of this work is to study a novel contextual bandit setting where the agent can only access to a probabilistic context bundle for arm selection and its goal is to maximize a time-varying utility function jointly determined by the multiple feedback signals received at the end of each round. In order to design an efficient learning algorithm, the key is how the agent leverages the available probabilistic context information to learn multiple feedback functions that jointly determine its utility. To study this problem, we consider a general setting where each individual feedback function can be nonlinear with respect to the selected arm and contexts, and apply the kernel method to transfer feedback function in the Reproducing kernel Hilbert Space (RKHS). We design a new algorithm by extending upper confidence bound (UCB) techniques to account for the probabilistic context information, using the expectation of reward over the probabilistic context distribution. For each feedback, we learn its relation with the selected arm given a probabilistic context bundle. Then, an arm is selected based on an estimated reward function in terms of all the estimated feedback values. Importantly, we prove that our algorithm achieves a sub-linear regret upper bound $\mathcal{O}(\sqrt{T\log(T)})$ when compared to an oracle that knows the optimal arm given any probabilistic context bundle. We also consider an alternative algorithm that simply uses the most probable context from the given probabilistic context bundle, and show its linear regret bound.

We apply our learning algorithm to the problem of deep neural network (DNN) model recommendation for edge inference on mobile devices. Our experiments show that our proposed algorithm outperforms the alternative solution that selects arms based on the most probable context. More importantly, our algorithm yields a sub-linear regret with respect to the oracle, demonstrating the effectiveness of our algorithm and validating the regret analysis.

2.2 Related Work

Contextual bandits have been studied in various settings due to their wide applications [62]. The study [65] proposes Lin-UCB algorithm, assuming a linear relationship between its context and expected reward, which applies ridge regression for estimated feedback. As for the nonlinear contextual bandits, [103, 36] both propose kernelized contextual bandit as a nonlinear version of Lin-UCB by finding linear members in RHKS. [7] utilizes neural networks to predict the rewards given the context and proposed a multi-expert approach to decide the parameters of networks. [120] provides a formal proof for neural network-based contextual UCB. [13] introduces the concept of contextual bandits with budget constraints, and proposes a resourceful contextual bandits algorithm that provably achieves $\mathcal{O}(\sqrt{T})$ regret bound. Another variant of contextual bandit considers that not all contextual information is accessible [20]. Similarly, [108] assumes the existence of hidden features and arm vectors from context together and proposes hLin-UCB algorithm.

Among the studies on probabilistic contextual bandits, a relevant one [57] considers that the agent only knows the probability distribution of context. The major difference is that we consider multiple feedbacks and a time-varying utility function. Another one is [118], which studies contextual bandit with perturbation noise on observed context. The authors assume a linear reward function with a single feedback, and propose an algorithm called NLin-Rel that achieves $\mathcal{O}(T^{\frac{7}{8}})$ regret bound under the assumption of identical noise. Different from this work, we consider non-linear reward function with probabilistic contexts.

As for multiple feedbacks, [72] proposes an algorithm based on Pareto optimality to solve a multi-objective problem under contextual settings, resulting in a regret bound that increases sub-linearly under the assumption of a linear feedback function. Another relevant work is [110], which considers a multi-objective online contextual ranking system with the assumption that some parameters in both feedback and reward are unknown. By setting linear feedback and logistic utility, the proposed UCB-based algorithm is shown to significantly increase the click-through rate. In contrast, we use a more general time-varying utility function to combine multiple feedback signals and consider probabilistic contexts.

2.3 Problem Formulation

We consider the problem of contextual bandit learning with probabilistic contexts and multiple feedbacks, and provide the mathematical formulation of this problem in this section.

In a standard bandit setting, deterministic context $x_t \in \mathcal{R}^M$ is available to the agent at each round. In many real cases, however, the agent only has the knowledge of a bundle of context candidates $\mathcal{X} = \left\{x^1, \cdots, x^N\right\}$ and the true context x_t is in the context bundle. At each round t, with some prior knowledge, the agent can get a collection of probabilities for context candidates, i.e. $Pr_t(\mathcal{X}) = \left\{P_t(x^1), \cdots, P_t(x^N)\right\}$ and $\sum_{i=1}^N P_t(x^i) = 1$. This can be done by using, for example, a well-trained DNN classifier and extracting the softmax layer output of the classifier. The extracted probabilities define a probability space over the context bundle $\mathcal{X}$. Now, the context at round t is a random variable X_t in the probability space with the probability measure $Pr\left(X_t = x_t^i\right) = P_t\left(x_t^i\right)$. Note that, with a notational change, our model can also be extended to continuous context with a probability density function.

Additionally, we consider a more practical case where the agent receives multiple feedbacks. The jth feedback $(j = 1, \cdots, J)$ with respect to action a and

the random context X_t can be expressed as

$$f_{a,t}^j = g_a^j(X_t) + \epsilon^j \tag{2.1}$$

where $g_a^j(\cdot)$ is a deterministic feedback function which can be linear or nonlinear, and ϵ^j is zero-mean Gaussian noise and ϵ^i and ϵ^j are mutually independent for $i \neq j$. Assume that for $j = 1, \cdots, J$, a kernel function $k^j : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$ can be found to represent $g_a^j(\cdot)$ in a Reproducing Kernel Hilbert space (RHKS) $\mathcal{F}^j$. In other words, the kernel function k^j corresponds to a feature map $\phi^j : \mathbb{R}^n \to \mathcal{F}^j$ which satisfies $k^j(x, x') = \phi^j(x)^\top \phi^j(x'), \forall x, x' \in \mathbb{R}^M$, and $g_a^j(x) = \phi^j(x)^\top \theta_a^j$.

The agent's reward is evaluated by a utility function $U_t : \mathcal{R}^J \to \mathcal{R}$, which may change over time and is known to the agent. Assuming that the Lipschitz constant of the utility function U_t is L_t and $L = \max_t L_t$, we have

$$|U_t(\mathbf{f}_1 - \mathbf{f}_2)| \leq L \left\| \mathbf{f}_1 - \mathbf{f}_2 \right\|. \tag{2.2}$$

If an action a is selected, then a reward $U_t(\mathbf{f}_{a,t})$ is obtained by the agent where $\mathbf{f}_{a,t} = \left[f_{a,t}^1, \cdots, f_{a,t}^J\right]$ is the feedback vector. We seek to maximize the expected reward over both the probabilistic context space and the noise space, which is denoted as $\mathbb{E}[U_t(\mathbf{f}_{a,t})]$, by selecting actions. For the convenience of analysis, we further assume that $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ where the expectation $\mathbb{E}_\epsilon[\cdot]$ is taken over the noise space. Example utility functions include a linear form ($U_t(\mathbf{f}) = \mathbf{u}_t^\top \mathbf{f}$) or a multiplication form ($U_t(\mathbf{f}) = \prod_{j=1}^J f^j$), which are also common functions used in multi-objective bandits [87, 113].

The best action at round t is defined as the action with highest expected reward, i.e.

$$a_t^* = \arg\max_a \mathbb{E}[U_t(\mathbf{f}_{a,t})] \tag{2.3}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over both noise space and context space. This best action oracle is reasonable and common for the cases with context uncertainty, and also considered as a benchmark in [57, 118]. With this oracle, the expected instant regret reg_t at every round can be expressed as

$$reg_t = \mathbb{E}\left[U_t(\mathbf{f}_{a_t^*,t}) - U_t(\mathbf{f}_{a_t,t})\right] \tag{2.4}$$

where the expectation $\mathbb{E}\left[\cdot\right]$ is taken over both noise space and context space. The algorithm needs to be designed to find an arm selection policy based on the history to minimize the cumulative regret $R_T = \sum_{t=1}^{T} reg_t$.

2.4 Algorithm

In this section, we first introduce the feedback prediction algorithm and then, given the predicted feedbacks and confidence widths, design a UCB-based algorithm with probabilistic contextual information.

2.4.1 Feedback Prediction

In order to select an action, the algorithm should be able to predict the feedbacks corresponding to each action, which then determines the resulting reward. Note that we cannot simply treat the overall utility function as a single feedback signal and directly predict it given incoming contextual information as in prior studies [36, 57], because the utility function in terms of multiple feedbacks is changing over time in our setting. To accomplish feedback prediction, we can estimate the parameter θ_a^j in feedback functions by kernel-based empirical risk minimization based on the history $\mathcal{H}_{a,t}^j = \left\{\left(\mathcal{X}, Pr_\tau(\mathcal{X}), f_{a,\tau}^j\right), \tau = 1, \cdots, t\right\}, j = 1, \cdots, J$. Denote the set of rounds when arm a is selected before round t as $\mathcal{T}_{a,t} = \left\{\tau_a^1, \tau_a^2, \cdots, \tau_a^{n_{a,t}}\right\}$. The kernel based empirical risk minimization is to solve the following problem

$$\hat{\theta}_a^j = \arg\min_{\theta_a^j} \frac{1}{n_{a,t}} \sum_{\tau \in \mathcal{T}_{a,t}} (\mathbb{E}[\phi^j(X_t)]^\top \theta_a^j - f_{a,\tau}^j)^2 + \lambda \|\theta_a^j\|^2 \tag{2.5}$$

where $\lambda \geq 0$ is a hyper-parameter.

Denote $\mathbf{\Phi}_a^j = \Big[\mathbb{E}\big[\phi^j(X_{\tau_a^1})\big], \cdots, \mathbb{E}\Big[\phi^j(X_{\tau_a^{n_{a,t}}})\Big]\Big]$ and $\mathbf{y}_a^j = \Big[f_{a,\tau_a^1}^j, f_{a,\tau_a^2}^j, \cdots, f_{a,\tau_a^{n_{a,t}}}^j\Big]^\top$. By solving the optimization problem (2.5), the parameter θ_a^j is estimated as

$$\hat{\theta}_a^j = {\mathbf{C}_a^j}^{-1} \mathbf{\Phi}_a^j \mathbf{y}_a^j \tag{2.6}$$

where $\mathbf{C}_a^j = \mathbf{\Phi}_a^j {\mathbf{\Phi}_a^j}^\top + \lambda \mathbf{I}$. Then, the estimated feedback with respect to candidate x_t^i can be calculated as

$$\hat{f}_{a,t}^{i,j} = \phi^j(x_t^i)^\top \hat{\theta}_{a,t}^j \tag{2.7}$$

whose confidence width [36, 57] is

$$w_{a,t}^{i,j} = \sqrt{\phi^j(x_t^i)^\top (\mathbf{C}_{a,t}^j)^{-1} \phi^j(x_t^i)}. \tag{2.8}$$

As the algorithm may not have access to the mapping function $\phi^j(x)$, we need to represent Eqn. (2.7) and Eqn. (2.8) by kernel function. By the Woodbury matrix identity, we have

$$\begin{aligned} \hat{f}_{a,t}^{i,j} &= \phi^j(x_t^i)^\top (\mathbf{\Phi}_{a,t}^j {\mathbf{\Phi}_{a,t}^j}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Phi}_{a,t}^j \mathbf{y}_{a,t}^j \\ &= \phi^j(x_t^i)^\top \mathbf{\Phi}_{a,t}^j ({\mathbf{\Phi}_{a,t}^j}^\top \mathbf{\Phi}_{a,t}^j + \lambda \mathbf{I})^{-1} \mathbf{y}_{a,t}^j \end{aligned} \tag{2.9}$$

and

$$\begin{aligned} \lambda {w_{a,t}^{i,j}}^2 = {} & \phi^j(x_t^i)^\top \phi^j(x_t^i) - \\ & \phi^j(x_t^i)^\top \mathbf{\Phi}_{a,t}^j ({\mathbf{\Phi}_{a,t}^j}^\top \mathbf{\Phi}_{a,t}^j + \lambda \mathbf{I})^{-1} {\mathbf{\Phi}_{a,t}^j}^\top \phi^j(x_t^i). \end{aligned} \tag{2.10}$$

Algorithm 1 Multi-Feedback Probabilistic Contextual UCB

```
Inputs :
Arm set $\mathcal{A}$, a horizon $T$, kernel function $k$ and parameter $\alpha$ and $\lambda$.
for $t = 1, \cdots, T$ do
  Receive a set of probabilities $Pr_t(\mathcal{X})$ for the candidates in the context bundle $\mathcal{X}$
  for $a \in \mathcal{A}$ do
    Calculate $\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})]$ and $\mathbb{E}[w_{a,t}]$ according to Eqn. (2.13) and Eqn. (2.14).
  end for
  $a_t = \arg\max_{a \in \mathcal{A}}(\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] + L\beta\mathbb{E}[w_{a,t}])$
  Receive feedback $\mathbf{f}_{a,t} = \left[f^1_{a,t}, \cdots, f^J_{a,t}\right]$.
  Update $\mathbf{y}^j_{a,t+1}$, $\mathbf{K}^j_{a,t+1}$ and $\mathbf{D}^j_{a,t+1}$
end for
```

Denote $\mathbf{k}^{i,j}_{a,t} = {\mathbf{\Phi}^j_{a,t}}^\top \phi^j(x^i_t)$ and $\mathbf{K}^j_{a,t} = {\mathbf{\Phi}^j_{a,t}}^\top \mathbf{\Phi}^j_{a,t}$. The pth entry in $\mathbf{k}^j_{a,t}$ is $\mathbb{E}\left[\phi^j(X_{\tau^p_a})\right]^\top \phi^j(x^i_t) = \sum_{n=1}^N P(x^i_{\tau^p_a}) k^j(x^n_{\tau^p_a}, x^i_t)$. Similarly, the entry of $\mathbf{K}^j_{a,t}$ in the pth row and qth column is $\mathbb{E}\left[\phi^j(X_{\tau^p_a})\right]^\top \mathbb{E}[\phi^j(X_{\tau^q_a})] = \sum_{n,m=1}^N P(x^n_{\tau^p_a}) P(x^m_{\tau^q_a}) k^j(x^n_{\tau^p_a}, x^m_{\tau^q_a})$. Now, the estimated feedback can be represented as

$$\hat{f}^{i,j}_{a,t} = {\mathbf{k}^{i,j}_{a,t}}^\top (\mathbf{D}^j_{a,t})^{-1} \mathbf{y}^j_{a,t} \tag{2.11}$$

and the confidence width is

$$w^{i,j}_{a,t} = \sqrt{\frac{1}{\lambda} k^j(x^i_t, x^i_t) - \frac{1}{\lambda} {\mathbf{k}^{i,j}_{a,t}}^\top (\mathbf{D}^j_{a,t})^{-1} \mathbf{k}^{i,j}_{a,t}} \tag{2.12}$$

where $\mathbf{D}^j_{a,t} = \mathbf{K}^j_{a,t} + \lambda \mathbf{I}$.

2.4.2 Multi-Feedback Probabilistic Contextual UCB

Based on the results of empirical risk minimization, the proposed Multi-Feedback Probabilistic Contextual UCB is given in Algorithm 1.

At each round, the algorithm needs to get the estimated expected reward and the corresponding expected confidence width. To do so, the algorithm first calculates estimated feedbacks and corresponding confidence widths according to Eqn. (2.11) and Eqn. (2.12), respectively. Then, given the utility function $U_t : \mathcal{R}^J \to \mathcal{R}$, the estimated reward with respect to the ith context candidate is predicted as $U_t(\hat{\mathbf{f}}^i_{a,t})$ where $\hat{\mathbf{f}}^i_{a,t} = [\hat{f}^{i,1}_{a,t}, \hat{f}^{i,2}_{a,t}, \cdots \hat{f}^{i,j}_{a,t}, \cdots]^\top$. If the exact context is not given, the estimated feedback $\hat{\mathbf{f}}_{a,t}$ is a random vector over the probabilistic context space, so the estimated expected reward over the probabilistic context space is written as

$$\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] = \sum_i P_t(x^i_t) U_t(\hat{\mathbf{f}}^i_{a,t}). \tag{2.13}$$

The confidence width is important for arm exploration, but it is not trivial to get the expected confidence width. Here, we calculate the upper bound of the expected confidence width over the probabilistic context space by exploiting Lipschitz continuity of the utility function. Concretely, if L is the Lipschitz constant of the utility function, the upper bound of expected confidence width over the probabilistic context space is calculated as

$$\mathbb{E}[w_{a,t}] = \sum_{i=1}^{N} P_t(x^i_t) \sum_{j=1}^{J} w^{i,j}_{a,t}. \tag{2.14}$$

The detailed derivation of Eqn. (2.14) will be given in Lemma 4.

With the estimated expected reward and the corresponding expected confidence width, the selected arm is $a_t = \arg\max_{a\in\mathcal{A}}(\mathbb{E}[U_t(\hat{\mathbf{f}}_{a,t})] + L\beta\mathbb{E}[w_{a,t}])$, where β is a hyperparameter to balance the exploration and exploitation.

Another arm selection policy is to select an arm based on the reward UCB with respect to the most probable context. Formally, in this way, the selected arm is

$$a_t = \arg\max_{a\in\mathcal{A}} \left(U_t\left(\hat{\mathbf{f}}_{a,t}\left(\bar{x}_t\right)\right) + L\beta w_{a,t}\left(\bar{x}_t\right)\right) \tag{2.15}$$

where $\bar{x}_t = \arg\max_{x\in\mathcal{X}} P_t\left(x\right)$ is the most probable context and the jth entry of $\hat{\mathbf{f}}_{a,t}$ is $\hat{f}^j_{a,t}(\bar{x}_t) = \phi^j(\bar{x}_t)^\top(\mathbf{\Phi}^j_{a,t}{\mathbf{\Phi}^j_{a,t}}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}^j_{a,t}\mathbf{y}^j_{a,t}$. In the next section, we will show that this arm selection policy has a linear regret under our regret definition in Eqn. (2.4).

2.5 Regret Analysis

In this section, we analyze the regret with respect to an oracle that also has the probabilistic context information and establish an upper bound on cumulative regret of Algorithm 1 which shows the cumulative regret sub-linearly increases with $\mathcal{O}(\sqrt{T\log T})$, followed by the proof sketch.

2.5.1 Cumulative Regret Bound

The following theorem provides an upper bound on the cumulative regret of Algorithm 1.

Theorem 1 *Assume at round t, the utility function $U_t(\mathbf{f}_{a,t}) \in [0,1]$ satisfies $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ with Lipschitz constant L, kernel function is $k^j(x,x') \leq c_k$ such that $\phi^j\left(x\right) \succeq 0$. At each round, the agent receives a probabilistic context set $\mathcal{X}$ and the corresponding* *probability set $Pr_t(\mathcal{X})$, selects arm from $\mathcal{A}$ by Algorithm 1 and get J different feedbacks. With probability $1-\delta$, the cumulative expected regret R_T of Algorithm 1 is bounded by*

$$\begin{aligned} R_T &\leq 2L\beta J|\mathcal{A}||\mathcal{X}|\sqrt{2q\gamma_m T\log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})} \\ &= \mathcal{O}(\sqrt{T\log T}) \end{aligned} \tag{2.16}$$

where γ_m is the maximum rank of $\mathbf{K}^j_{a,t}$, $q=\max(1,\frac{c_k}{\lambda})$ and

$$\beta = (\sqrt{\tfrac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda}).$$

Remark 2 *Theorem 1 shows that, for the bandit setting with probabilistic contexts and multiple feedbacks, our proposed algorithm can achieve a sub-linear cumulative expected regret bound $\mathcal{O}(\sqrt{T\log T})$. This demonstrates the effectiveness of our proposed algorithm.*

Remark 3 *Compared with the cumulative regret bound of kernel-UCB in the standard bandit setting [34, 36], the cumulative regret bound of the proposed algorithm is scaled by Lipschitz constant L, number of feedbacks J and size of context bundle $|\mathcal{X}|$. As a result, the regret in our setting is more difficult to be reduced than that in the standard setting. Nonetheless, by the proposed algorithm, the cumulative regret can still be guaranteed to be sub-linear.*

2.5.2 Proof Sketch

The proof sketch of Theorem 1 is given below. Compared with other UCB algorithms [1, 36, 57], the consideration of probabilistic context, multiple noisy feedbacks and Lipschitz utility function adds new challenges to the regret bound proof. First, since the algorithm predicts feedbacks instead of the reward, the predicted feedbacks can be guaranteed to converge to the expected feedbacks by Lemma 1 in [36], but we still need to bound the gap between the estimated reward and the true expected reward. Second, since the proposed algorithm only has probabilistic contexts, we can bound the expected reward estimation error by the expected confidence width, but it is still challenging to get the sum of the confidence width over time. Next, we show several important lemmas to address the aforementioned challenges.

First, by exploiting the Lipschitz continuity of utility function, the confidence width of estimated reward is bounded in Lemma 4, which also explains the setting of confidence width in Algorithm 1.

Lemma 4 (Concentration of Empirical Risk Minimization) *Assume the utility function $U(\cdot)$ is in a linear form or multiplication form with Lipschitz constant L. With probability at least $1-\frac{\delta}{T}$, for $\forall a \in \mathcal{A}$, we have*

$$\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t})\right]-\mathbb{E}\left[U_t(\mathbf{f}_{a,t})\right]\right| \leq L\beta\mathbb{E}[w_{a,t}] \tag{2.17}$$

where $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}}+c\sqrt{\lambda})$.

Proof. Let $g_{a,t}^{i,j} = g_a^j\left(x_t^i\right)$ and $\mathbf{g}_{a,t}^i = \left[g_a^{i,1}, \cdots, g_a^{i,j}\right]^\top$.By the assumption $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$, we have

$$\begin{aligned}
&\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t})\right]-\mathbb{E}\left[U_t(\mathbf{f}_{a,t})\right]\right|=\left|\sum_{i=1}^{N} P_t\left(x_t^i\right)\left(U_t\left(\hat{\mathbf{f}}_{a,t}^i\right)-U_t\left(\mathbf{g}_{a,t}^i\right)\right)\right| \\
&\leq \sum_{i=1}^{N} P_t\left(x_t^i\right) L\left\|\hat{\mathbf{f}}_{a,t}^i-\mathbf{g}_{a,t}^i\right\| \leq L\sum_{i=1}^{N} P_t\left(x_t^i\right)\sum_{j=1}^{J}\left|\hat{f}_{a,t}^{i,j}-g_{a,t}^{i,j}\right|.
\end{aligned} \tag{2.18}$$

By Lemma 1 in [36], we have $\left|\hat{f}_{a,t}^{i,j}-g_{a,t}^{i,j}\right| \leq \beta w_{a,t}^{i,j}$ with probability at least $1-\frac{\delta}{JT}$. Thus, with probability at least $1-\frac{\delta}{T}$, we have $\left|\mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a,t})-U_t(\mathbf{f}_{a,t})\right]\right| \leq L\beta\sum_{i=1}^N P_t(x_t^i)\sum_{j=1}^J w_{a,t}^{i,j} = L\beta\mathbb{E}[w_{a,t}]$. ■

Then, by using Lemma 4, we will bound the regret by the expected confidence width in the next lemma.

Lemma 5 (Regret Bound by Confidence Width) *Assume that the utility function $U_t(\cdot)$ satisfies $\mathbb{E}_\epsilon[U_t(\mathbf{f}_{a,t})] = U_t(\mathbb{E}_\epsilon[\mathbf{f}_{a,t}])$ with Lipschitz constant L. With probability at least $1-\frac{\delta}{T}$, the cumulative regret satisfies*

$$R_T = \sum_{t=1}^{T} reg_t \leq 2L\beta \sum_{t=1}^{T} \mathbb{E}[w_{a_t,t}] \tag{2.19}$$

where $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda})$.

Proof. By similar proof techniques for standard UCB [1, 57], with probability at least $1-\frac{\delta}{T}$, the instant regret is bounded as

$$\begin{aligned} reg_t &= \mathbb{E}\left[U_t(\mathbf{f}_{a_t^*,t}) - U_t(\hat{\mathbf{f}}_{a_t^*,t}) + U_t(\hat{\mathbf{f}}_{a_t^*,t}) - U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq L\beta\mathbb{E}[w_{a_t^*,t}] + \mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a_t^*,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq L\beta\mathbb{E}[w_{a_t,t}] + \mathbb{E}\left[U_t(\hat{\mathbf{f}}_{a_t,t})\right] - \mathbb{E}\left[U_t(\mathbf{f}_{a_t,t})\right] \\ &\leq 2L\beta\mathbb{E}[w_{a_t,t}] \end{aligned} \tag{2.20}$$

where the first and third inequalities hold by Lemma 4 and the second inequality holds by arm selection policy in Algorithm 1. In this way, the cumulative regret is bounded as Eqn. (2.19) ■

The next challenge is to bound the sum of confidence width, which is expressed as

$$\sum_{t=1}^{T} \mathbb{E}[w_{a_t,t}] = \sum_{t=1}^{T}\sum_{i=1}^{N} P_t(x_t^i) \sum_{j=1}^{J} \sqrt{\phi^j(x_t^i)^\top (\mathbf{C}_{a,t}^j)^{-1} \phi^j(x_t^i)}. \tag{2.21}$$

We cannot directly use Sylvester's determinant theorem or Schur's determinant identity like in the proofs of Lemma 11 in [1] and Lemma 7 in [36]. Thus, we first derive Lemma 6 to get an upper bound of $\mathbb{E}[w_{a_t,t}]$, and then get the sum of the expected confidence width in Lemma 6

Lemma 6 (Sum of Confidence Width) *Assume kernel function k^j is chosen such that mapping function $\phi^j(x) \succeq 0$, we have*

$$\sum_{t=1}^{T} \mathbb{E}[w_{a_t,t}] \leq J|\mathcal{X}|\sqrt{2q\gamma_m T \log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})} \tag{2.22}$$

where γ_m is the maximum rank of $\mathbf{K}^j_{a,t}$, $q = \max(1, \frac{c_k}{\lambda})$.

Proof. First, we bound $\mathbb{E}[w^j_{a,t}]$ by $\bar{w}^j_{a,t}$ where $\bar{w}^j_{a,t} = \sqrt{\mathbb{E}[\phi^j(X_t)]^\top(\mathbf{C}^j_{a,t})^{-1}\mathbb{E}[\phi^j(X_t)]}$. Let $w^{i,j}_{a_t,t} = \sqrt{\phi^j(x^i_t)^\top(\mathbf{C}^j_{a,t})^{-1}\phi^j(x^i_t)}$. Then, we have

$$\begin{aligned}\left(P_t(x^i_t)w^{i,j}_{a_t,t}\right)^2 &= P_t(x^i_t)\phi^j(x^i_t)^\top(\mathbf{C}^j_{a,t})^{-1}P_t(x^i_t)\phi^j(x^i_t) \\ &\leq \mathbb{E}[\phi^j(X_t)]^\top(\mathbf{C}^j_{a,t})^{-1}\mathbb{E}[\phi^j(X_t)]\end{aligned} \tag{2.23}$$

where the inequality holds because $\phi^j(x) \succeq 0$ and thus $\mathbb{E}[\phi^j(X_t)] = \sum_{n=1}^{|\mathcal{X}|} P_t(x^n_t)\phi^j(x^n_t) \geq P_t(x^i_t)\phi^j(x^i_t)$. By taking squared root of both sides of Eqn. (2.23), we have $P_t(x^i_t)w^{i,j}_{a,t} \leq \bar{w}^j_{a,t}$, and thus $\mathbb{E}[w^j_{a,t}] = \sum_{i=1}^{|\mathcal{X}|} P_t(x^i_t)w^{i,j}_{a,t} \leq |\mathcal{X}|\bar{w}^j_{a,t}$. Since $\sum_{t=1}^{T} \bar{w}^j_{a_t,t}$ can be bounded by Lemma 8 in [36], i.e. $\sum_{t=1}^{T} \bar{w}^j_{a_t,t} \leq \sqrt{2q\gamma_m T \log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})}$, the inequality (2.22) can be proved. ■

By substituting Eqn. (2.22) into Eqn. (2.19), we can get the cumulative regret bound in of Algorithm 1 in Theorem 1.

2.5.3 Regret of Arm Selection by Most Probable Context

The arm selection policy based on the most probable context is shown in Eqn. (2.15). We define the error between expected reward and the most probable reward as

$$M_{a,t} = \mathbb{E}_{\mathcal{X}}\left[U_t\left(\mathbf{g}_{a,t}\right)\right] - U_t\left(\mathbf{g}_a\left(\bar{x}_t\right)\right) \tag{2.24}$$

where the expectation is taken over context space, $\mathbf{g}_{a,t} = \left[g_a^1\left(X_t\right), \cdots, g_a^J\left(X_t\right)\right]^\top$ and $\mathbf{g}_a\left(\bar{x}_t\right) = \left[g_a^1\left(\bar{x}_t\right), \cdots, g_a^J\left(\bar{x}_t\right)\right]^\top$. Assume that $|M_{a,t}| \leq M$ for $t = 1, \cdots, T$ and $x \in \mathcal{X}$ Then, the cumulative expected regret is given in the theorem below and includes a linear term.

Theorem 7 *Under the same assumptions as in Theorem 1 and the algorithm selects arm from $\mathcal{A}$ based on Eqn.* (2.15)*, then with probability $1-\delta$, the cumulative expected regret defined by Eqn.* (2.4) *is bounded by*

$$\begin{aligned} R_T \leq 2L\beta J|\mathcal{A}||\mathcal{X}|\sqrt{2q\gamma_m T \log(\frac{(T+1)c_k+\lambda}{d^{\frac{1}{\gamma_m}}\lambda})}+ \\ \sum_{t=1}^{T}\left(M_{a_t^*,t} - M_{a_t,t}\right) = \mathcal{O}(\sqrt{T\log T} + 2MT) \end{aligned} \tag{2.25}$$

where γ_m is the maximum rank of $\mathbf{K}_{a,t}^j$), $q = \max(1, \frac{c_k}{\lambda})$ and $\beta = (\sqrt{\frac{\log(2TJ|\mathcal{A}|/\delta)}{2}} + c\sqrt{\lambda})$.

2.6 Simulation Results

We now apply Algorithm 1 to the problem of DNN model selection for mobile devices and show its performance in terms of average reward and cumulative regret.

2.6.1 Application to DNN Model Selection

The recent breakthrough in DNN model compression has made it possible to run DNN inference on edge devices (e.g., mobile phones and tablets). While they can have similar inference accuracies, different DNN models have different latencies and energy consumption under different system conditions. Thus, it is crucial to select an optimal DNN model for edge inference with the best user experience. This is challenged by the fact that, although the basic configuration of an edge device (e.g., CPU, OS, RAM) requesting a DNN model is exposed to the model provider, the device's actual resource management and system condition (e.g., available system resources) that decide the latency and energy of the deployed DNN model can only be known probabilistically. Thus, the problem of DNN model selection is suitable for our considered bandit setting with probabilistic contexts and multiple feedbacks.

Concretely, the DNN models to deploy on edge devices constitute the set of arms, an edge device's actual system condition is the context, and we consider DNN inference latency l and energy consumption e as the two feedback signals. Our goal is to select optimal DNN models for edge devices that arrive sequentially. For evaluation purposes, we run experiments and collect measured data of five image classification DNN models from TensorFlow Hub running on two cellphones (Vivo V1838A and Google Pixel 3a) and two tablets (Samsung - Galaxy Tab A7 and Vankyo MatrixPad Z4). We use these four devices to represent four types of actual system conditions (i.e., context in our study) in an edge device requesting a DNN model. In other words, when an edge device arrives, its actual system condition is assumed to fall into one of the conditions as specified by the four different devices in our evaluation. While we can further run experiments on these devices under different usage scenarios to have more fine-grained types of contexts, our current setup is enough to validate our theoretical analysis. Note that although an edge device's basic hardware configuration is accessible to the DNN model provider, its actual system condition (i.e., context in our problem) is only known probabilistically for DNN model selection.

Table 2.1: Average Energy Consumption

	Phone 1	Phone 2	Tablet 1	Tablet 2
InceptionV2Q	0.45 J	0.07 J	6.18 J	1.41 J
InceptionV4Q	1.88 J	0.22 J	11.66 J	6.59 J
InceptionV4F	5.04 J	1.14 J	37.29 J	10.87 J
MobileNetV1Q	0.13 J	0.03 J	2.00 J	0.69 J
MobileNetV1F	0.18 J	0.04 J	2.00 J	0.60 J

Table 2.2: Average Latency

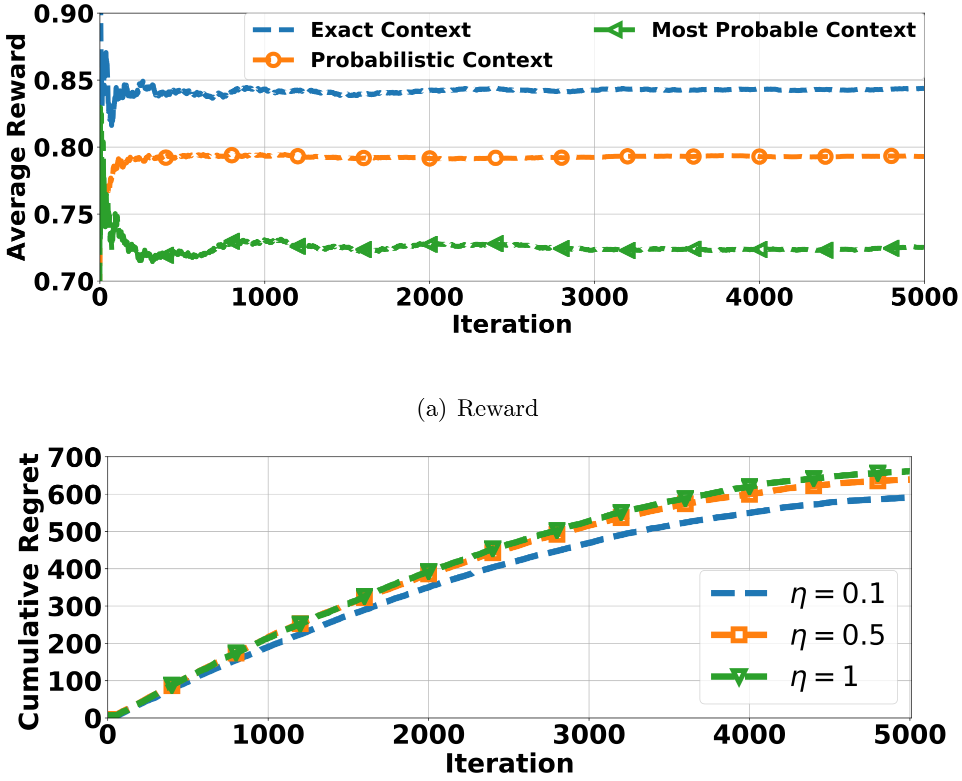
	Phone 1	Phone 2	Tablet 1	Tablet 2
InceptionV2Q	$0.33s$	$0.11s$	$2.60s$	$0.57s$
InceptionV4Q	$1.40s$	$0.35s$	$4.45s$	$2.53s$
InceptionV4F	$2.01s$	$1.23s$	$18.95s$	$4.58s$
MobileNetV1Q	$0.10s$	$0.05s$	$0.83s$	$0.22s$
MobileNetV1F	$0.13s$	$0.07s$	$0.60s$	$0.25s$

We assume that the utility function for a DNN model selection decision is a weighted linear combination of energy consumption and latency, while noting that the weights can change for different devices (e.g., energy consumption plays a more important role for devices with a small battery capacity). For illustration, we assume that the utility function can be either $-0.36e - 0.54l + 1$ or $-0.25e - 0.65l + 0.9$, which is randomly determined over time. We compare our algorithm with kernel contextual bandit algorithms that utilize the exact context and the most probable context, respectively. To simulate the probabilistic environment, we randomly generate the $Pr_t(\mathcal{X})$ as input at each round. We use the radial basis function kernel $k(x, x') = \exp(-\rho\|x - x'\|_2^2)$.

2.6.2 Results

In Fig. 2.1(a), we show the average reward achieved by different algorithms. Naturally, the algorithm with the exact true context achieves the highest reward. Nonetheless, the reward of our algorithm is greater than that of the straightforward algorithm that utilizes the most probable context as if it were the true context (similar to a standard UCB algorithm). The reason for the low reward achieved by using the most probable context is that the stored possibly erroneous contexts can be uncorrelated with the received feedback, thus resulting in biased estimation of the feedback functions and hence inaccurate reward prediction further.

In Fig. 2.1(b), to validate the sub-linear regret, we compare the cumulative regret of our algorithm to the oracle that knows the optimal arm for any probabilistic context bundle. We use entropy of context's probability distribution $H(Pr_t(\mathcal{X}))$ as a measure for how random the provided context bundle is. We denote $H_{max} = \log_2 |\mathcal{X}|$ as the largest entropy and η $(0 \leq \eta \leq 1)$ as a threshold to bound randomness for different rounds, where $H(Pr_t(\mathcal{X})) \leq \eta H_{max}$. The smaller η, the more concentrated probabilistic distribution $Pr_t(\mathcal{X})$ of a context bundle (or, less randomness). If $\eta = 0$, then the distribution only reveals the exact context. The result shows that the regret of our algorithm is sub-linearly increasing, regardless of randomness of probabilistic bundle.



(a) Reward

(b) Regret

Figure 2.1: Performance comparison.

Chapter 3

Contextual Bandit with Delayed Feedback

3.1 Introduction

In a standard contextual bandit setting, given context information revealed at the beginning of each round, the agent selects an arm for the upcoming round; then, the resulting reward will be promptly provided to the agent at the end of each round, and utilized for learning the reward function in terms of context-arm pairs and refining future arm selections. Nonetheless, this setting may not be satisfied in practice [104]. Take the user rating as an example. It is common that a user only provides rating feedback (i.e., reward) several hours/days later or even declines to given ratings, whereas meanwhile many item recommendation decisions need to be made. Additionally, in some applications, the true reward can only be inaccurately observed with errors, which add more noises to the agent's feedback. For example, when applying contextual bandit to machine learning model selection for inference on edge devices [71], the model performance (e.g., latency performance) on a device can only be observed based on empirical measurement, which naturally deviates from the true model performance and hence constitutes noisy reward feedback.

In this chapter, we extend the standard contextual bandit setting to address delayed or missing feedback. Concretely, the reward feedback for an arm selected at the beginning of a round may not be sent back to the agent at the end of this round; instead, the reward feedback is either missing or only sent to the agent after some a priori unknown delays with some observation noise.

We first propose a bandit learning algorithm that learns the reward function by only using available reward feedbacks and selects arms based on the upper confidence bound (UCB) to account for the balance between exploration and exploitation. Importantly, when the maximum delay is finite, we theoretically prove that the upper bound on the cumulative regret (i.e., the difference between the reward achieved by our algorithm and the optimal oracle's reward) grows sub-linearly in time as $\mathcal{O}(\sqrt{T \log T})$, compared to the oracle that knows the optimal arm given any context information. When delay is infinite or some rewards are missing, the sub-linear regret bound may fail. Instead, motivated by semi-supervised learning that produces pseudo labels for unlabeled data to further improve the model performance [28], we generate fictitious estimates of rewards that have yet to arrive based on currently available feedbacks. Thus, by combining semi-supervised learning with online contextual bandit learning, we propose a novel extension and design another algorithm, which finds fictitious values for currently unavailable reward feedbacks to minimize the estimation error. To our knowledge, our algorithm is the first to leverage semi-supervised learning in contextual bandits with delayed and missing feedback.

To evaluate our algorithms, we conduct a simulation study on a classical Yahoo Module dataset to build an online context-aware news recommend system and find suitable articles for users. Our empirical results show that the delayed UCB algorithm yields a sub-linear regret with respect to the oracle, validating our regret analysis in finite-delay setting. Moreover, the incurred regret can be further effectively reduced by using our proposed estimations for those feedback values that have yet to arrive when the feedback is delayed or missing with a high probability.

3.2 Related Work

Adversarial bandit is another important variant of the bandit setting [10], where an adversary at each iteration chooses the reward policy for each arm. Further, [72] considers multi-objective optimization of generalized linear bandits under contextual settings and defines the best arm in terms of Pareto optimality. Another variant of contextual bandit assumes context information is not complete. For example, [108] considers the existence of partially hidden context and proposes to learn both reward function and hidden information. In these studies, reward feedback is still provided to the agent without delays.

For bandits with delayed feedback, [81] considers a fixed delay in bandits, and [26] later proposes EXP3-based algorithms for non-stochastic bandits with fixed delays. Stochastic bandits with random delays have been considered by [83]. Also, [16] considers the adversarial setting and propose delayed-EXP3 algorithm. The study [25] considers a more general assumption, where delayed feedback is composite and anonymous. In [99], author considers the bandit problem with fixed delay in non-stochastic setting and proposed algorithm based on EXP3. In [104], DeLin-UCB is proposed for a contextual bandit setting with delayed feedback, and [121] considers a general linear form and proposes DUCB to achieve a sublinear regret bound. The most close work to us is perhaps [21], which utilizes clustering to estimate the reward feedback for unlabeled contexts. In contrast, we propose semi-supervised learning to estimate fictitious rewards for those that are delayed/missing.

3.3 Problem Formulation

We first consider the setting of contextual bandit with delayed feedback. Given context information at each round $t = 1, 2, \cdots, T$, the agent/learner needs to select an arm (e.g., to serve a request or a user depending on specific applications). We denote $x_{a,t} \in \mathcal{R}^M$ as the context, which is a representation of the environment information or feature regrading arm a at the t-th round, for $a \in \mathcal{A} = \{1, 2, \cdots, K\}$ and $t = 1, 2, \cdots, T$. For a selected arm a at round t, we denote the resulting reward as $y_{a,t} \in \mathcal{R}$. Nonetheless, due to feedback delays, the agent can only receive the reward feedback at the beginning of the $(t + d_t)$-th round, where $d_t \geq 1$ is the delay for the arm selected at round t. Note that it is possible that the learner simultaneously receives multiple feedback signals for arms selected in prior rounds.

We denote the expected reward function as $g(x_{a,t})$, and we assume that the expected rewards are similar in similar contexts. This assumption is formalized by the following Hölder condition for $g(x_{a,t})$.

Assumption 1 *For any arm $a \in \mathcal{A}$, there exist $L > 0$ and $\beta > 0$, such that for any $x, x' \in \mathcal{R}^M$, it holds that*

$$|g(x) - g(x')| \leq L||x - x'||^{\beta}$$

where $|| \cdot ||$ denotes the Euclidean norm in $\mathcal{R}^M$

We use the kernel method to model non-linear reward functions in terms of the context and arm. Given a kernel function $k(x, x') = \phi(x)^{\top}\phi(x'), \forall x, x' \in \mathcal{R}^M$, we can express the expected reward function as $g(x_{a,t}) = \phi(x_a)^{\top}\theta$ in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ corresponding to the kernel function $k(x, x')$. Specifically, the actual reward feedback $y_{a,t}$ received by the agent (after a delay of d_t rounds) for its arm a selected at round t is written as

$$y_{a,t} = g(x_{a,t}) + \epsilon_t = \phi(x_a)^{\top}\theta + \epsilon_t, \tag{3.1}$$

where ϵ_t is the zero-mean and delay-independent noise. For the ease of analysis, we assume that ϵ_t is independent and identically distributed.

The goal of the agent is to maximize its total expected reward, or equivalently minimize its cumulative regret, over T rounds. We define the best arm given context $x_{a,t}$ at round t as the arm that leads to the highest expected reward, i.e.,

$$a_t^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}\left[y_{a,t}\right] = \arg\max_{a \in \mathcal{A}} g(x_{a,t}). \tag{3.2}$$

With arm a_t selected at round t, the expected instant regret reg_t can be expressed as

$$reg_t = \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right], \tag{3.3}$$

where the expectation is taken over the noises ϵ_t. Thus, the agent needs to find an arm selection policy based on the received feedback signals and context-arm history to minimize the cumulative regret:

$$R_T = \sum_{t=1}^{T} reg_t. \tag{3.4}$$

3.4 Delayed Contextual UCB

For the considered bandit setting with delayed feedback, we provide an algorithm based on delayed contextual upper confidence bound (UCB). We first utilize a kernel-based empirical risk minimization policy for reward estimation based on received feedback. Then, given the predicted feedback and its confidence width, we design a delayed contextual UCB arm selection policy.

3.4.1 Reward Estimation

At each round, in order to select an arm, the agent should be able to estimate the expected reward corresponding to each arm given the provided context. Similar to previous work [57], to achieve feedback prediction, we can estimate the parameter θ in feedback functions by kernel-based empirical risk minimization based on the history. Due to existence of delay in feedback, we denote the set of rounds whose rewards are fed back to the agent prior to round t as $\mathcal{T}_t = \{\tau_1, \tau_2, \cdots, \tau_{n_t}\}$, where $n_t = |\mathcal{T}_t|$. Denote $\mathbf{\Phi}_t = \left[\phi(x_{\tau_1}), \cdots, \phi(x_{\tau_{n_t}})\right]$, $\mathbf{y}_t = \left[y_{\tau_1}, \cdots, y_{\tau_{n_t}}\right]^\top$. Then, we formulate the kernel-based empirical risk minimization as an optimization problem below:

$$\hat{\theta}_t = \arg\min_{\theta} \frac{1}{n_t} \sum_{\tau \in \mathcal{T}_t} (\phi(x_\tau)^\top \theta - y_\tau)^2 + \lambda \|\theta\|^2 \tag{3.5}$$

where $\lambda \geq 0$ is a hyper-parameter.

By solving the optimization problem in (3.5), the parameter θ is estimated at round t in a closed form as

$$\hat{\theta}_t = {\mathbf{C}_t}^{-1}\mathbf{\Phi}_t\mathbf{y}_t$$

where $\mathbf{C}_t = \mathbf{\Phi}_t{\mathbf{\Phi}_t}^\top + \lambda\mathbf{I}$. Then the expected reward with respect to $x_{a,t}$ can be calculated as

$$\hat{g}_{a,t} = \phi(x_{a,t})^\top{\mathbf{C}_t}^{-1}\mathbf{\Phi}_t\mathbf{y}_t. \tag{3.6}$$

By the Woodbury matrix identity, we rewrite Eqn. (3.6) as

$$\begin{aligned}\hat{g}_{a,t} &= \phi(x_{a,t})^\top(\mathbf{\Phi}_t{\mathbf{\Phi}_t}^\top + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_t\mathbf{y}_t \\ &= \phi(x_{a,t})^\top\mathbf{\Phi}_t(\mathbf{K}_t + \lambda\mathbf{I})^{-1}\mathbf{y}_t \\ &= {\mathbf{k}_{a,t}}^\top(\mathbf{D}_t)^{-1}\mathbf{y}_t,\end{aligned} \tag{3.7}$$

where $\mathbf{k}_{a,t} = {\mathbf{\Phi}_t}^\top\phi(x_{a,t})$, $\mathbf{K}_t = {\mathbf{\Phi}_t}^\top\mathbf{\Phi}_t$ and $\mathbf{D}_t = \mathbf{K}_t + \lambda\mathbf{I}$. Note that $\mathbf{K}_t$ and $\mathbf{k}_{a,t}$ can be computed by the kernel function $k(\cdot,\cdot)$ which is available to the agent.

The estimation error is bounded as follows.

Lemma 8 *Suppose that the expected reward $g_{a,t}$ belongs to the RKHS generated by kernel function k, i.e. $g_{a,t} = \phi\left(x_{a,t}\right)^\top\theta$. If Eqn. (3.7) is used for reward estimation, then with probability at least $1 - \frac{\delta}{T}$, the estimation error is*

$$|\hat{g}_{a,t} - g_{a,t}| \leq (\alpha + \lambda)\, w_{a,t} \tag{3.8}$$

where $w_{a,t} = \sqrt{\phi(x_{a,t})^\top(\mathbf{C}_t)^{-1}\phi(x_{a,t})}$ and $\alpha = \sqrt{\frac{1}{2}\ln\frac{2KT}{\delta}}$. ■

The proof of Lemma 8 is in the supplementary material. To compute the reward estimation error bound, we apply Woodbury matrix identity again and get

$$\lambda {w_{a,t}}^2 = \phi(x_{a,t})^\top \phi(x_{a,t}) - \\ \phi(x_{a,t})^\top \mathbf{\Phi}_t (\mathbf{K}_t + \lambda \mathbf{I})^{-1} {\mathbf{\Phi}_t}^\top \phi(x_{a,t}).$$

Therefore, the estimation error bound can be expressed as

$$w_{a,t} = \sqrt{\frac{1}{\lambda}(k(x_{a,t}, x_{a,t}) - {\mathbf{k}_{a,t}}^\top (\mathbf{D}_t)^{-1} \mathbf{k}_{a,t})}. \tag{3.9}$$

By Lemma 8, we get the upper confidence bound of the expected reward, i.e.

$$g_{a,t} \leq \hat{g}_{a,t} + (\alpha + \lambda) w_{a,t},$$

which is used for arm selection given context information.

3.4.2 Delayed Contextual UCB for Arm Selection

The delayed contextual UCB-based arm selection algorithm is shown in Algorithm 2. Specifically, the agent chooses an arm that has the maximum UCB to balance exploration and exploitation. Assume that at round t, the agent receives a set of reward feedbacks for arms selected at rounds in the set $S_t = \{\tau_1, \cdots, \tau_I\}$, where $\tau_i + d_{\tau_i} = t$ for $i = 1, \cdots, I$. If S_t is not empty, then for each $\tau_i \in \mathcal{S}_t$, the algorithm augments the received reward $y_{a_{\tau_i}, \tau_i}$ into $\mathbf{y}_t$ and updates $\mathbf{K}_t$. Then, by empirical risk minimization, the estimated expected reward and its corresponding estimation error can be calculated by Eqn. (3.7) and Eqn. (3.9), respectively. Finally, the agent selects arm a_t at round t based on the UCB policy.

Algorithm 2 Delayed Contextual UCB

```
**Inputs** : kernel function $k(,\cdot,)$ and parameter $\alpha$ and $\lambda$.
**for** $t = 1, \cdots, T$ **do**
    **if** length($\mathbf{y}_t$) $= 0$ **then**
        Randomly choose arm $a_t$
    **else**
        **if** $|\mathcal{S}_t| \neq 0$ **then**
            For $\tau_i \in \mathcal{S}_t$, augment $y_{a_{\tau_i}, \tau_i}$ into $\mathbf{y}_t$ and update $\mathbf{K}_t$
        **end if**
        Receive context $x_{a,t}$, $a = 1, \cdots, K$
        **for** $a \in \mathcal{A}$ **do**
            Calculate $\hat{g}_{a,t}$ in Eqn. (3.7) and $w_{a,t}$ in Eqn. (3.9)
        **end for**
        Select arm $a_t = \arg\max_{a \in \mathcal{A}}(\hat{g}_{a,t} + (\alpha + \lambda) w_{a,t})$
    **end if**
**end for**
```

3.5 Regret Analysis

In this section, we show that if the maximum feedback delay is finite, the cumulative regret sub-linearly increases with $\mathcal{O}(\sqrt{T \log T})$. Later, we show that sub-linearity when feedback is missing (infinite delay).

3.5.1 Finite Maximum Delay

Cumulative Regret Bound

Theorem 9 *Assume that the reward is normalized $y_{a,t} \in [0,1]$, kernel function is $k(x,x') \leq c_k$, $\|\theta\|_2 \leq 1$, d is the dimension of $\phi(x)$ and feedback delay $d_t \leq \tau_{\max}$. At each round, the agent has context $x_{a,t}, \forall a \in \mathcal{A}$ and selects arm by Algorithm 2. With probability $1-\delta$, $0 \leq \delta \leq 1$, the cumulative expected regret R_T is bounded by*

$$R_T \leq 2(\alpha + \lambda)\, \tau_{\max} \sqrt{2 \lfloor T/\tau_{\max} \rfloor d \log\Big(1 + \frac{c_k}{d\lambda} \lfloor T/\tau_{\max} \rfloor\Big)} + 4\tau_{\max}$$

where $\lfloor \cdot \rfloor$ is the floor function and $\alpha = \sqrt{\frac{1}{2} \ln \frac{2KT}{\delta}}$. ■

Theorem 9 shows that Algorithm 2 can achieve a sub-linear cumulative regret bound in the form of $\mathcal{O}(\tau_{\max}\sqrt{\lfloor T/\tau_{\max} \rfloor \log \lfloor T/\tau_{\max} \rfloor})$ for $\tau_{max} \geq 1$. This implies that as $T \to \infty$, the agent can eventually identify the optimal context-specific arms, resulting in a vanishing average regret. When $\tau_{max} = 1$ (i.e., the reward feedback for one round is received by the agent prior to the beginning of the next round without further delays), the cumulative regret bound of Algorithm 2 reduces to that in the standard no-delay bandit setting [34, 36].

Proof of Regret Bound

Because of delayed feedback in the considered bandit setting, the proof of our cumulative regret bound is more challenging than that of the traditional bandit and shown below. Since the feedback delay is less than or equal to $t_{\max}$, it is possible that no feedback is received during the first $t_{\max}$ rounds. If this happens, the agent chooses arm randomly for the first $t_{\max}$ rounds. In view of this, we divide the cumulative regret into two parts:

$$R_T = \sum_{t=1}^{\tau_{\max}} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right] + \sum_{t=\tau_{\max}+1}^{T} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right].$$

Define the starting regret as $R_T^s = \sum_{t=1}^{\tau_{\max}} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right]$ and the continuing regret as $R_T^c = \sum_{t=\tau_{\max}+1}^{T} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right]$. Clearly, the starting regret $R_T^s \leq 2\tau_{\max}$. Now the challenge is to bound the continuing regret part.

We first bound the instantaneous regret of UCB-based arm selection in Lemma 10 (proof in appendix).

Lemma 10 *If UCB is used for arm selection, then with probability at least* $1-\frac{\delta}{T}$, $0 \leq \delta \leq 1$, *for* $\forall a \in \mathcal{A}$

$$reg_t \leq 2\left(\alpha + \lambda\right) w_{a,t}.$$

Then, with probability at least $1-\delta$, $0 \leq \delta \leq 1$, the continuing regret can be bounded as

$$R_T^c = \sum_{t=\tau_{\max}+1}^{T} reg_t \leq 2\left(\alpha + \lambda\right) \sum_{t=\tau_{\max}+1}^{T} w_{a_t,t}.$$

Recall that $w_{a,t} = \sqrt{\frac{1}{\lambda}(k(x_{a,t}, x_{a,t}) - \mathbf{k}_{a,t}{}^{\top}(\mathbf{D}_t)^{-1}\mathbf{k}_{a,t})}$ and its RKHS representation is in the form as $w_{a,t} = \sqrt{\phi(x_{a,t})^{\top}(\mathbf{C}_t)^{-1}\phi(x_{a,t})}$. Since C_t only contains a part of history contexts before round t due to varying feedback delays, we cannot directly bound $\sum_{t=\tau_{\max}+1}^{T} w_{a_t,t}$ as Lemma 11 in [1] or Lemma 7 in [36]. In order to prove the bound of $\sum_{t=\tau_{\max}+1}^{T} w_{a_t,t}$, we assume the total round T is an integer multiple of $\tau_{\max}$ without loss of generalization, i.e. $T = (1+m)\tau_{\max}$. If an upper bound can be proved under this assumption, it can be generalized to general total number of bounds T. In the next lemma, we divide the $T - \tau_{\max}$

rounds into τ_{max} groups, each with m elements and prove the bound of $\sum_{t=\tau_{\max}+1}^{T} w_{a_t,t}$ by bounding the sum of $w_{a_t,t}$ in each group.

Lemma 11 *If* $T = (m+1)\tau_{max}$*, we have*

$$\sum_{t=\tau_{max}+1}^{T} ||\phi(x_{a,t})||^2_{\mathbf{C}_t^{-1}} \leq 2\tau_{max} d \log\left(1 + \frac{mc_k}{d\lambda}\right)$$

The proof of Lemma 11 is included in the supplementary material. We can now prove Theorem 9. **Proof.** If $T = (m+1)\tau_{\max}$, $m \in \mathbb{Z}^+$, by Lemma 10 and Lemma 11, we have

$$\begin{aligned} R_T^c &\leq 2\left(\alpha + \lambda\right) \sum_{t=\tau_{\max}+1}^{T} w_{a_t,t} \\ &\leq 2\left(\alpha + \lambda\right) \sqrt{\left(T - \tau_{max}\right) \sum_{t=\tau_{max}+1}^{T} w_{a_t,t}^2} \\ &\leq 2\left(\alpha + \lambda\right) \sqrt{T 2\tau_{max} d \log\left(1 + \frac{mc_k}{d\lambda}\right)} \end{aligned}$$

Therefore, for $T = (\xi + 1)\tau_{max}$, the cumulative regret is bounded by

$$\begin{aligned} R_T &= R_T^c + R_T^s \\ &\leq 2\left(\alpha + \lambda\right) \sqrt{2T\tau_{max} d \log(1 + \frac{c_k \xi}{d\lambda})} + 2\tau_{max} \end{aligned}$$

If $T \geq \tau_{\max}$ and is not divisible by $\tau_{\max}$, we define $t_r = T - \lfloor T/\tau_{\max} \rfloor \tau_{\max} + \tau_{\max}$ and divide the cumulative regret

$$R_T = \sum_{t=1}^{t_r} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right] + \sum_{t=t_r+1}^{T} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right].$$

Note that $T - t_r \geq \tau_{\max}$ and is divisible by $\tau_{\max}$. Therefore, by using Lemma 10 and

Lemma 11 for the second term, we have

$$\begin{aligned}
&\sum_{t=t_r+1}^{T} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right] \\
\leq& 2\left(\alpha+\lambda\right) \sum_{t=t_r+1}^{T} w_{a_t,t}. \\
\leq& 2\left(\alpha+\lambda\right) \sqrt{\left(T-t_r\right) \sum_{t=t_r+1}^{T} w_{a_t,t}^2} \\
\leq& 2\left(\alpha+\lambda\right) \sqrt{2\left\lfloor T/\tau_{\max}\right\rfloor \tau_{\max}^2 d \log\left(1+\frac{c_k}{d\lambda}\left\lfloor T/\tau_{\max}\right\rfloor\right)}
\end{aligned}$$

Since the first term $\sum_{t=1}^{t_r} \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right] \leq 2t_r \leq 4\tau_{\max}$, the cumulative regret is bounded as

$$\begin{aligned}
R_T \leq& 2\left(\alpha+\lambda\right)\tau_{\max}\sqrt{2\left\lfloor T/\tau_{\max}\right\rfloor d \log\left(1+\frac{c_k}{d\lambda}\left\lfloor T/\tau_{\max}\right\rfloor\right)} \\
&+ 4\tau_{\max},
\end{aligned}$$

thus completing the proof. ■

3.5.2 Missing Feedback

Feedback delays may be arbitrarily long (i.e., missing). We denote $\mathcal{S}_T^r$ as context set with received or delayed feedback and context set $\mathcal{S}_T^m$ for missing feedback up to time T. Under this setting, the cumulative regret up to time T can be represented as

$$R_T = R(\mathcal{S}_T^r) + R(\mathcal{S}_T^m)$$

Although context in $\mathcal{S}_T^r$ may have delayed feedback, according to Theorem 9 the cumulative regret caused by $\mathcal{S}_T^r$ is still bounded sub-linearly. We denote $x_{a,\psi} \in \mathcal{S}_T^m$ as context with missing feedback at time step ψ .By using Lemma 8, the estimation error bound at time ψ is

$$\left|\hat{g}_{a,\psi} - g_{a,\psi}\right| \leq \left(\alpha+\lambda\right) w_{a,\psi}$$

where $\alpha = \sqrt{\frac{1}{2}\ln\frac{2KT}{\delta}}$ and $w_{a,\psi} = \sqrt{\phi(x_{a,\psi})^\top(\mathbf{C}_\psi)^{-1}\phi(x_{a,\psi})}$. Then, according to Lemma 10 the cumulative regret by $\mathcal{S}_T^m$ is bounded by

$$R(\mathcal{S}_T^m) \leq 2\,(\alpha + \lambda)\sum_{\psi} w_{a,\psi} \tag{3.10}$$

Clearly, contexts in $\mathcal{S}_T^m$ do not contribute to reward estimation in Eqn. (3.7) because of lacking feedback information. Therefore, $\phi(x_{a,\psi})$ is not appended in to $\mathbf{\Phi}_\psi$. From Lemma 11, the sub-linearity of $\sum_\psi w_{a,\psi}$ is not guaranteed and further, sub-linear upper bound of regret by Algorithm 2 with missing feedback may not be attained.

3.6 UCB With Semi-supervised Learning

In this section, we exploit the available context information and leverage the idea of semi-supervised learning [28] to improve the agent's estimation of reward functions. Specifically, while waiting for the delayed or possibly missing feedbacks, the agent knows the context and selected arms for these feedbacks, and hence it can obtain fictitious estimates of the delayed or missing feedbacks based on the feedback history observed so far. Thus, the available context information and selected arms can potentially contribute to the overall learning process, as shown in other contexts (e.g., image classification) using semi-supervised learning [28]. Next, we will consider one different way to exploit the available context information to obtain fictitious estimates of the delayed and missing rewards: minimizing the estimation error by similar context (MINSIM).

3.6.1 Minimizing Estimation Error by Similar Context

Algorithm 3 Extension of Delayed Contextual UCB

```
Inputs : kernel function $k$ and parameter $\alpha$, $\lambda$ and $\kappa_t$.
for $t = 1, \cdots, T$ do
  if length($\hat{\mathbf{y}}_t = 0$) then
    Random choose arm $a_t$
  else
    if $|\mathcal{S}_t| \neq 0$ then
      For $\tau_i \in \mathcal{S}_t$, augment $y_{a_{\tau_i}, \tau_i}$ into $\hat{\mathbf{y}}_t$
      Move $\phi(x_{a,\tau_i})$ from $\tilde{\mathbf{\Phi}}_t$ to $\hat{\mathbf{\Phi}}_t$.
    end if
    if $|\tilde{\mathbf{\Phi}}_t| \neq 0$ then
      Find fictitious reward $\tilde{\mathbf{y}}_t$ by Eqn. (3.11)
    end if
    Receive context $x_{a,t}$, $a = 1, \cdots, K$
    Augment $\phi(x_{a,t})$ into $\tilde{\mathbf{\Phi}}_t$
    for $a \in \mathcal{A}$ do
      MINSIM: Calculate $\bar{g}_{a,t}$ by Eqn. (3.14) and $\bar{w}_{a,t}$ according to Eqn. (3.15).
    end for
    MINSIM: $a_t = \arg\max_{a \in \mathcal{A}} \bar{g}_{a,t} + (\alpha + \lambda)\bar{w}_{a,t}$
  end if
end for
```

Based on the context and reward history, the agent can find the similar context with received feedbacks for those with delayed or missing feedbacks. Thus, we can view the reward of similar context as the "(pseudo) true" reward for the delayed and missing rewards, and apply ridge regression learning over the perturbation.

First, we use $\tilde{\mathbf{\Phi}}_t$ to store contexts without feedback yet (i.e., the contexts whose corresponding rewards are delayed or missing, and have yet to be provided to the agent), $\hat{\mathbf{\Phi}}_t$ to store contexts whose reward feedbacks have arrived and the corresponding rewards are $\hat{\mathbf{y}}_t$. Thus, we can use $\bar{\mathbf{\Phi}}_t$ to represent all the experienced contexts up to the beginning of round t, such that $\bar{\mathbf{\Phi}}_t^\top = [\tilde{\mathbf{\Phi}}_t^\top, \hat{\mathbf{\Phi}}_t^\top]$. Once a delayed reward feedback is provided, it will be appended to $\hat{\mathbf{y}}_t$ and its corresponding context information will be transferred from $\tilde{\mathbf{\Phi}}_t$ to $\hat{\mathbf{\Phi}}_t$. We denote $\tilde{\mathbf{y}}_t$ as the reward feedbacks that have not arrived and need to be estimated with a perturbation.

As we have all the experienced context information, we can find perturbed reward of each element in $\tilde{\mathbf{y}}_t$ based on the previous context in $\hat{\mathbf{\Phi}}_t$ and corresponding feedback in $\hat{\mathbf{y}}_t$. Specifically, giving a time-varying perturbation threshold κ_t of context, we assume context x_a^i in $\tilde{\mathbf{\Phi}}_t$ and the context x_a^k in $\hat{\mathbf{\Phi}}_t$ meet the threshold condition.

$$||x_a^i - x_a^k|| \leq \kappa_t \tag{3.11}$$

By using the Hölder condition in Assumption 1, the expectation of $\tilde{y}_t^i$ in the vector $\tilde{\mathbf{y}}_t$ can be bounded by the expectation of $\hat{y}_t^k$ as follows:

$$\left|\mathbb{E}\left[\tilde{y}_t^i\right] - \mathbb{E}\left[\hat{y}_t^k\right]\right| \leq L||x_a^i - x_a^k||^\beta \leq L{\kappa_t}^\beta \tag{3.12}$$

Therefore, we utilize the true reward feedback $\tilde{y}_t^i$ for context x_a^i. In this way, by finding fictitious rewards into $\tilde{\mathbf{y}}_t$, all contexts in $\hat{\mathbf{\Phi}}_t$ are also "labeled" with feedbacks. However, one challenge for finding fictitious rewards is to guarantee that there always exists a context in $\hat{\mathbf{\Phi}}_t$ that satisfies the κ_t threshold for every context in $\tilde{\mathbf{\Phi}}_t$. To achieve this, we set κ_t as a time-decreasing parameter, which allows more coarse fictitious rewards to alleviate the shortage of contexts at the beginning, and produce more accurate fictitious reward to improve learning afterward.

After we find fictitious rewards in $\tilde{\mathbf{y}}_{\mathbf{t}}$ with context threshold κ_t, the next step is to learn the reward function in a semi-supervised manner given the perturbed rewards shown in Eqn. (3.12). Specifically, by using kernel-based empirical risk minimization to estimate $\bar{\theta}_t$, we solve the following optimization problem:

$$\bar{\theta}_t = \arg\min_{\bar{\theta}} \left\{ \|\tilde{\mathbf{y}}_t - (\tilde{\mathbf{\Phi}}_t)^\top \tilde{\theta}\|^2 + \|\hat{\mathbf{y}}_t - (\hat{\mathbf{\Phi}}_t)^\top \bar{\theta}\|^2 + \lambda \|\bar{\theta}\|^2 \right\} \tag{3.13}$$

Without the first term, the problem in Eqn. (3.13) reduces to the kernel-based ridge regression in Algorithm 2, where only contexts with feedbacks are utilized. By taking $\tilde{\mathbf{y}}_t$ as perturbation vector for delayed and/or missing rewards with the corresponding contexts $\tilde{\mathbf{\Phi}}_t$ and minimizing the estimation error over $\tilde{\mathbf{y}}_t$, we can estimate $\tilde{\theta}_t$ in a semi-supervised manner. Finding a closed-form solution $\bar{\theta}_t$ by solving the problem in Eqn. (3.13) is same as solving Eqn. (3.5). Based on [36], we solve Eqn. (3.13) and obtain the following solution

$$\bar{\theta}_t = \bar{\mathbf{C}}_t^{-1} \bar{\mathbf{\Phi}}_t \bar{\mathbf{y}}_t$$

where $\bar{\mathbf{y}}_t = [\tilde{\mathbf{y}}_t, \hat{\mathbf{y}}_t]$ and $\tilde{\mathbf{C}}_t = \bar{\mathbf{\Phi}}_t \bar{\mathbf{\Phi}}_t^\top + \lambda \mathbf{I}$. Accordingly, at time t, the estimated reward $\bar{g}_{a,t} = \phi(x_{a,t})^\top \bar{\theta}_t^i$ is

$$\bar{g}_{a,t} = \phi(x_{a,t})^\top \bar{\mathbf{C}}_t^{-1} \bar{\mathbf{\Phi}}_t \bar{\mathbf{y}}_t \tag{3.14}$$

Like in Algorithm 2 for arm selection, we also add an exploration term of confidence width $\bar{w}_{a,t}$ into the estimated reward:

$$\bar{w}_{a,t} = \sqrt{\phi(x_{a,t})^\top (\bar{\mathbf{C}}_t)^{-1} \phi(x_{a,t})} \tag{3.15}$$

Thus, based on ridge regression and UCB for arm selection, our MINSIM approach for extending the delayed contextual UCB is described in Algorithm 3.

3.6.2 Regret Analysis

To show the advantage of Algorithm 3 over Algorithm 2 given delayed and missing feedbacks, we show that the cumulative regret bound of MINSIM is sub-linear with time. To prove the sub-linear bound, we firstly bound the estimation error of Algorithm 3 as follow.

Lemma 12 *Suppose that the expected reward $g_{a,t}$ belongs to the RKHS generated by kernel function k, i.e. $g_{a,t} = \phi\left(x_{a,t}\right)^\top \theta$. If Eqn.* (3.14) *is used for reward estimation and entry in vector $\phi(x_{a,t})^\top \bar{\mathbf{C}}_t^{-1} \bar{\mathbf{\Phi}}_t$ is bounded by $|V_{max}|$, then with probability at least $1 - \frac{\delta}{T}$, the estimation error is*

$$|\bar{g}_{a,t} - g_{a,t}| \leq (\alpha + \lambda)\, \bar{w}_{a,t} + L|V_{max}| t {\kappa_t}^\beta \tag{3.16}$$

where $\bar{w}_{a,t} = \sqrt{\phi(x_{a,t})^\top (\bar{\mathbf{C}}_t)^{-1} \phi(x_{a,t})}$ and $\alpha = \sqrt{\frac{1}{2} \ln \frac{2KT}{\delta}}$. ■

The proof of Lemma 12 is available in the supplementary material. By using the fictitious reward vector $\tilde{\mathbf{y}}_\mathbf{t}$, the term $L|V_{max}| t {\kappa_t}^\beta$ in Eqn. (3.16) is from the contexts with missing and delayed feedbacks. From the conclusion in Lemma 10, with probability $1 - \frac{\delta}{T}$, the instant regret is bounded as

$$reg_t \leq 2(\alpha + \lambda)\bar{w}_{a,t} + 2L|V_{max}| t {\kappa_t}^\beta$$

When we sum up all instant regrets over time, $\sum_{t=1}^{T} \bar{w}_{a,t}$ can be proven with $\mathcal{O}(\sqrt{T \log(T)})$ by using Lemma 11 in the non-delay setting ($\tau_{\max} = 1$). Therefore, the choice of κ_t is vital to achieve sub-linear upper bound of cumulative regret reward.

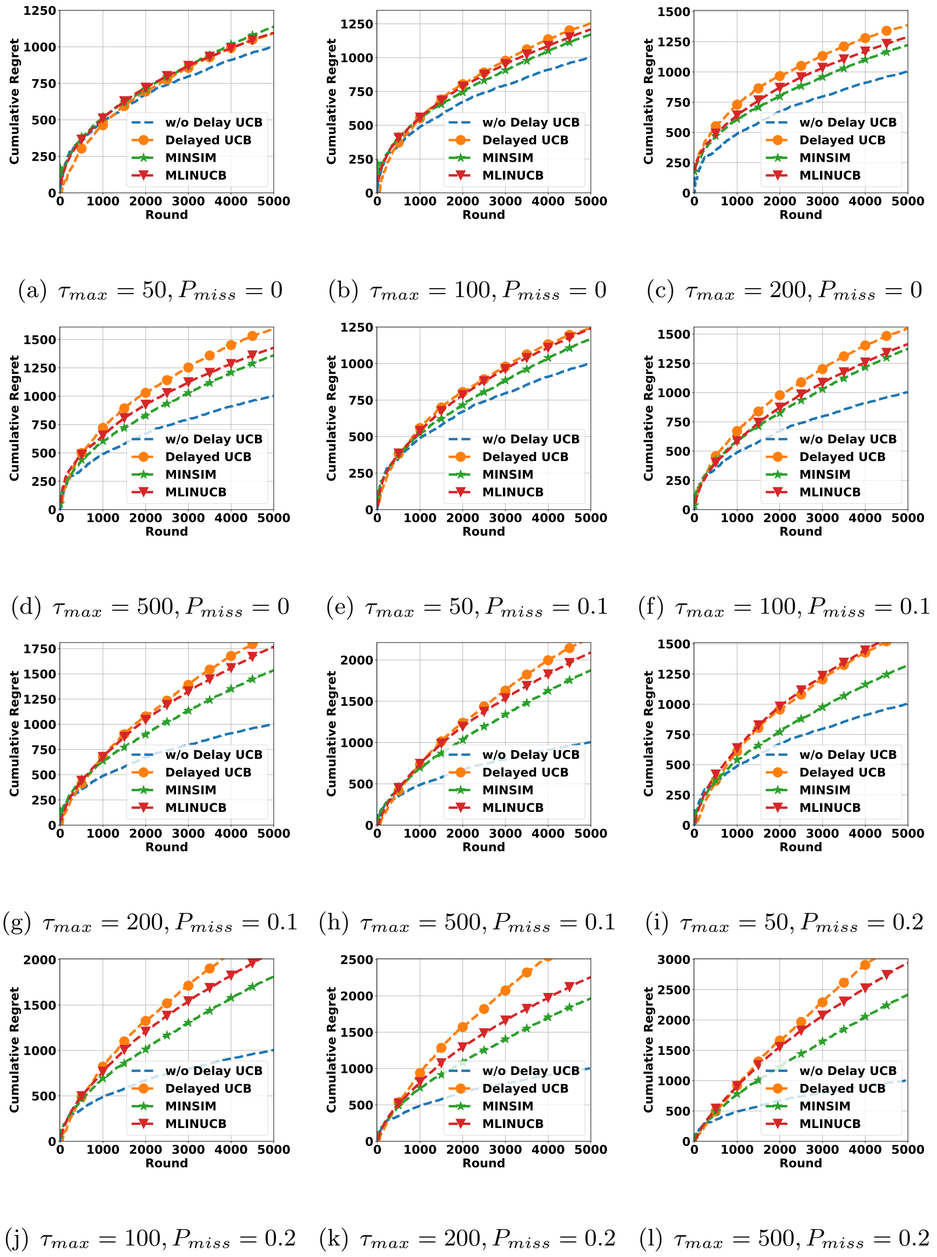
Theorem 13 *Assume that the reward is normalized as $y_{a,t} \in [0,1]$, kernel function is $k(x,x') \leq c_k$, $\|\theta\|_2 \leq 1$, d is the effective dimension of $\phi(x)$, entry in vector $\phi(x_{a,t})^\top \bar{\mathbf{C}}_t^{-1} \bar{\mathbf{\Phi}}_t$ is bounded by $|V_{max}|$ and let $\kappa_t = \zeta t^{\frac{-3}{\beta}}$, where $\zeta > 1$. At each round, the agent has context $x_{a,t}, \forall a \in \mathcal{A}$ and selects arm by Algorithm 3. With probability $1-\delta$, $0 \leq \delta \leq 1$, the cumulative expected regret R_T is bounded by*

$$R_T \leq 2\left(\alpha + \lambda\right) \sqrt{2Td \log\left(1 + \frac{c_k}{d\lambda} T\right)} + 2L|V_{max}|\zeta \frac{\pi^2}{3}$$

where $\frac{\pi^2}{3} = 2\sum_{t=1}^{\infty} t^{-2}$and $\alpha = \sqrt{\frac{1}{2} \ln \frac{2KT}{\delta}}$. ■

We set $\kappa_t = \zeta t^{\frac{-3}{\beta}}$ to guarantee the existence of fictitious feedback when contexts with feedback are scarce, and also the reward perturbation is decreasing with time. This method is different from widely-used self-training techniques [28] which provides fictitious feedback through the already learned model. The reason is that in RKHS, our reward function can be viewed linear with gaussian noise. Thus, it is futile to estimate the reward by utilizing the learned function to generate fictitious feedback for delayed and missing rewards.

To sum up, by using Algorithm 3, the agent takes advantage of more information from the context to produce fictitious rewards for further updating reward estimation, although the actual reward signals are still delayed or missing. This is expected to decrease the regret compared to Algorithm 2 when delayed and missing feedbacks are significant.



(a) $\tau_{max} = 50, P_{miss} = 0$ (b) $\tau_{max} = 100, P_{miss} = 0$ (c) $\tau_{max} = 200, P_{miss} = 0$

(d) $\tau_{max} = 500, P_{miss} = 0$ (e) $\tau_{max} = 50, P_{miss} = 0.1$ (f) $\tau_{max} = 100, P_{miss} = 0.1$

(g) $\tau_{max} = 200, P_{miss} = 0.1$ (h) $\tau_{max} = 500, P_{miss} = 0.1$ (i) $\tau_{max} = 50, P_{miss} = 0.2$

(j) $\tau_{max} = 100, P_{miss} = 0.2$ (k) $\tau_{max} = 200, P_{miss} = 0.2$ (l) $\tau_{max} = 500, P_{miss} = 0.2$

Figure 3.1: Cumulative regrets for different algorithms.

3.7 Experiments and Results

In this section, to evaluate our proposed delayed algorithms, we consider the application of contextual news recommendation to improve click through rate (CTR).

3.7.1 Yahoo! Today Module

To test our proposed algorithms, we use the Yahoo Module dataset [65]. It includes the browsing history from Yahoo! Front Page Module, where the featured tab recommends one article from a large set of candidates. The dataset is a log of random traffic on the Today Module, and ensures that the article is randomly chosen to serve the user. The dataset records detailed user/article features for context-aware problem. The feedback in dataset is click information (click or not click), which indicates user's preferences for recommended articles. Each data entry is in the form of <user, article, click feedback>.

Delayed and missed click feedback of recommended articles or news is very common because the article location may not be noticeable to users, or users decide not to react immediately even they see the recommended item and spend additional time before providing any feedback. Additionally, since the user features (e.g. sex and age) as well as article features (e.g. topic) are highly heterogeneous, it is challenging to build a context-aware article recommendation in the delayed and missing feedback scenario. This problem fits well into our bandit setting — articles are the arms to select, and the clicks are feedbacks be provided to the learner and can be delayed or missing .

3.7.2 Experimental Setup

In pre-processing, we remove incomplete entries (user or arm context is not recorded) in the dataset and group user contexts after selecting 20 candidate articles. Thus, the size of our arm set is $|\mathcal{A}| = 20$. Different from previous work [65], we consider user-article context vector by concatenating vectors (both dimension of 6) instead of outer product, thus $x_{a,t} \in \mathcal{R}^{12}$. We assume the click feedback (0 or 1) for each user-article pair follows a Bernoulli distribution and its probability being clicked is the expected reward. Because the expected reward is unknown, it is impossible to evaluate the expected regret. To solve that, we utilize a deep neural network (DNN) to learn the expected reward function in an offline manner. The built DNN has total 4 layers and uses Relu as activation function. Specifically, given the user-article vector input, we consider the single normalized output inferred by the well-trained network is probability of article being clicked by user (expected reward).

During the online bandit learning, the learner is estimating reward function based on 1 (click) or 0 (not click) feedback and meanwhile, selecting arm with highest UCB score of probability being clicked. Some feedbacks are delayed, following a uniform distribution in $[0, \tau_{max}]$. Some feedbacks are assumed to have infinite delay (missing) with probability P_{miss}. We test four different values of τ_{max} and two different P_{miss} values. We use RBF kernel function, $k(x, x') = \exp(-\rho\|x - x'\|_2^2)$, for our kernel computation. To better evaluate our proposed algorithms, we compare them with two benchmarks: non-delay UCB (i.e., assuming $\tau_{max} = 1$ and $P_{miss} = 0$) and MLINUCB algorithm from [21].

3.7.3 Experimental Result

In Figure 3.1, we show the cumulative regrets for no-delay contextual UCB (w/o Delay UCB), delayed UCB algorithm that learns the reward function by only using available click feedback, MINSIM algorithms that finds fictitious values for delayed and missing feedback, and MLINUCB which utilizes clusters to predict feedback values.

In general, as the maximum feedback delay τ_{max} increases, the cumulative regret incurred by the algorithms also increase. However, when the occurrence of missing feedback is infrequent, all the cumulative regret curves exhibit a good sub-linearity as time goes on. Importantly, it is worth noting that the when τ_{max} is 50, the difference between the delayed contextual UCB and our semi-supervised algorithms is not significant, neither is the difference between our algorithms and the non-delayed UCB algorithm. This is expected, since a small feedback delay has a negligible role in affecting the bandit learning process. But, the regret difference between our algorithms and the non-delayed UCB algorithm becomes more significant as more feedbacks are likely missing. Moreover, the results show that by using Algorithm 2, we can effectively decrease the regret when the feedback delay is large enough and missing feedbacks are more frequent. This is because when the feedback information is more lacking, the learning process is more affected and thus, Algorithm 2 that finds fictitious feedbacks for delayed or missing feedback is more beneficial.

Chapter 4

Contextual Combinatorial Bandits with Arm Removal and Submodular Utility

4.1 Introduction

In real-world scenarios such as crowd sourcing [69], adaptive routing [12], wireless spectrum allocation [64] and influence maximization in social networks [97], the agent are provided with context information and can select a set of multiple arms, rather than an individual one, subject to a budget or cardinality constraint. Such problems fit well into contextual combinatorial bandits, which is a generalization of the contextual MAB model [84, 32]. As a result, the agent's overall utility is jointly determined by feedback signals of all the selected arms. Moreover, to capture the effect of diminishing returns, the utility function can be monotone and submodular in terms of the feedbacks of the selected arms [49, 30, 96]. The monotonicity and submodularity apply widely in practice, such as optimal sensor placement [59], sniffer channel assignment [29] and recommender systems [14]. Nonetheless, in general, maximizing a monotone submodular function is a challenging NP-hard problem, even when the feedback for each selected arm is known in advance under a non-learning setting.

In addition, another crucial consideration in contextual combinatorial bandits is that some selected arms may be deliberately or accidentally removed/nullified from decision arm set (which we call arm removal or failure). This occurs in a variety of scenarios [82]: (i) in influence maximization problems, some selected users may not continue spreading the information as requested; (ii) in the sensor placement problem, some selected sensors may fail due to internal malfunction or adversarial jamming attacks; and (iii) in a tactical environment, the selected nodes can be sabotaged and disabled for functioning. Consequently, the removed arms may not contribute anything to the overall utility. It is noteworthy that, unlike sleeping arms [88] or volatile arms [30], the removed arms are not removed from total arm set permanently or temporarily; instead, they are still available for selection but they will not contribute to the agent's actual utility. While the agent can learn the overall feedback for each arm by explicitly accounting for the likelihood of arm removal, a robust arm selection strategy is more suitable when the agent is uncertain a priori about which arms will be removed. That is, to achieve robustness, the agent focuses on the worst-case utility in the presence of arm removal. In the submodular maximization literature, robust optimization in the presence of removed actions is very challenging, with only approximation algorithms available to date [82]. Combined with unknown feedbacks for the selected arms that require online learning, robustness for submodular maximization is naturally even more challenging.

In this chapter, motivated by the above considerations, we study a novel and challenging contextual combinatorial bandit setting with arm removal and submodular utility: the agent can select multiple arms (subject to a cardinality constraint), but some selected arms may be removed and the overall utility is jointly determined by all the feedback signals of non-removed arms through a monotone submodular utility function. The challenges are two-fold. First, the agent has no knowledge a priori about the exact feedback for each selected arm. Thus, the agent must carefully balance exploitation and exploration, and learn the feedback signals for different arms and context. Second, even assuming that the agent has perfectly learnt the feedback signals for different arms and context, robust optimization of a monotone submodular utility in the presence of arm removal is NP-hard with only approximation algorithms available for general settings [82].

To address the challenges, we propose a novel approximation-based robust combinatorial bandit algorithm, called arm Removal Robust Contextual Combinatorial MAB (R2C2-MAB). The algorithm is designed building upon contextual combinatorial bandits, along with a greedy approximation algorithm [79] and a robust submodular maximization method [17]. Specifically, we consider a general non-parameterized (unknown) relation between the feedback signal/reward received by each arm and context, while the agent's overall utility is a monotone submodular function of all the feedback signals of non-removed arms. While learning the feedback signals online, the agent's goal is robust arm selection, such that the worst-case utility is maximized in the presence of arm removal. We prove that R2C2-MAB achieves a sublinear regret over time compared to the oracle that knows the exact feedback signals for all the arms and context information. To evaluate the efficiency of proposed R2C2-MAB algorithm, we consider a sniffer channel assignment problem in a tactical environment. This is shown to be a combinatorial problem, since it chooses a subset of sniffers to deploy on a channel to maximize the overall monitoring performance (monotone and submodular).

The main contributions of our work are summarized as follow:

(1) We study a novel and challenging problem — contextual combinatorial bandits with arm removal and submodular utility. This setting applies to a variety of problems in networked systems involving combinatorial decisions, especially in tactical or adversarial environments where some selected arms can be nullified and contribute zero to the agent's utility.

(2) We propose a new algorithm, called R2C2-MAB, that can robustly select arms to maximize the worst-case utility in the presence of arm removal and provably guarantees a sublinear cumulative regret with time.

(3) We consider the sniffer channel assignment problem and evaluate R2C2-MAB against a number of well-established bandit algorithms. The numerical result shows that R2C2-MAB outperforms the existing algorithms and validates our analysis.

4.2 Related Work

In the literature of combinatorial bandit, [84] considers contextual combinatorial bandit with semi-bandit feedback and nonlinear rewards, [66] focuses on contextual combinatorial bandits with cascading feedback and nonlinear reward, [64] considers fairness and sleeping arms, and [30] considers contextual combinatorial bandit with volatile arms. Our work differs from these studies in that we focus on worst-case robustness with arm removal. We note the crucial difference between our considered arm removal and sleeping or volatile arms: removed arms are those whose corresponding rewards are removed or nullified, whereas volatile arms mean that they are either temporarily or permanently unavailable.

Our work is also related to adversarial bandits or corrupted bandits [73, 80, 95, 8, 70, 10, 11, 54], many of which consider that the adversary maliciously presents rewards *observed* by the agent to mislead reward estimation and arm selection. A recent study [117] also considers that the adversarial can maliciously modify the actual reward received by the agent. While this setting can capture arm removal (i.e., setting the rewards for removed arms as zero), our consideration of contextual combinatorial bandits with a submodular utility function sets our work apart.

More recently, bandit algorithms with imperfect information have been studied. For example, some focus on robust reward estimation and exploration [8, 50], and others train a robust policy directly [112, 95]. The studies [107, 57, 116, 114] focus on imperfect contextual information, and subsequently [56, 115] consider robust arm selection in the presence of context errors. Nonetheless, our work assumes a different type of imperfection

— arm removal — than uncertain context.

Finally, our work is related to submodular function maximization [48, 63, 22]. For a monotone submodular function, $(1-1/e)$-approximation can be achieved by greedy approaches subject to a cardinality constraint [79]. The study [43] shows that the greedy algorithm achieves a 1/2-approximation for maximizing the same objective subject to a general matroid constraint, and the ratio is later improved to a tighter bound [24]. Considering action removal, robust submodular maximization is studied in [82, 17], where 0.387-approximation is proved by using greedy algorithm as subroutines. The submodular function is perfectly known in these studies, whereas we do not know the rewards associated with each selected arm and hence need learning while balancing exploitation and exploration.

4.3 Problem Formulation

In this section, we formulate the contextual combinatorial bandit problem with arm removal and submodular function. The decision timeline is discretized into time slots.

Consider a learning agent that interacts with the environment and makes sequential decisions for a horizon of T time slots. We let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of arms for selection (e.g., the set of users for scheduling in a wireless network). Due to the resource constraint, the agent can select up to b arms from $\mathcal{N}$, such that $b < N$ (the case of $b = N$ is trivial as the agent will simply choose all the arms due to monotone utility). The agent can observe side information (context) for each arm $x_n^t \in \mathcal{X} \triangleq [0, 1]^D, \forall n \in \mathcal{N}$, where $\mathcal{X}$ is the context space and D is the dimension of context vector. Without loss of generality, we normalize the context space within $[0, 1]^D$ as in the literature [30]. The agent collects the observed context information of all arms in $\mathbf{x}^t = \{x_n^t\}_{n\in\mathcal{N}}$.

In each time slot t, the agent selects a set of arms $\mathcal{N}_s^t$ with $|\mathcal{N}_s^t| \leq b$ based on the context information $\mathbf{x}^t$ and the knowledge about feedback signal corresponding to each arm learnt from the previous time slots. Due to existence of adversaries or arm failures, some selected arm may be removed or nullified, denoted as $\mathcal{N}_r^t \subseteq \mathcal{N}_s^t$. We let size $|\mathcal{N}_r^t| \leq \tau$, where τ is the upper bound on the number of removal arms out of the selected arm set. Generally, we have $\tau < b$ and more details on τ will be discussed later.

At the end of the each slot, the set of selected but non-removed arms is $\mathcal{N}_{nr}^t = \mathcal{N}_s^t \backslash \mathcal{N}_r^t$. For each arm $n \in \mathcal{N}_{nr}^t$, the agent receives a feedback signal d_n^t based on its context information. We also say that d_n^t is the feedback reward to be consistent with the bandit literature, while noting that the agent's overall utility is a function of all the non-removed feedback signals/rewards. We denote $\mathbf{d}_{nr}^t = \{d_n^t\}_{n\in\mathcal{N}_{nr}^t}$ as the collections of feedback signals/rewards obtained by selecting arms $\mathcal{N}_s^t$. In contextual bandits, it is commonly assumed that the reward function is, for example, linear in the context (plus a observation noise term) [35]. By contrast, in our work, the reward $d_n^t(x_n^t)$ corresponding to a non-removed arm $n \in \mathcal{N}_{nr}^t$ is determined by the context information x_n^t through an *unknown* and *non*-parameterized function subject to the Hölder condition (specified in Section 4.4.2) [30]. We assume the that reward d_n^t is bounded by $[0, d_{max}]$, where d_{max} is the maximum reward of taking a single arm in one time slot.

To evaluate the decision for selecting arms $\mathcal{N}_s^t$, the agent has a known *utility* function $u(\mathbf{d}_{nr}^t, \mathcal{N}_{nr}^t)$, as assumed in the literature [30, 29]. That is, the agent's online learning is mainly focused on the individual feedback signal/reward $d_n^t(x_n^t)$. To make our analysis tractable while being applicable to practical applications, we assume that the utility function is monotone and submodular, which includes the summation over the reward of each arm as a special case. Formally, monotonicity and submodularity are defined as follows.

Definition 14 (Monotonicity) *A set function $F : 2^V \rightarrow \mathbf{R}_+$ is defined to be monotone if $F(A) \leq F(B)$ for all $A \subseteq B \subseteq V$.*

Definition 15 (Submodularity) *A set function $F : 2^V \rightarrow \mathbf{R}_+$ is submodular if it admits the property of diminishing marginal cost:*

$$F(A \cup \{e\}) - F(A) \geq F(B \cup \{e\}) - F(B) \tag{4.1}$$

for all $A \subseteq B \subseteq V$ and $e \in V \backslash B$.

The goal of the agent is to select up to b arms to maximize the utility with possible arm removal of up to τ in a finite time horizon T. Thus, the bandit problem in the presence of arm removal can be formally written as:

$$\begin{aligned} \mathbf{P1}: \quad & \max_{(\mathcal{N}_s^t)_{t=1,\dots,T}} \sum_{t=1}^{T} \mathbb{E}\left[u(\mathbf{d}_{nr}^t, \mathcal{N}_{nr}^t)\right] \\ s.t. \quad & |\mathcal{N}_s^t| \leq b, b-\tau \leq |\mathcal{N}_{nr}^t| \leq b, \mathcal{N}_{nr}^t \subseteq \mathcal{N}_s^t \subset \mathcal{N}, \forall t \end{aligned} \tag{4.2}$$

where we keep the expectation over random contexts. Note that the key challenges for solving the problem **P1** are: (i) the agent does not known the function $d_n^t(x_n^t)$ and needs to learnt it online; and (ii) robust submodular maximization in the presence of arbitrary arm removal (i.e., any subset of selected arms can be removed subject to τ in total) [82].

4.4 Robust Contextual Combinatorial Bandit with Arm Removal

In this section, we first show the problem hardness and then propose an online learning algorithm, called R2C2-MAB. Our algorithm learns the feedback reward function given context information and robustly selects arms to maximize a submodular utility subject to arm removal.

4.4.1 Problem Hardness

Before proposing R2C2-MAB, we first show the problem hardness by assuming that the agent already perfectly knows the feedback reward functions for different arms and context information. That is, the online learning problem reduces to robust submodular optimization with action/arm removal. We describe two existing algorithms, which are instrumental for the development of R2C2-MAB.

Greedy Algorithm Without Arm Removal

We begin with a simplified case where there is no arm removal (i.e., $\tau = 0$ and $\mathcal{N}_{nr}^t = \mathcal{N}_s^t$) from selected arm set $\mathcal{N}_s^t$ and the feedback rewards $\boldsymbol{d}^t$ for all arms are perfectly known. Since the problem **P1** can be decoupled into T subproblems, the agent decides the best arm set based on $\mathcal{N}_s^t = \arg\max_{(\mathcal{N}_s^t)} u(\boldsymbol{d}_s^t, \mathcal{N}_s^t)$. Even under this idealized and simplified setting, solving **P1** is NP-hard due to its combinatorial nature. Instead, we consider a greedy approximation algorithm (described in Algorithm 4). The greedy algorithm selects arms sequentially to achieve the largest marginal utility increment. When arm $n \in \mathcal{N} \backslash \mathcal{N}_s^t$, the

Algorithm 4 Greedy Algorithm

```
**Inputs** : Arm set $\mathcal{N}$, utility function $u$ and budget $b$.
**for** $t = 1, 2, \ldots, T$ **do**
    **Initialization** : $\mathcal{N}_0 \leftarrow \emptyset, j \leftarrow 0$
    **while** $j \leq b$ **do**
        According to knowledge $\boldsymbol{d}^t$, select arm $n_j$ by
        $n_j = \arg\max_{n_j \in \mathcal{N} \setminus \mathcal{N}_{j-1}} \Delta_u(n_j, \mathcal{N}_{j-1})$
        $\mathcal{N}_j = \mathcal{N}_{j-1}\{\cup n_j\}$
    **end while**
    **Return** : $\mathcal{N}_s^t = \mathcal{N}_j$.
**end for**
```

marginal utility increment by adding n into $\mathcal{N}_s^t$ defined as $\Delta_u(n, \mathcal{N}_s^t) = u(\boldsymbol{d}_s^t \cup \{d(x_n^t)\}, \mathcal{N}_s^t \cup \{n\}) - u(\boldsymbol{d}_s^t, \mathcal{N}_s^t)$. Importantly, the greedy approximation algorithm guarantees to achieve no less than $\beta \in (0, 1]$ of the optimum, i.e., $u(\boldsymbol{d}_s^t, \mathcal{N}_s^t) \geq \beta u(\boldsymbol{d}_{s^*}^t, \mathcal{N}_{s^*}^t)$, where $u(\boldsymbol{d}_{s^*}^t, \mathcal{N}_{s^*}^t)$ is the utility of optimal arm set $\mathcal{N}_{s^*}^t$ by solving **P1**. The value of β depends on the structure of the combinatiorial optimization problem and varies with different settings. Notably, $\beta = 1 - \frac{1}{e}$ can be achieved for our monotone submodular maximization [42, 30].

Partitioned Robust With Arm Removal

In our setting, the agent has no prior knowledge about which arms will be removed from $\mathcal{N}_s^t$, and instead only knows the maximum number of removed arms τ. Therefore, in every time slot t, the agent has to select arms in a *robust* manner by considering τ arm removals from utility. From **P1**, we can reformulate the problem explicitly using its robust

Algorithm 5 Partitioned Robust Submodular optimization (PRO)

```
**Inputs** : Arm set $\mathcal{N}$, utility $u$, budget $b$, removal budget $\tau$, and greedy approximation $\mathcal{A}$.
**Initialization** : $S_0, S_1 \leftarrow \emptyset$
**for** $i = 0, \cdots, \lceil \log \tau \rceil$ **do**
    **for** $j = 1, \cdots, \lceil \frac{\tau}{2^i} \rceil$ **do**
        $B_j \leftarrow \mathcal{A}(\mathcal{N} \backslash S_0, u, 2^i)$
        $S_0 \leftarrow S_0 \cup B_j$
    **end for**
**end for**
$S_1 \leftarrow \mathcal{A}(\mathcal{N} \backslash S_0, u, b - |S_0|)$
**Return** : $S \leftarrow S_0 \cup S_1$
```

version as follows.

$$\begin{aligned} \mathbf{P2}: \qquad & \sum_{t=1}^{T} \max_{\mathcal{N}_s^t \subset \mathcal{N}} \min_{\mathcal{N}_r^t \subset \mathcal{N}_s^t} u(\boldsymbol{d}_s^t \backslash \boldsymbol{d}_r, \mathcal{N}_s^t \backslash \mathcal{N}_r^t) \\ & s.t. \qquad |\mathcal{N}_s^t| \leq b, |\mathcal{N}_r^t| \leq \tau, \forall t \end{aligned} \tag{4.3}$$

Given unknown arm removal, the greedy approximation algorithm can perform arbitrarily badly for **P2** in the worst case. The prior study [17] provides a new Partitioned Robust (PRO) submodular maximization algorithm (described in Algorithm 5) that achieves a constant-factor (0.387) approximation guarantee for solving **P2**.

The key idea of Algorithm 5 is that it utilizes greedy approximation algorithm (Algorithm 4) as a subroutine. The output of the algorithm is a set of size b that is robust against the worst-case removal of τ actions. As we can see in Algorithm 5, the output set consists of two sets S_0 and S_1: set S_0 is the robust part of the solution set S, which consists of $\lceil \log \tau \rceil + 1$ partitions. For every partition i, it consists of $\lceil \frac{\tau}{2^i} \rceil$ buckets B_j, and every bucket contains 2^i elements. The intuition of designing PRO is from fact that the objective value of the submodular function from buckets $i = 0, ..., j$ with removals cannot be too much smaller than the objective value in bucket j without removals, and the loss by the removals in bucket j is at most a small fraction of the objective value from previous buckets. Generally, the union of these buckets achieves a sufficiently high objective value, which is hence robust to removal. The size $|S_0|$ is related with τ, which can be represented as $|S_0| = \sum_{i=0}^{\lceil \log \tau \rceil} \lceil \frac{\tau}{2^i} \rceil 2^i$. However, without restrictions on τ, it is impossible to achieve a constant approximation ratio: for example, in the trivial case $\tau = b$ the agent cannot receive any utility at all. Thus, there is a limit on τ such that $|S_0| = \sum_{i=0}^{\lceil \log \tau \rceil} \lceil \frac{\tau}{2^i} \rceil 2^i \leq b$, as shown in [17].

4.4.2 Algorithm Design

In practice, the feedback rewards for different arms and context information are unknown to the agent a priori. Thus, at time slot t, the agent cannot directly use Algorithm 5 to get an approximation solution. Next, we propose R2C2-MAB to address this challenge.

To keep tractability, our algorithm is based on the assumption that taking the same arm on similar context information will result in similar reward. Mathematically, this assumption can be represented by the Hölder condition on reward $d(x_n^t)$ for arm n,

$$|d(x) - d(x')| \leq L||x - x'||^{\alpha} \tag{4.4}$$

where $L > 0$ and $\alpha > 0$. R2C2-MAB uniformly partitions the context space $\mathcal{X}$. By partition, we split the entire context space into small hypercubes of similar contexts. Then, the feedback reward in each hypercube is bounded due to Hölder condition. Similar to other bandit algorithms, our algorithm is interspersed with exploration and exploitation.

For exploration, the agent randomly selects a set of arms. During exploration, the agent learns the reward functions for arms which have not been explored sufficiently. Otherwise, the agent turns to exploitation. Because of the removal of selected arms, the agent needs to chooses arms in a robust manner, maximizing the worst-case utility based on previous history with similar contexts. After choosing a set of arms, the agent observes the reward of non-removal arms at the end of each time slot. In this way, the agent learns context-specific reward functions over time.

The pseudo-code of R2C2-MAB is presented in Algorithm 6. In the beginning, given context space $\mathcal{X} = [0,1]^D$, R2C2-MAB creates a partition $\mathcal{P}_T$ with time horizon T, which splits the context space evenly into $(h_T)^D$ sets, where h_T is an hyper-parameter which determines the number of hypercubes in the partition. The estimated reward of each hypercube $p \in \mathcal{P}_T$ can be computed by the accumulated historical rewards that fall into the hypercube. Letting $\mathcal{H}(p)$ represent the reward history of hypercube p, the estimated reward for contexts falling in p

$$\hat{d}(p) = \frac{1}{|\mathcal{H}^t(p)|} \sum_{d \in \mathcal{H}^t(p)} d \tag{4.5}$$

Additionally, the counter $C^t(p) = |\mathcal{H}(p)|$ indicates the number of times that the arms corresponding to contexts from hypercube p are selected and non-removed.

In each time slot t, the agent first observes the context x_n^t for all its arms. Then, R2C2-MAB determines the hypercube p_n^t containing the context x_n^t for arm n. The collection of these hypercubes set is denote as $P^t = \{p_n^t\}_{\forall n \in \mathcal{N}}$. As aforementioned, due to existence of potential arm removal, the arm selection considers maximizing the worst-case utility function. For the agent, to solve T subproblems in **P2**, we use Algorithm 5 with the estimated reward $\hat{d}(P^t)$ of all the arms.

Another challenge is that the proposed algorithm has to balance the exploration and exploitation in case there are hypercubes in P^t that have not been explored sufficiently. We denote $\mathcal{N}_{ue}^t$ as

$$\mathcal{N}_{ue}^t = \{n : \forall C^t(p_n^t) \leq K(t)\} \tag{4.6}$$

where $K(t)$ is a monotonically increasing in t and a hyperparameter decided by the agent (in Section 4.5). For every time slot t, if $\mathcal{N}_{ue}^t$ is not empty and $|\mathcal{N}_{ue}^t| \leq b$, we select all the arms in $\mathcal{N}_{ue}^t$ and the other $b - |\mathcal{N}_{ue}^t|$ arms by the greedy approximation algorithm (Algorithm 4). The reason that Algorithm 5 is not preferred for the rest $b - |\mathcal{N}_{ue}^t|$ arms, is to prevent the maximal removal τ or $|S_0|$ larger than $b - |\mathcal{N}_{ue}^t|$. If $\mathcal{N}_{ue}^t$ is empty, then the agent relies on its reward functions learnt thus far and employs Algorithm 5 to maximize the utility function in a robust manner.

4.5 Regret Analysis

In this section, we analyze R2C2-MAB described in Algorithm 6 by deriving an upper performance bound of the cumulative regret. Our key result shows that R2C2-MAB achieves a sublinear regret $O(\log(T)T^{\frac{\alpha+D}{2\alpha+D}})$ compared to the oracle that knows the reward functions for all arms and context information.

4.5.1 Regret Definition

To analyze regret of Algorithm 5, we consider an oracle, which knows the reward functions $d(x)$ but not which selected arms will be removed or not. In other words, the oracle also faces a robust submodular maximization problem whereas the agent in our setting needs online learning additionally. This type of oracle is similar to the one considered in robust bandits [56] with imperfect contexts where the oracle also does not know the perfect context.

For each arm n and each $p \in \mathcal{P}_t$, we define $\bar{d}(p) = \sup_{x\in p} d(p)$ and $\underline{d}(p) = \sup_{x\in p} d(p)$ as the highest and the lowest rewards for contexts in hypercube p. Since we need to compare the reward in different hypercubes, we set the geometric center of each hypercube p as the reference context, denoted as $x^*(p)$. Therefore, after receiving context P^t, the optimal b arm set $\mathcal{N}_s^*$ chosen by the oracle satisfies

$$\mathcal{N}_s^* = \underset{\mathcal{N}_s\in\mathcal{E}(\mathcal{N},b)}{\arg\max} \min_{\mathcal{N}_r\in\mathcal{E}(\mathcal{N}_s,\tau)} u(\boldsymbol{d}_s(x^*)\backslash\boldsymbol{d}_r(x^*), \mathcal{N}_s\backslash\mathcal{N}_r) \tag{4.7}$$

where $\mathcal{E}(\mathcal{N}, b)$ is the collection set of all b-arm subset from $\mathcal{N}$ and $\boldsymbol{d}_s(x^*) \overset{\Delta}{=} \{d(x^*(p_n^t))\}_{n\in\mathcal{N}_s}$. For simplicity, we let $g(\boldsymbol{d}_s(x^*), \mathcal{N}_s) \overset{\Delta}{=} \min_{\mathcal{N}_r\in\mathcal{E}(\mathcal{N}_s,\tau)} u(\boldsymbol{d}_s(x^*)\backslash\boldsymbol{d}_r(x^*), \mathcal{N}_s\backslash\mathcal{N}_r)$. Therefore, the regret $R(T)$ can be represented as

$$\mathbb{E}[R(T)] = 0.387 \cdot \sum_{t=1}^{T} \mathbb{E}\left[g(\boldsymbol{d}_s^*, \mathcal{N}_s^*)\right] - \sum_{t=1}^{T} \mathbb{E}\left[g(\boldsymbol{d}_s, \mathcal{N}_s)\right]. \tag{4.8}$$

The approximation ratio 0.387 comes from Algorithm 5 used by the oracle [17]. That is, due to the lack of polynomial time optimal solutions for robust submodular optimization [82, 17], the best approximation-guaranteed utility the oracle can achieve is $0.387\cdot\sum_{t=1}^{T} \mathbb{E}\left[g(\boldsymbol{d}_s^*, \mathcal{N}_s^*)\right]$. This scaled-down coefficient for regret analysis is also commonly used in other bandit settings with submodular utility [30].

4.5.2 Analysis

In this subsection, we show the details of the regret analysis for R2C2-MAB.

The regret $R(T)$ consists of two parts.

$$\mathbb{E}[R(T)] = \mathbb{E}[R_{explore}(T)] + \mathbb{E}[R_{exploit}(T)] \tag{4.9}$$

Since there are two phases — exploration and exploitation — in Algorithm 6, $\mathbb{E}[R_{explore}(T)]$ is the regret due to exploration for hypercubes and $\mathbb{E}[R_{exploit}(T)]$ is the regret by exploitation. Next, we bound $\mathbb{E}[R_{explore}(T)]$ and $\mathbb{E}[R_{exploit}(T)]$ separately.

Lemma 16 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$, the exploration regret $\mathbb{E}[R_{explore}(T)]$ is bounded by*

$$\mathbb{E}[R_{explore}(T)] \leq 0.387 \cdot bd_{max} 2^D (\log(T) T^{z+\gamma D} + T^{\gamma D}). \tag{4.10}$$

Proof. In the exploration phase, there exists at least one arm with context x_n^t, such that $C^t(p) \leq K(t) = t^z \log(t), \exists p \in \ P^t$. Since there are totally $(h_T)^D$ partitioned hypercubes, we have at most $(h_T)^D \lceil T^z \log(T) \rceil$ exploration phases. Since the utility function on is monotone and submodular given selected arms, the utility function of choosing b arms is upper bounded by bd_{max}. Then, the maximum regret of wrong selection in one exploration phase is bounded by $0.387bd_{max}$, due to the submodularity of utility function. For $\mathbb{E}[R_{explore}(T)]$, we have:

$$\begin{aligned}\mathbb{E}[R_{explore}(T)] &\leq 0.387 \cdot bd_{max}(h_T)^D \lceil T^z \log(T) \rceil \\ &= 0.387 \cdot bd_{max} \lceil T^\gamma \rceil^D \lceil T^z \log(T) \rceil \\ &\leq 0.387 \cdot bd_{max} 2^D (\log(T) T^{z+\gamma D} + T^{\gamma D})\end{aligned}$$

The last inequality is due to the fact that $\lceil T^\gamma \rceil^D \leq (2T^\gamma)^D = 2^D T^{\gamma D}$ and $\lceil T^z \log(T) \rceil \leq T^z \log(T) + 1$ This completes the proof. ■

The next step is to bound the regret $\mathbb{E}[R_{exploit}(T)]$ from Algorithm 6. Since we use the greedy approximation as a subroutine process included in Algorithm5, the exploitation regret is caused by suboptimal solutions from Algorithm 4. The reason that the greedy algorithm offers suboptimal solution is using the estimated reward $\hat{\boldsymbol{d}}^t$ to maximize marginal utility instead of $\boldsymbol{d}^t$. To bound the exploitation regret of Algorithm 6, we need to analyze the performance of greedy approximation by using the estimated reward of each arm. Because there are various greedy approximation subroutines problems with different budgets in Algorithm 5, we need to consider a general setting.

Lemma 17 *Given an arm set $\tilde{N}$ to select $\tilde{b}$ arms ($\tilde{b} \leq |\tilde{N}|$) in the subroutine of Algorithm6, the utility function $\tilde{u}$ is monotone, submodular, and satisfies $u(\underline{\boldsymbol{d}}^t_g, \tilde{\mathcal{N}}^t_g) - u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) \geq At^\theta$, where $\tilde{\mathcal{N}}^t_g$ is the optimal arm set by greedy approximation and $\tilde{\mathcal{N}}^t_{sub}$ is the suboptimal arm set. In time slot t, the probability of event V^t_{sub}, i.e., selecting $\tilde{\mathcal{N}}^t_{sub}$ over $\tilde{\mathcal{N}}^t_g$, is bounded by*

$$Pr\{V^t_{sub}\} \leq 2 \cdot \tilde{b} t^{-2} \tag{4.11}$$

Proof. We offer a sketch of proof, leaving the detailed proof in the appendix. In R2C2-MAB, when V^t_{sub} happens, it indicates that the utility of selecting arms in $\tilde{\mathcal{N}}^t_{sub}$ is higher than the utility of selecting arms in $\tilde{\mathcal{N}}_g^t$. Thus, we have

$$Pr\{V_{sub}^t\} = Pr\{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t)\} \tag{4.12}$$

The right side of Equation (4.12) indicates that at least one of three following events happens when $H(t) \geq 0$

$$\begin{aligned}
E_1 =& \{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) + H(t)\} \\
E_2 =& \{u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) \leq u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - H(t)\} \\
E_3 =& \{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t), \\
& \quad u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) < u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) + H(t), \\
& \quad u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) > u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - H(t)\}
\end{aligned} \tag{4.13}$$

Hence, we have

$$\{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t)\} \subseteq E_1 \cup E_2 \cup E_3 \tag{4.14}$$

The next step is to bound the probability of E_1, E_2 and E_3 separately. For $Pr\{E_1\}$, we have

$$\begin{aligned}
Pr\{E_1\} &= Pr\{u(\hat{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) \geq u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) + H(t)\} \\
&\leq Pr\{\hat{d}(p^t_n) \geq \bar{d}(p^t_n) + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\} \\
&\leq Pr\{\hat{d}(p^t_n) \geq \mathbb{E}[\hat{d}(p^t_n)] + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\} \\
&= \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} Pr\{\hat{d}(p^t_n) \geq \mathbb{E}[\hat{d}(p^t_n)] + \frac{H(t)}{\tilde{b}}\} \\
&\leq \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} \exp\left(\frac{-2C^t(\hat{p}^t_n)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} \exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \tilde{b}\exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right)
\end{aligned} \tag{4.15}$$

The first inequality of Equation (4.15) comes from $\{u(\hat{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) \geq u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) + H(t)\} \subseteq \{\hat{d}(p^t_n) \geq \bar{d}(p^t_n) + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\}$, which can be proved by *reductio ad absurdum*. The last three steps of Equation (4.15) utilize the Chernoff-Hoeffding bound. Since this is in the exploitation phase, there are least $t^z\log(t)$ times counted in $C(p), \forall p \in \mathcal{P}_t$. To bound $Pr\{E_1\}$, we choose $H(t) = \tilde{b}d_{max}t^{\frac{-z}{2}}$ and then have

$$\begin{aligned}
Pr\{E_1\} &\leq \tilde{b}\exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&= \tilde{b}\exp\left(\frac{-2t^z\log(t)(\tilde{b}d_{max}t^{\frac{-z}{2}})^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \tilde{b}\exp(-2\log(t)) \\
&\leq \tilde{b}t^{-2}
\end{aligned} \tag{4.16}$$

Similarly, the $Pr\{E_2\}$ can be bounded in the same way.

$$\begin{aligned} Pr\{E_2\} &\leq \tilde{b} \exp\left(\frac{-2t^z \log(t) H(t)^2}{(\tilde{b} d_{max})^2}\right) \\ &\leq \tilde{b} t^{-2} \end{aligned} \tag{4.17}$$

Last, to bound $Pr\{E_3\}$, we can find $H(t)$ that satisfies:

$$H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{At^{\theta}}{2} \tag{4.18}$$

Under the condition in Equation (4.18), the probability $Pr\{E_3\}$ is zero (details in the appendix).

Combining (4.16) and (4.17), we have

$$\begin{aligned} Pr\{V_{sub}^t\} &\leq Pr\{E_1 \cup E_2 \cup E_3\} \\ &\leq Pr\{E_1\} + Pr\{E_2\} + Pr\{E_3\} \\ &= 2 \cdot \tilde{b} t^{-2} \end{aligned} \tag{4.19}$$

■ The assumption of the utility difference $u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq At^{\theta}$ shows that the gap between the utility of the worst reward in $\tilde{\mathcal{N}}_g^*$ and the best reward for subset $\tilde{\mathcal{N}}'$ is shrinking as time grows, where $A > 0$ and $\theta < 0$ are the parameters for analysis.

After bounding the probability of selecting suboptimal arm sets in the greedy approximation, we can bound $\mathbb{E}[R_{exploit}(T)]$ by a constant.

Lemma 18 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^{\gamma} \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$. We assume $u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq At^{\theta}$ and condition $H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{At^{\theta}}{2}$ holds true, where $H(t) := \tilde{b}_{max} d_{max} t^{\frac{-z}{2}}$. For all $1 \leq t \leq T$, the regret due to exploitation $\mathbb{E}[R_{exploit}(T)]$ is bounded by*

$$\mathbb{E}[R_{exploit}(T)] \leq 0.387 \cdot b d_{max} \mathbb{S}(\tilde{b}_{max}, \tau) \tag{4.20}$$

where $\mathbb{S}(\tilde{b}_{max}, \tau) = \sum_{t=1}^{\infty} 1 - (1 - 2\tilde{b}_{max} t^{-2})^m$ *is a convergent series regarding the largest budget* b_{max} *in greedy subroutine and number of subroutine* $m = \sum_{i=0}^{\lceil \log \tau \rceil} i + 1$.

Proof. We let $U(t)$ represent the event that at least one greedy approximation $\mathcal{A}$ out of $m = \sum_{i=0}^{\lceil \log \tau \rceil} i + 1$ subroutines produces suboptimal arms. Then, the $R_m(T)$ can be represented as

$$\begin{aligned} R_{exploit}(T) &= \sum_{t=1}^{T} I_{\{U(t)\}} \times \Big(u(\boldsymbol{d}_g^t, \tilde{\mathcal{N}}_g^t) - u(\boldsymbol{d}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t)) \Big) \\ &\leq 0.387 \cdot bd_{max} \sum_{t=1}^{T} I_{\{U(t)\}} \end{aligned} \tag{4.21}$$

Taking expectation over contexts, we have

$$\begin{aligned} \mathbb{E}[R_{exploit}(T)] &\leq 0.387 \cdot bd_{max} \sum_{t=1}^{T} \mathbb{E}[I_{\{U(t)\}}] \\ &= 0.387 \cdot bd_{max} \sum_{t=1}^{T} Pr\{U(t)\} \end{aligned} \tag{4.22}$$

The next step is to bound $Pr\{U(t)\}$. Since we have the probability bound for the event that any greedy approximation selects suboptimal arms from Equation (4.19), it implies

$$Pr\{U(t)\} \leq 1 - (1 - 2\tilde{b}_{max} t^{-2})^m \tag{4.23}$$

where $\tilde{b}_{max}$ is the maximum budget of greedy approximation subroutines in Algorithm 5. Combining Equations (4.22) and (4.23), we have

$$\begin{aligned} \mathbb{E}[R_m(T)] &\leq 0.387 \cdot bd_{max} \sum_{t=1}^{T} Pr\{U(t)\} \\ &\leq 0.387 \cdot bd^{max} \sum_{t=1}^{T} 1 - (1 - 2\tilde{b}_{max} t^{-2})^m \\ &\leq 0.387 \cdot bd^{max} \sum_{t=1}^{\infty} 1 - (1 - 2\tilde{b}_{max} t^{-2})^m \\ &\leq 0.387 \cdot bd^{max} \mathbb{S}(\tilde{b}_{max}, \tau) \end{aligned} \tag{4.24}$$

where $\mathbb{S}(\tilde{b}_{max}, \tau) = \sum_{t=1}^{\infty} 1 - (1 - 2\tilde{b}_{max} t^{-2})^m$, which is convergent series. ■

Combining Lemma 16 and Lemma 18, we have a regret upper bound for $\mathbb{E}[R_T]$,

Theorem 19 *Let* $K(t) = t^z \log(t)$ *and* $h_T = \lceil T^\gamma \rceil$, *where* $0 < z < 1$ *and* $0 < \gamma < \frac{1}{D}$. *We assume* $u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq At^\theta$ *and condition* $H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{At^\theta}{2}$ *holds true, where* $H(t) := \tilde{b}_{max} d_{max} t^{\frac{-z}{2}}$. *For all* $1 \leq t \leq T$ *the total regret* $\mathbb{E}[R(T)]$ *defined in Equation* (4.8) *is upper bounded by*

$$\begin{aligned} \mathbb{E}[R(T)] \leq 0.387 \cdot (bd_{max} 2^D (\log(T) T^{z+\gamma D} + T^{\gamma D} \\ + (\gamma D + bd_{max} \mathbb{S}(\tilde{b}_{max}, \tau)) \end{aligned} \tag{4.25}$$

To bound the leading order of Equation (4.25) to be sublinear with time, we can choose parameters as follows: $z = \frac{\alpha}{2\alpha + D} \in (0, 1)$ and $\gamma = \frac{z}{\alpha} \in (0, \frac{1}{D})$. Then, the regret reduces to

$$\begin{aligned} \mathbb{E}[R(T)] \leq 0.387 \cdot \Big(bd_{max} 2^D (\log(T) T^{\frac{\alpha+D}{2\alpha+D}} + T^{\frac{D}{2\alpha+D}}) \\ + \cdot bd_{max} \mathbb{S}(\tilde{b}_{max}, \tau) \Big) \end{aligned} \tag{4.26}$$

Thus, the leading order of our regret bound is $O(\log(T) T^{\frac{\alpha+D}{2\alpha+D}})$.

4.6 Application to Sniffer Channel Assignment

In this section, we apply R2C2-MAB to a concrete application: sniffer channel assignment (SCA) for wireless channel monitoring.

4.6.1 Models

Consider a wireless channel passively monitored by a set of sniffers for purposes such as security and usage compliance [29]. To deal with potential security threats in the wireless network, there exists a passive monitoring system, which is independent of the wireless network, sensing channels and capturing packets. For forensics reason, the aim of monitoring system is to capture data packets of the targeted channels as many as possible. The problem is to decide which subset of sniffers should be allocated for channel monitoring given different channel conditions for the sniffers.

The time span is divided into time slots $t = 1, 2..., T$ in a slotted manner. The monitoring system consists of N data sniffers and a central agent for assignment decision making. We denote the set of sniffers by $\mathcal{N} \triangleq \{1, 2, ..., N\}$. Generally, we have a assignment budget b of sniffers, where $b < N$. Each sniffer is equipped with an antenna, which allows it to sense traffic over the wireless channel after being assigned. Besides traffic monitoring, there also exist channel inspectors that can periodically sense sniffer channel gain for different sniffers as context information [92]. In every time slot t, the assignment decision for sniffers $\mathcal{N}$ is denoted as $\mathbf{a}^t = \{a_1^t, a_2^t, \ldots, a_N^t\}$ and $a_n^t \in \{0, 1\}$, where 1 means assigned to the channel and 0 otherwise, subject to $\sum_{a \in \mathbf{a}^t} a \leq b$.

The feedback signal/reward for each sniffer is the probability of successfully identifying data packets on the assigned channel, also called capture probability denoted by Pr_n^t, which is defined as the probability of successfully capturing one data packet. The capture probability depends on the sniffer's monitoring channel condition. Specifically, according to the theory of channel secrecy capacity, a crucial factor determining the capture probability for a sniffer on the assigned channel is the signal-to-interference-plus-noise ratio (SINR). If sniffer n is selected, we let $SINR_n^t = \frac{PG_n^t}{I_n^t + N_0}$ be the context, where N_0 is the white noise power, P is the channel transmission power, G_n^t is the channel gain for the assigned sniffer, and I_n^t is the inference power. Further, in the channel, the data transmitted by target users can be precisely retrieved from by sniffer n (i.e., packet is successfully captured) only if the SINR for sniffer n is sufficiently good (i.e., $SINR_n^t \geq SINR_{th}$ where the threshold $SINR_{th}$ is unknown). Because the sniffer channel conditions can change, the capture probability Pr_n^t also varies over time. Thus, we define the capture probability Pr_n^t for sniffer n as $Pr_n^t = Pr\{SINR_n^t \geq SINR_{th}\}$, which is unknown a priori to the agent. Note that while each wireless transmission frame only lasts around 100ms, each sniffer assignment are held constant for a much longer time (e.g., tens of seconds or even a few minutes), such that sniffers have sufficient time to calculate the capture probabilities and hence the agent can collect feedback.

To increase the overall capture probability, the agent can assign up to b sniffers on the target channel. We denote $\mathcal{N}_s^t = \{n \in \mathcal{N} | \forall a_n^t = 1, a_n^t \in \mathbf{a}^t\}$ as the sniffer set assigned to the target channel at time slot t. Considering that different sniffers are located in different positions and hence have independent capture probabilities, we let Pr^t indicate the overall channel capture probability (i.e., utility), expressed as

$$Pr^t(\mathbf{a}^t, \mathbf{Pr}^t) = 1 - \prod_{n \in \mathcal{N}_s^t} (1 - Pr_n^t) \tag{4.27}$$

When $\mathcal{N}_s^t = \emptyset$, we have $Pr^t = 0$ because there is no sniffer assigned.

The selected sniffers can be removed due to, e.g., intentional jamming or malfunctioning. Here, we assume that up to $\tau < b$ selected sniffers can be nullified and do not contribute anything to the overall utility of the agent. While the agent can explicitly learn the removal probability for each sniffer and account for it when selecting sniffers, this does not address the worst-case scenario, since any sniffer can be removed with a non-zero probability and also the removal probability may not be stationary [17, 82]. Thus, to maximize the worst-case capture probability (i.e, utility), the agent robustly selects sniffers by solving the following problem in the presence of sniffer removal:

$$\begin{aligned}\mathbf{P3}: \qquad & \sum_{t}^{T} \max_{\mathcal{N}_s^t} \min_{\mathcal{N}_r^t} \left(1 - \prod_{n \in \mathcal{N}_s^t \setminus \mathcal{N}_r^t} (1 - Pr_n^t)\right) \\ s.t. \qquad & \mathcal{N}_s^t \subset \mathcal{N}, |\mathcal{N}_s^t| \leq b, \mathcal{N}_r^t \subset \mathcal{N}_s^t, |\mathcal{N}_r^t| \leq \tau, \forall t\end{aligned} \tag{4.28}$$

4.6.2 Applicability of R2C2-MAB

Our algorithm R2C2-MAB achieves worst-case robustness with a provable sublinear regret for contextual combinatorial bandits with arm removal and monotone submodular utility. Here, we confirm that R2C2-MAB is suitable the sniffer channel assignment problem formulated in **P3**.

Note first that the problem **P3** is clearly a contextual combinatorial bandit problem, where the channel SINR for each sniffer is the context, the sniffers are arms, and the selection decision is combinatorial subject to a cardinality constraint. Next, we show that the utility function is monotone and submodular.

Proposition 20 *The utility function of the sniffer channel assignment problem defined in Equation* (4.27) *is monotone and submodular.*

Proof. The proof follows monotonicity and submolarity definitions, and is available in the appendix. ■

The monotonicity of the utility function implies that the optimum of **P3** will be when b sniffers are selected. Thus, the constraint $|\mathcal{N}_s^t| \leq b$ in **P3** will hold with equality $|\mathcal{N}_s^t| = b$. The submodularity indicates that, if no sniffers are removed, the strategy that selects sniffers based on the maximal marginal utility is an efficient algorithm with a good approximation ratio [42, 82, 17]. In the presence of sniffer removal, a robust algorithm is warranted, for which PRO described in Algorithm 5 achieves an approximation ratio under the assumption that the capture probability for each individual sniffer is perfectly known [17].

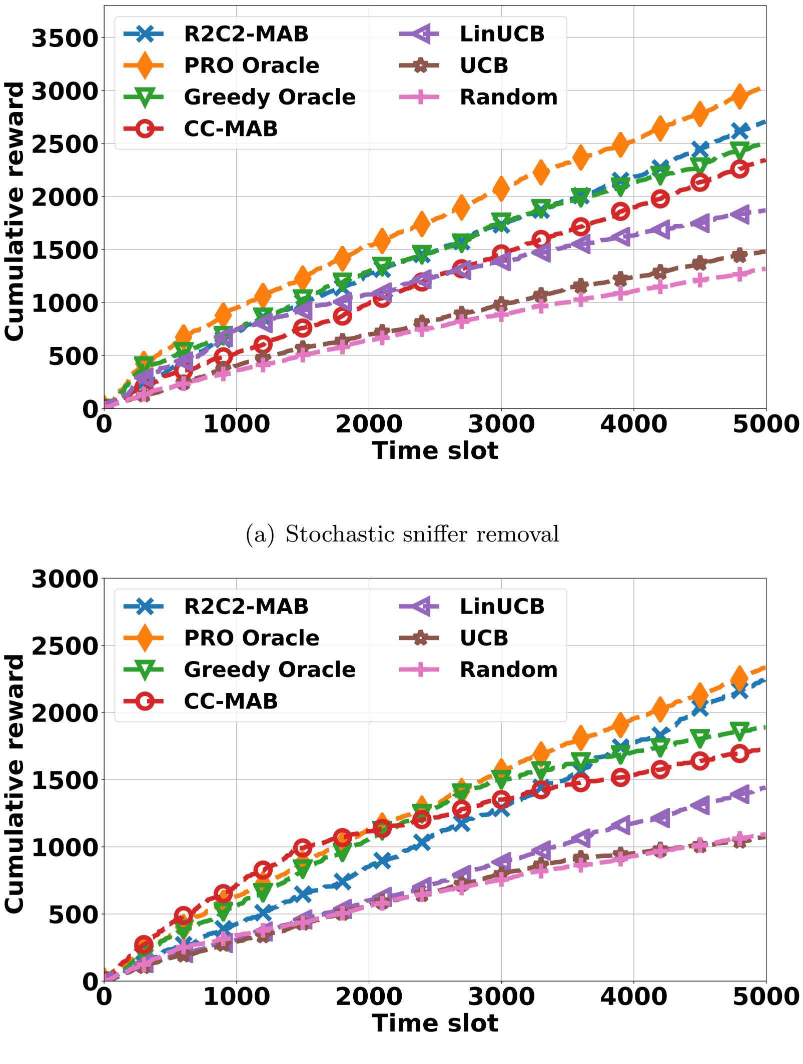
In summary, we use sniffer channel assignment as a concrete application for our considered bandit setting and R2C2-MAB.

4.7 Numerical Evaluation

To empirically evaluate R2C2-MAB, we consider the sniffer channel assignment problem as formulated in Section 4.6. Our results highlight that R2C2-MAB achieves a total reward close to that of PRO Oracle, while outperforming other existing bandit algorithms that either do not exploit the submodularity structure or neglect arm removal.

4.7.1 Settings

We consider a single wireless channel and 8 sniffers ($N = 8$). In each time slot, the sniffer selection budget is limited to 5 ($b = 5$) and up to 2 selected sniffers may be removed (due to malicious attacks or malfunctioning), satisfying removal budget restriction according to Algorithm 5 [82]. We consider two different sniffer removal scenarios: stochastic removal and worst-case adversarial removal. Excluding removed sniffers, we will calculate the overall utility, i.e., channel capture probability, based on the individual capture probability of the remaining selected sniffers.



(a) Stochastic sniffer removal

(b) Adversarial sniffer removal

Figure 4.1: Cumulative reward comparison among different algorithms.

The context of sniffers to estimate the capture probability is their sniffer-channel SINR. We assume that the monitored wireless channel gain G_n^t for sniffer n is varying due to dynamic wireless fading. The context is known context to the agent for sniffer assignment. To formulate the heterogeneous monitoring performance of sniffers, we assume each sniffer n has a unique capture probability function. The actual capture probability function is unknown a priori to the agent and, in our simulation, is generated based on prior studies by considering the background traffic statistics [29].

To evaluate R2C2-MAB, we use the following baselines:

(1) **PRO Oracle:** The PRO oracle knows the exact capture probability of each sniffer given different channel conditions. In each time slot, it chooses 5 sniffers using the PRO algorithm presented in Algorithm 2 [17].

(2) **Greedy Oracle:** The greedy oracle knows the same information as the PRO Oracle, but it chooses 5 sniffers using the greedy approximation algorithm presented in Algorithm 1 by ignoring arm removal.

(3) **CC-MAB:** CC-MAB is the contextual combinatorial MAB algorithm proposed by [30]. In our problem, it uses the SINR context information, learns the rewards of each sniffer under different channel gains, and chooses arms by greedy approximation as in Algorithm 1 without addressing robustness in the presence of arm removal.

(4) **LinUCB:** The classical contextual bandit algorithm selects 5 sniffers as one super-arm in each round. The total number of sniffer decisions (super arms) is $\binom{8}{5} = 56$, and each one has a context vector $x_{sa}^t \in [0,1]^5$, made up by the normalized SINR of 5 sniffers. Notice that LinUCB ignores submodular property and arm removal when selecting sniffers.

(5) **UCB:** UCB (Upper Confidence Bound) is similar to the LinUCB setting, which selects one decisons our of 56 super-arms, but it does not take advantage of the SINR context information and submodular property.

(6) **Random:** The Random algorithm picks 5 sniffers randomly from the set of 8

sniffers in each round.

The simulation runtime is set to $T = 5000$ slots, and each simulation with two different sniffer jamming schema is repeated for 100 times to obtain an averaged value. We evaluate the performance using the metric of cumulative reward.

4.7.2 Stochastic Sniffer Removal

In this scheme, we assume that the environment decides which selected sniffers are removed in a random manner (without knowing each sniffer's capture probability). That is, the environment will stochastically pick 2 sniffers out of 5 selected ones.

The simulation result in Figure 4.1(a) shows that the PRO oracle proposed by [17] has the higher cumulative reward, which is about 60% overall successful channel capture probability on average. This is expected, since the PRO oracle considers sniffer removal explicitly and knows the channel capture probability for each sniffer. Our proposed R2C2-MAB outperforms all the other existing algorithms, including the greedy oracle that knows the exact capture probability of each sniffer but does not consider robustness for arm removal. This indicates that the stochastic sniffer removal by the environment can degrade the performance of greedy significantly, while R2C2-MAB explicitly considers arm removal even though it does not know the exact channel capture probability for each sniffer. Moreover, from the noticeable performance gap between CC-MAB and LinUCB, it shows that monitoring performance can improve greatly if sniffers are assigned with the help of submodular maximization (even though both CC-MAB and LinUCB ignore arm removal).

4.7.3 Adversarial Sniffer Removal

In this case, the environment is more adversarial and removes the selected sniffers (e.g., by intentionally jamming them or making them malfunction) to explicitly reduce the overall utility. Thus, the sniffer performance is expected to become worse than the case of stochastic sniffer removal.

The results in Figure 4.1(b) show that the performance of all algorithms degrade because of the worst-case adversarial sniffer removal. Nonetheless, the two PRO-based methods (PRO Oracle and R2C2-MAB) still have high average channel capture probabilities and outperform the other solutions. More importantly, our R2C2-MAB is getting even closer to the PRO oracle. This result validates that R2C2-MAB is more robust to adversarial arm removal, since it explicitly considers the worst-case arm removal when selecting arms. Although CC-MAB studied in [30] can eventually learn the probability function in terms of the SINR context, its monitoring performance is not as good as R2C2-MAB as it uses simple greedy algorithms without considering arm removal.

Algorithm 6 R2C2-MAB

```
**Inputs** : $T$, $h_T$ and $K_n(t)$.
**Initialization** : Partition $\mathcal{P}_T$; set $C^0(p), \forall p \in \mathcal{P}_T$
for $t = 1, \cdots, T$ do
    arm $n \in \mathcal{N}$ observe contexts $\mathbf{x}_n^t$.
    Find the hypercube $p_n^t$ for each arm $n$
    Identify under-explored arms $\mathcal{N}_{ue}^t$
    if $|\mathcal{N}_{ue}^t| \neq 0$ then
        if $|\mathcal{N}_{ue}^t| \geq b$ then
            Random pick $b$ arms from $\mathcal{N}_{ue}^t$.
        else
            Pick all arms from $\mathcal{N}_{ue}^t$ and select $b - |\mathcal{N}_{ue}^t|$ arms from $\mathcal{N}_t/\mathcal{N}_{ue}^t$ by Algorithm 4
        end if
    else
        Select $b$ arms from $\mathcal{N}$ by by Algorithm 5
    end if
    for $n \in \mathcal{N}_{nr}^t$ do
        Receive reward $d_n^t$
        Find $p_n^t$, where $x_n^t \in p_n^t$
        Update $\hat{d}(p_n^t) = \frac{\hat{d}(p_n^t)C(p_n^t)+d_n^t}{C(p_n^t)+1}$
        Update $C(p_n^t) = C(p_n^t) + 1$
    end for
end for
```

Chapter 5

Federated Contextual Bandit with Differential Privacy

5.1 Introduction

Practical applications of the contextual bandit are increasingly moving into the large-scale decentralized platform, ranging from recommendation [119], finance [51] and clinical trials [39]. The challenge of decentralization is that the real-world data are highly heterogeneous, which are non-independent and identically distributed (non-IID) and highly imbalanced [18] and further urges the collaboration between these heterogeneity to maximize performance [38], which is now known as federated learning (FL)

Federated learning is an emerging distributed machine learning paradigm that has attracted attentions from both academia and industry, because of its overall applicability. The objective of the federated paradigm is to allow collaborative learning with larger amounts of decentralized data, which are exogenously generated at edge devices (from multiple clients, servers, etc.) [55]. FL focuses on many agents collaboratively training a machine learning model under the coordination of a central server while keeping the local data private to others [77]. In real world, privacy leakage have been increasingly reported in recommendation system [23]. An adversary can acquire considerable amount of private information based on the recommendation sequences. Different from the offline learning, online methods directly interact with sensitive data, (e.g., clicks or purchasing history), and timely update the models, which leaves a serious loophole of leaking privacy to the adversary [101, 3]. Realizing its importance, the differential privacy, one of the most powerful definitions of privacy, is utilized to prevent the algorithm's sequential output from revealing private information [53]. Therefore, FL with differential privacy can mitigate the data privacy risks resulting from traditional centralized machine learning, by realizing the principles of designated data collection and minimization.

Most previous work in federated setting, focus on proposing provably algorithms with privacy guarantee on the distributed supervised learning [58, 47]. However, the contextual bandit problem, involves contexts and rewards, which both typically contain sensitive user information [74]. Therefore, federated contextual bandit (FCB) is a very interesting problem for cooperatively learning the environment "on-the fly", while keeping local collected information private. Recently, there is an increasing number of work, focusing on single-agent bandit learning with privacy guarantee [90, 74]. Moreover, without privacy protection, some work only consider bandit learning in distributed settings [38, 75]. However, as the connection between works aforementioned, the federated bandit learning with privacy guarantee is still lack of attention.

In this chapter, we introduce federated contextual bandit with differential privacy, given heterogeneous reward information from multiple agents. Our goal is to provide an algorithm to enable collaborative learning among decentralized sequential decision-makers in the contextual bandit setting, but with strong sub-linear regret upper bound guarantee even with privacy guarantees of each agent's local information. Motivated by the above considerations, we study a novel federated setting with multiple agents coordinating with a center: each of agents will periodically communicate to the center in an epoch manner, to upload their local learnt model with privacy guarantee and receives global aggregated model to continue learning for next epoch. The challenge is that the privacy guarantee will lead suboptimal performance of each agent [38]. To address this challenge, we propose a novel federated contextual bandit algorithm with cut-off threshold for global updates, which is designed building upon non-parameterized contextual bandits [30]. We prove that our proposed algorithm achieves a sublinear cumulative regret of all agents over time compared to the oracle that knows the exact exact reward for all the arms, by properly selecting the cut-off threshold based on the privacy budget.

To evaluate the efficiency of proposed federated bandit algorithm, we consider a contextual admission decision problem for the COVID-19 pandemic. This is shown to fit our federated learning problem with privacy guarantee, since it is beneficial for hospitals collaboratively learning the comprehensive decision schema to maximize the overall utilization of medical resource.

5.2 Related Work

The problem of differentially private online learning was first introduced in [41], ensuring the privacy of the individual entries of the loss vectors. Another tree-based aggregation scheme for releasing the cumulative sums of vectors in a differentially private manner was considered by [40], ensuring that the total amount of noise added for each cumulative sum is only dependent on the number of vectors. [52] proposed gradient-based algorithms that achieve (ϵ, δ) -differntial privacy to protect entire loss vectors. [78] proposed differentially private variants of UCB and Thompson sampling algorithms. [100] proved lower regret bound to design (ϵ, δ)-differentially private algorithms for the stochastic multi-armed bandit problem. [86] prove a tighter regret low bound for multi-armed bandit problem with local differential privacy.

Collaborative bandit algorithms or bandit learning in multi-agent distributed settings has received attention from several academic communities. [111] modeled dependency among social influence through a collaborative reward generation setting. [61] introduced multi-agent bandit for cooperative estimation over a network with delays [27] considered the structure of user dependency as model regularization, assuming similar model to connected users. For the contextual case, recent work has considered collaborative estimation without privacy in networks [38, 109].

The concept of federated bandits has been studied upon by a few works. [67] considers strictly IID local models. [2] study contextual bandits as an example of the federated residual learning framework. [122] focuses on sharing information through gossiping among clients with privacy protection. [91] introduce federated bandits with personalization, where the agent incorporate both global and local models Our work has very different focuses than previous literature. Our work builds on the remarkable work of [37], which in turn improves the collaborative LinUCB algorithm with differential privacy by limiting the global update with thresholds. Our work utilizes non-parameterized bandit framework introduced by [30] and our work guarantee differential privacy, while achieve sublinear regret upper bound.

5.3 Preliminary

In this section, we introduce federated bandit learning with differential privacy. Federated Learning allows multiple agents (e.g. mobile devices and edge server) to collaboratively learn a model while not sharing all the local data, lowering the burden for the cloud to store the massive data.

5.3.1 Federated Bandits

As a combination of federated learning and contextual bandit, federated bandits allows $M(M \geq 2)$ agents are each solving the same contextual bandit in parallel, with local datasets. Each agent observes context information, receives their own arm sets, and selects actions independently of others. In this work, similar to the distributed learning, we consider the centralized environment where there exists a center that periodically communicate with agents, collects their local models and aggregate into a global model back to each agent.

5.3.2 Differential Privacy

In federated settings, each agent prefer to preserve the privacy of their local data in communication, like contexts information. Differential privacy (DP) is the most effective measure to quantify the privacy level of an algorithm. A DP mechanism can challenge the adversary to distinguish two similar data streams. We first introduce the t-neighboring data records as any two data records that differ by only one entry.

Definition 21 *For agent i, two data sequence $S_i = \{x_{i,t}\}_{t=1}^T$ and $S_i^{'} = \{x'_{i,t}\}_{t=1}^T$ is defined as t-neighbors if for each $t' \neq t$, $x_{i,t'} = x'_{i,t'}$*

Let $\mathcal{C}$ be the all possible output set for a randomized algorithm $\mathcal{A}$. Now, we define the notion of ϵ-differential privacy.

Definition 22 *A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for any two t-neighboring data streams, S_i and $S_i^{'}$, and for all $\mathcal{O} \in \mathcal{C}$,*

$$Pr\{\mathcal{A}(S_i) \in \mathcal{O}\} \leq e^{\epsilon} \cdot Pr\{\mathcal{A}(S_i') \in \mathcal{O}\} \tag{5.1}$$

Intuitively, a randomized algorithm $\mathcal{A}$ controls the ability of adversary to distinguish whether or not a specific data record is present in the contextual learning.

5.4 Problem Formulation

In this section, we formulate the differentially-private federated bandit problem with M learning agent and one center. Consider a single learning agent i that interacts with the local environment and makes sequential decisions for a horizon of T time slots. We let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of arms for selection. The agent can observe side information (context) for each arm $x_{i,n}^t \in \mathcal{X} \triangleq [0,1]^D, \forall n \in \mathcal{N}$, where $\mathcal{X}$ is the context space and D is the dimension of context vector. Without loss of generality, we normalize the context space within $[0,1]^D$. The agent i collects the observed context information of all arms in $\mathbf{x}_i^t = \{x_{i,n}^t\}_{n \in \mathcal{N}}$.

In each time slot t, the agent selects an arm n_i^t based on the context information $\mathbf{x}_i^t$ and the knowledge about the reward corresponding to each arm learnt from the previous time slots. At the end of the each time slot, the agent receives a reward feedback $d_{n_i}^t$ based on its context information $x_{i,n}^t$. In most of previous works, it is widely assumed that the reward function is in the linear relationship with the context information. By contrast, in our work, the reward function corresponding to a selected arm is determined solely by the context information $x_{i,n}^t$ through an *unknown* and *non*-parameterized function $d(x_{i,n}^t)$. Without loss of generality, we assume the that reward $d_{n_i}^t$ is bounded by $[0,1]$ for all learning agent.

For communication between agents and the center, we assume the center will collect the local model learned by each agent every epoch, which has a fixed time length τ $(\tau \geq 1)$. After very τ time slots, each agent will send their learned models $\boldsymbol{\theta}_i$ with additive noise η_i to guarantee ϵ-differentially private. The most commonly used additive noise mechanism is Laplacian, where η_i follows a zero-mean Laplace distribution with a scale related to ϵ.

The goal of the federated bandits is to minimize the cumulative group pseudoregret, which can be formally written as

$$R_M(T) = \sum_{i=1}^{M} \sum_{t=1}^{T} \left(d_{n_i^*}^t - d_{n_i}^t \right) \tag{5.2}$$

where $d^t_{n^*_i}$ indicates the reward of optimal arm x^t_{i,n^*} for agent i at time slot t.

5.5 Algorithm Design

In this section, we introduce our algorithm for federated bandit learning with differential privacy. To begin with, we consider the single-agent setting to learn non-parametrized function. Next, we will study the federated setting with multiple agents.

5.5.1 Non-parametric Contextual Bandits

For a single agent, in order to learn a non-parametrized function given the local context information, we assume the reward function satisfies Hölder condition, which indicates the agent will receive similar reward by taking the same arm toward the similar context information. Mathematically, the Hölder condition for reward function $d(x)$ can be represented as

$$|d(x) - d(x')| \leq L||x - x'||^\alpha \tag{5.3}$$

where $L > 0$ and $\alpha > 0$. This is a natural assumption in practice, which can be exploited together with the context information to learn future arm decisions.

Our bandit algorithm uniformly partition the context space, maintained by each agent i. By partition, we split the entire context space into small hypercubes of similar contexts. Then, an agent will learn expected reward of each hypercube independently. Similar to other bandit algorithms, our algorithm is interspersed with exploration phases and exploitation phases. In the exploration phases, the agent randomly select arms. During exploration, agent learn the reward patterns of arms which have not been explored before.

Otherwise, the algorithm is in an exploitation phase, where agent chooses arm in a greedy manner, only considering the highest expected reward based on previous history with similar contexts. After choosing the arm, the agent observes the reward at the end of every time slot. In this way, the algorithm learns context-specific function over time. The algorithm design challenge lies in how to partition the context space and how to balance exploration and exploitation.

The pseudo-code of non-parametric contextual bandits is presented in Algorithm 7. In the beginning, given context space $\mathcal{X} = [0,1]^D$, our algorithm creates a partition $\mathcal{P}_T$ with time horizon T, which splits the context space evenly into $(h_T)^D$ sets. h_T is an hyper-parameter which determines the number of hypercubes in the partition. The estimated reward of each hypercube $p \in \mathcal{P}_T$ can be computed by the accumulated reward falls. Let $\mathcal{H}^t(p)$ represent the reward history of hypercube p up to time slot t, then the estimated reward for contexts falls in p

$$\hat{d}(p) = \frac{1}{|\mathcal{H}^t(p)|} \Sigma_{d \in \mathcal{H}^t(p)} d \tag{5.4}$$

Additionally, $C^t(p) = |\mathcal{H}(p)|$, which is a counter for agent recording the number of times that selected arm context x_n^t is included in hypercube p.

In each time slot t, agent i first observes $\mathbf{x}_i^t$ of all arm. Algorithm determines the hypercube sets P_i^t contains $\mathbf{x}_i^t$, defined as

$$P_i^t = \{p_{i,n}^t : \exists x_{i,n}^t \in p_{i,n}^t, \forall p_{i,n}^t \in \mathcal{P}_T\} \tag{5.5}$$

Aforementioned, one challenge is that the proposed algorithm has to balance the exploration and exploitation in case there exist hypercubes in P_i^t has not been explored sufficiently. Therefore, we denote under-explored hypercubes as

$$\mathcal{N}^{ue,t} = \{p : \exists p \in P_i^t, C_i^t(p) \leq K(t)\} \tag{5.6}$$

where $K(t)$ monotonically increasing function.

For every time slot t, if $\mathcal{N}_t^{ue}$ is not empty, then we just randomly pick one arm from $\mathcal{N}_t^{ue}$. If $\mathcal{N}_t^{ue}$ is empty, then we choose b arms by

$$\arg\max_{n \in \mathcal{N}} \hat{d}(p_{i,n}^t) \tag{5.7}$$

5.5.2 Federated Bandits with Differential Privacy

Based on the non-parametric contextual bandits for a single agent, we introduce our algorithm for federated bandit learning with differential privacy. We consider the centralized environment where there exists a center that collects local mode (hypercubes) of each agent every τ time. When $(t \bmod \tau) = 0$, each agent will send the information of all hypercube $\boldsymbol{\theta}_i^t$ they maintain, including hypercube counter $C_i^t(p)$ and estimated reward $\hat{d}(p)$. Otherwise, the agent learn the reward function from local information by Algorithm 7. Due to all agents are learning the same non-parametrized reward function, their partition $\mathcal{P}_t$ over context space is consistent.

To guarantee ϵ-differentially private through the communication toward the center, each agent deliberately add zero-mean Laplacian noise η_i on estimated reward $\hat{d}(p)$ for every hypercube, denoted as $\tilde{d}(p)$. According to Laplace mechanism, $\eta_i \sim Lap(0, \frac{\Delta \hat{d}}{\epsilon})$. $\Delta \hat{d}$ is sensitivity of a estimated reward function $\hat{d}$, in our case, we defined as:

$$\Delta \hat{d} = \max |\hat{d}(\mathcal{H}(p)) - \hat{d}(\mathcal{H}'(p))| \tag{5.8}$$

Algorithm 7 Non-parametric Contextual Bandits

```
**Inputs** : $T$, $h_T$ and $K(t)$.
**Initialization** : Partition $\mathcal{P}_T$; set $C_i^0(p) = 0, \hat{d}(p) = 0, \forall p \in \mathcal{P}_T$
**for** $t = 1, \cdots, T$ **do**
    Agent observe contexts $\mathbf{x}_i^t$.
    Find the hypercube set $P_i^t$
    Identify under-explored hypercubes $\mathcal{N}^{ue,t}$
    **if** $|\mathcal{N}^{ue,t}| \neq 0$ **then**
        Random select arm from $\mathcal{N}^{ue,t}$.
    **else**
        Select arm from $\mathcal{N}$ by (5.7)
    **end if**
    Receive reward $d_{n_i}^t$
    Update $\hat{d}(p_{i,n}^t) = \frac{\hat{d}(p_{i,n}^t)C_i^t(p_{i,n}^t)+d_{n_i}^t}{C_i^t(p_{i,n}^t)+1}$
    Update $C_i^t(p_{i,n}^t) = C_i^t(p_{i,n}^t) + 1$
**end for**
```

where $\mathcal{H}(p)$ and $\mathcal{H}'(p)$ are t-neighboring reward history for hypercube p with same length. Apparently, $\Delta\hat{d} = 1$ because the reward is bounded in $[0, 1]$. Then, after adding noise as $\eta_i \sim Lap(0, \frac{1}{\epsilon})$ to estimated reward, each agent will send $\tilde{\boldsymbol{\theta}}_i^t$ to assure that communication achieves ϵ-differential level privacy.

After the center received the information from all agents, it aggregate estimated reward of each hypercube as

$$\tilde{d}(p) = \frac{\sum_{i=1}^{M}\left(C_i^t(p)\cdot\tilde{d}\right)}{\sum_{i=1}^{M}C_i^t(p)} \tag{5.9}$$

and sum up counters of each hypercube

$$C^t(p) = \sum_{i=1}^{M} C_i^t(p) \tag{5.10}$$

At the end of time slot, the center return the aggregated hypercube information $\boldsymbol{\theta}_c^t$ to each agent. For next τ-time epoch, each agent continue learning based on updated counter and estimated reward from $\boldsymbol{\theta}_c^t$.

However, the global update $\boldsymbol{\theta}_c^t$ is biased because of the global estimated reward of each hypercube is the linear combination of Laplacian variables. In federated learning, the privacy guarantee yield sub-optimal learning performance and we defer the detail to the subsequent section. To alleviate this bias, the agent need to decide which hypercube should be cut off from the global updates and keep local estimates. We define the cut-off threshold ρ for each hypercube. For agent i, after receiving $\boldsymbol{\theta}_c^t$, if $C_i^t(p) \geq \rho$ for hypercube p, then agent will not update from global information for hypercube p. The threshold ρ is to limit the total number of update from global $\boldsymbol{\theta}_c^t$ after each of agents fully takes advantage of collective information $\boldsymbol{\theta}_c^t$. Further, the design of ρ curbs the regret caused by the bias from communication. We present the centralized algorithm in Algorithm 8 that details our approach.

Algorithm 8 Centralized Federated Bandit

```
**Inputs** : $T$, $\tau$, $\epsilon$, $\rho$ and $K(t)$.
**Initialization** : Partition $\mathcal{P}_T$; set $C^0(p) = 0, \tilde{d}(p) = 0, \forall p \in \mathcal{P}_T$
**for** $t = 1, \cdots, T$ **do**
    **if** $t \bmod \tau = 0$ **then**
        $\forall$ Agents generate $\tilde{\boldsymbol{\theta}}_i^t$ by $Lap(0, \frac{1}{\epsilon})$ and communicate to center.
        Center aggregate hypercube information $\tilde{d}(p)$ by Eqn. (5.9) and $C^t(p)$ by Eqn. (5.10)
        Center send aggregated $\boldsymbol{\theta}_c^t$ to agents.
        **for** $i = 1, \cdots, M$ **do**
            **for** $\forall p \in \mathcal{P}_T$ **do**
                **if** $C_i^t(p) \leq \rho$ **then**
                    $\hat{d}_i(p) = \tilde{d}(p)$, $C_i^t(p) = C^t(p)$.
                **end if**
            **end for**
        **end for**
    **else**
        $\forall$ Agents implement **Algorithm 7**
    **end if**
**end for**
```

5.6 Regret Analysis

The regret bound is derived based on the assumption that the expected reward of arms are similar in similar contexts. To analyze regret of Algorithm 8, we define

$\bar{d}(p) = \sup_{x \in p} d(p)$ and $\underline{d}(p) = \sup_{x \in p} d(p)$ as highest and lowest demand overall context in hypercube p. Since we need to compare expected reward in different hypercubes, so we set geometric center of each hypercube p as the reference context, denoted as $x^*(p)$. Therefore, the optimal arm for agent i at time slot t satisfies

$$n_i^* = \underset{p_{i,n}^t \in \mathcal{P}_T}{\arg\max}\, d(x^*(p_{i,n}^t)) \tag{5.11}$$

Therefore, the regret $R(T)$ can be represented as

$$R(T) = \sum_{i=1}^{M} \sum_{t=1}^{T} \left(d(x^*(p_{i,n^*}^t)) - d(x^*(p_{i,n}^t)) \right) \tag{5.12}$$

To be specific, we define the sub-optimal arm (hypercube) set $\mathcal{N}^{sub,t}$ as

$$\mathcal{N}^{sub,t} = \{\exists p_{i,n}^t \in P_i^t, \underline{d}(p_{i,n^*}^t) - \bar{d}(p_{i,n}^t) \geq \rho t^{\frac{-\epsilon}{\rho}}\} \tag{5.13}$$

Consequently, $\mathcal{N}^{near,t} := \mathcal{N}/\mathcal{N}^{sub,t}$ is the near-optimal arm set.

After defining different arm sets, the $R(T)$ can be bounded by sum of following four parts.

$$R(T) \leq \mathbb{E}[R_e(T)] + \mathbb{E}[R_s(T)] + \mathbb{E}[R_n(T)] + \mathbb{E}[R_m(T)] \tag{5.14}$$

The $\mathbb{E}[R_e(T)]$ is regret due to exploration for hypercubes in Algorithm 7. Specifically, $\mathbb{E}[R_m(T)]$ is by choosing non-optimal arms because of the bias from global update through communication to the center, $\mathbb{E}[R_s(T)]$ is due to choosing sub-optimal arm and $\mathbb{E}[R_n(T)]$ is regret caused by near-optimal arm. The reason that the $R(T)$ is upper bounded by the sum of those four parts, is the existence of the overlapping regret among $\mathbb{E}[R_m(T)]$, $\mathbb{E}[R_s(T)]$ and $\mathbb{E}[R_n(T)]$. For convenience, we consider those regrets separately, and in this way, we still can prove cumulative regret of proposed centralized federated bandit algorithm is sublinear with time when four regrets are bounded respectively.

Lemma 23 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$, the exploration regret $\mathbb{E}[R_e(T)]$ is bounded by*

$$\mathbb{E}[R_e(T)] \leq 2^D (\log(T) T^{z+\gamma D} + T^{\gamma D}) \tag{5.15}$$

Proof. In the exploration phase, there exist at least a hypercube p in P_i^t, that $C_i^t(p) \leq K(t) = t^z \log(t)$. There are total $(h_T)^D$ partitioned hypercubes, and the global update of hypercube counters is intact. Then we have at most $(h_T)^D \lceil T^z \log(T) \rceil$ exploration phases. We bound the reward function $d(x)$ by 1. For $\mathbb{E}[R_e(T)]$, we have

$$\begin{aligned} \mathbb{E}[R_e(T)] &\leq (h_T)^D \lceil T^z \log(T) \rceil \\ &= \lceil T^\gamma \rceil^D \lceil T^z \log(T) \rceil \\ &\leq 2^D (\log(T) T^{z+\gamma D} + T^{\gamma D}) \end{aligned}$$

The last inequality is due to the fact that $\lceil T^\gamma \rceil^D \leq (2T^\gamma)^D = 2^D T^{\gamma D}$

This completes the proof. ■

Lemma 24 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$. We assume the condition $H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{\rho t^{\frac{-\epsilon}{\rho}}}{2}$ holds true, where $H(t) := t^{\frac{-z}{2}}$. For all $1 \leq t \leq T$ the sub-optimal regret $\mathbb{E}[R_s(T)]$ is bounded by*

$$\mathbb{E}[R_s(T)] \leq M \frac{\pi^2}{3} \tag{5.16}$$

Proof. For $1 \leq t \leq T$, we let $W(t)$ be the event that slot t is in exploitation phase for agent i, which indicates $C_i^t(p) \geq K(t) = t^z \log(t), \forall p \in P_i^t$. Let $u(t)$ be the event that agent picked the sub-optimal hypercube $p_{i,n'}^t$, other than p_{i,n^*}^t. According to designed algorithm, when $u(t)$ happens, it indicates estimated reward of p_{i,n^*}^t is not the highest,

$\hat{d}(p^t_{i,n'}) \geq \hat{d}(p^t_{i,n^*}), \exists p^t_{i,n'} \in P^t_i$. Thus, we have

$$Pr\{u(t)\} = Pr\{\hat{d}(p^t_{i,n'}) \geq \hat{d}(p^t_{i,n^*})\} \tag{5.17}$$

The right side of Eqn.(5.17) indicates at least one of three following events happen when $H(t) \geq 0$

$$\begin{aligned}
E_1 =& \{\hat{d}(p^t_{i,n'}) \geq \bar{d}(p^t_{i,n'}) + H(t), W(t)\} \\
E_2 =& \{\hat{d}(p^t_{i,n^*}) \leq \underline{d}(p^t_{i,n^*}) - H(t), W(t)\} \\
E_3 =& \{\hat{d}(p^t_{i,n'}) \geq \hat{d}(p^t_{i,n^*}), \\
& \hat{d}(p^t_{i,n'}) \leq \bar{d}(p^t_{i,n'}) + H(t), \\
& \hat{d}(p^t_{i,n^*}) \geq \underline{d}(p^t_{i,n^*}) - H(t), W(t)\}
\end{aligned} \tag{5.18}$$

Hence, we have

$$\{\hat{d}(p^t_{i,n'}) \geq \hat{d}(p^t_{i,n^*})\} \subseteq E_1 \cup E_2 \cup E_3 \tag{5.19}$$

The next step is to bound probability of E_1, E_2 and E_3 separately. For $Pr\{E_1\}$, we have

$$\begin{aligned}
Pr\{E_1\} &= Pr\{\hat{d}(p^t_{i,n'}) \geq \bar{d}(p^t_{i,n'}) + H(t), W(t)\} \\
&\leq Pr\{\hat{d}(p^t_{i,n'}) \geq \mathbb{E}[\hat{d}(p^t_{i,n'})] + H(t), W(t)\} \\
&= Pr\{\hat{d}(p^t_{i,n'}) - \mathbb{E}[\hat{d}(p^t_{i,n'})] \geq H(t), W(t)\} \\
&\leq \exp\left(\frac{-2C^t_i(p^t_{i,n'})H(t)^2}{(d^{max})^2}\right) \\
&\leq \exp\left(-2t^z \log(t) H(t)^2\right)
\end{aligned} \tag{5.20}$$

The first inequality of Eqn. (5.20) comes from the fact that $\mathbb{E}[\hat{d}(p)] \leq \bar{d}(p), \forall p \in \mathcal{P}_t$. The last two steps of Eqn. 5.20 are due to Chernoff-Hoeffding bound and $W(t)$ implies at least

$t^z \log(t)$ times counted in $C_i^t(p), \forall p \in P_i^t$ and $d^{max} = 1$. Similarly, the $Pr\{E_2\}$ can be bounded in the same way.

$$Pr\{E_2\} \leq \exp\left(-2t^z \log(t) H(t)^2\right) \tag{5.21}$$

Last, to bound $Pr\{E_3\}$, we rewrite $\hat{d}(p), \forall p \in \mathcal{P}_t$ as

$$\hat{d}(p) = \frac{1}{C(p)} \sum_{\tau : x_\tau \in p} d(x_\tau) + \epsilon_\tau \tag{5.22}$$

where x_τ are the context falling in hypercube p and ϵ_τ is the deviation from estimated reward of hypercube p. Moreover, we define best and worst context for p as $\bar{x}(p) := \arg\max_{x \in p} d(x)$ and $\underline{x}(p) := \arg\min_{x \in p} d(x)$ respectively. After, we have

$$\begin{aligned} \bar{d}(p) &= \frac{1}{C(p)} \Sigma_{\tau : x_\tau \in p} d(\bar{x}_\tau(p)) + \epsilon_\tau \\ \underline{d}(p) &= \frac{1}{C(p)} \Sigma_{\tau : x_\tau \in p} d(\underline{x}_\tau(p)) + \epsilon_\tau \end{aligned} \tag{5.23}$$

From Hölder condition, we have for $\forall p \in \mathcal{P}_T$

$$\begin{aligned} \bar{d}(p) - \hat{d}(p) &\leq L D^{\frac{\alpha}{2}} h_T^{-\alpha} \\ \hat{d}(p) - \underline{d}(p) &\leq L D^{\frac{\alpha}{2}} h_T^{-\alpha} \end{aligned} \tag{5.24}$$

We consider three components of E_3 separately. For the first component, we have

$$\{\hat{d}(p_{i,n'}^t) \geq \hat{d}(p_{i,n^*}^t)\} \subseteq \{\bar{d}(p_{i,n'}^t) \geq \underline{d}(p_{i,n^*}^t)\} \tag{5.25}$$

For second part of E_3, by Eqn. (5.24) we have

$$\begin{aligned} &\{\hat{d}(p_{i,n'}^t) \leq \bar{d}(p_{i,n'}^t) + H(t)\} \\ &\quad \subseteq \{\hat{d}(p_{i,n'}^t) - L D^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \bar{d}(p_{i,n'}^t) + H(t)\} \\ &\quad = \{\hat{d}(p_{i,n'}^t) \leq \bar{d}(p_{i,n'}^t) + H(t) + L D^{\frac{\alpha}{2}} h_T^{-\alpha}\} \end{aligned} \tag{5.26}$$

For the last component of E_3, using Eqn. (5.24) again, we have

$$\begin{aligned}
&\{\hat{d}(p_{i,n^*}^t) \geq \underline{d}(p_{i,n^*}^t) - H(t)\} \\
&\qquad \subseteq \{\hat{d}(p_{i,n^*}^t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \geq \underline{d}(p_{i,n^*}^t) - H(t)\} \\
&\qquad = \{\hat{d}(p_{i,n^*}^t) \geq \underline{d}(p_{i,n^*}^t) - H(t) - LD^{\frac{\alpha}{2}} h_T^{-\alpha}\}
\end{aligned} \tag{5.27}$$

Combining Eqn. (5.25), (5.26) and (5.27), the probability of E_3 is bounded by

$$\begin{aligned}
Pr\{E_3\} \leq Pr\{&\bar{d}(p_{i,n'}^t) \geq \underline{d}(p_{i,n^*}^t), \\
&\hat{d}(p_{i,n'}^t) \leq \bar{d}(p_{i,n'}^t) + H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha}, \\
&\hat{d}(p_{i,n^*}^t) \geq \underline{d}(p_{i,n^*}^t) - H(t) - LD^{\frac{\alpha}{2}} h_T^{-\alpha}\}
\end{aligned} \tag{5.28}$$

We know the following condition is satisfied:

$$H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{\rho t^{\frac{-\epsilon}{\rho}}}{2} \tag{5.29}$$

Since we know the definition of sub-optimal arm by Eqn.(5.13), together with Eqn. (5.29), implies that

$$\begin{aligned}
\underline{d}(p_{i,n^*}^t) - \bar{d}(p_{i,n'}^t) \geq 2H(t) + 2LD^{\frac{\alpha}{2}} h_T^{-\alpha} \\
\underline{d}(p_{i,n^*}^t) - H(t) - LD^{\frac{\alpha}{2}} h_T^{-\alpha} \geq \bar{d}(p_{i,n'}^t) + H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha}
\end{aligned} \tag{5.30}$$

Apparently, Eqn. (5.30) is contradict with Eqn. (5.28), which turns out $Pr\{E_3\} = 0$ under condition Eqn. (5.29). Therefore, we let $H(t) = t^{\frac{-z}{2}}$, then from (5.20), we have

$$\begin{aligned}
Pr\{E_1\} &\leq \exp\left(-2t^z \log(t) H(t)^2\right) \\
&= \exp\left(-2t^z \log(t) (t^{\frac{-z}{2}})^2\right) \\
&= \exp(-2\log(t)) \\
&= t^{-2}
\end{aligned} \tag{5.31}$$

Similarly,

$$Pr\{E_2\} \leq t^{-2} \tag{5.32}$$

Combining conclusions (5.31) and (5.32), we have

$$\begin{aligned} Pr\{u(t), W(t)\} &\leq Pr\{E_1 \cup E_2 \cup E_3\} \\ &\leq Pr\{E_1\} + Pr\{E_2\} + Pr\{E_3\} \\ &= 2t^{-2} \end{aligned} \tag{5.33}$$

Since we have the probability bound for one agent sub-optimal hypercube in Eqn. (5.33), with Eqn. (5.33), we have

$$\begin{aligned} \mathbb{E}[R_m(T)] &\leq M \sum_{t=1}^{T} Pr\{u(t), W(t)\} \\ &\leq M \sum_{t=1}^{T} 2t^{-2} \leq M \sum_{t=1}^{\infty} 2t^{-2} \\ &\leq M \frac{\pi^2}{3} \end{aligned} \tag{5.34}$$

■

Lemma 25 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$. We define the near-optimal arm set as $\underline{d}(p^t_{i,n^*}) - \bar{d}(p^t_{i,n}) \leq \rho t^{\frac{-\epsilon}{\rho}}, \exists p^t_{i,n} \in P^t_i$. For all $1 \leq t \leq T$ the near-optimal regret $\mathbb{E}[R_n(T)]$ is bounded by*

$$\mathbb{E}[R_n(T)] \leq M \left(L D^{\frac{\alpha}{2}} T^{1-\gamma\alpha} + \frac{\rho^2}{\rho - \epsilon} T^{1-\frac{\epsilon}{\rho}} \right) \tag{5.35}$$

Proof. Considered $W(t)$ event for exploitation phase, the loss due to near-optimal is defined as

$$R_n(T) = \sum_{t=1}^{T} I_{\{Q(t)\}} \times \left(d(x^*(p^t_{i,n^*})) - d(x^*(p^t_{i,n^\dagger})) \right) \tag{5.36}$$

where $Q(t)$ indicates event that choosing near-optimal arm.

Further, by taking expectation, we have

$$\begin{aligned}\mathbb{E}[R_n(T)] &= \sum_{t=1}^{T} \mathbb{E}\left[I_{\{Q(t)\}} \times \Big(d(x^*(p^t_{i,n^*})) - d(x^*(p^t_{i,n^\dagger}))\Big)\right] \\ &= \sum_{t=1}^{T} Pr\{Q(t)\} \times \Big(d(x^*(p^t_{i,n^*})) - d(x^*(p^t_{i,n^\dagger}))\Big) \\ &\leq \sum_{t=1}^{T} \Big(d(x^*(p^t_{i,n^*})) - d(x^*(p^t_{i,n^\dagger}))\Big)\end{aligned} \tag{5.37}$$

After applying Hölder condition multiple times, we have

$$\begin{aligned}&\sum_{t=1}^{T} \Big(d(x^*(p^t_{i,n^*})) - d(x^*(p^t_{i,n^\dagger}))\Big) \\ &\leq \sum_{t=1}^{T} \left(\inf_{x \in p^t_{i,n^*}} d(x) - d(x^*(p^t_{i,n^\dagger})) + bLD^{\frac{\alpha}{2}} h_T^{-\alpha}\right) \\ &\leq \sum_{t=1}^{T} \left(\inf_{x \in p^t_{i,n^*}} d(x) - \sup_{x \in p^t_{i,n^\dagger}} d(x) + 2bLD^{\frac{\alpha}{2}} h_T^{-\alpha}\right) \\ &= \sum_{t=1}^{T} \Big(\underline{d}(p^t_{i,n^*}) - \bar{d}(p^t_{i,n^\dagger}) + 2bLD^{\frac{\alpha}{2}} h_T^{-\alpha}\Big) \\ &= \sum_{t=1}^{T} \Big(\rho t^{\frac{-\epsilon}{\rho}} + 2bLD^{\frac{\alpha}{2}} h_T^{-\alpha}\Big)\end{aligned} \tag{5.38}$$

Then, we use the fact that $h_T^{-\alpha} = \lceil T^\gamma \rceil^{-\alpha} \leq T^{-\gamma\alpha}$, we have

$$\begin{aligned}\mathbb{E}(R_n(T)) &\leq \sum_{t=1}^{T}\sum_{i=1}^{M} \Big(\rho t^{\frac{-\epsilon}{\rho}} + 2LD^{\frac{\alpha}{2}} h_T^{-\alpha}\Big) \\ &\leq M\left(2LD^{\frac{\alpha}{2}} T^{1-\gamma_\alpha} + \frac{\rho}{1-\frac{\epsilon}{\rho}} T^{1-\frac{\epsilon}{\rho}}\right) \\ &= M\left(2LD^{\frac{\alpha}{2}} T^{1-\gamma_\alpha} + \frac{\rho^2}{\rho-\epsilon} T^{1-\frac{\epsilon}{\rho}}\right)\end{aligned} \tag{5.39}$$

■

Combining Lemma 23, 24 and 25, we have regret bound for $\mathbb{E}[R_T]$,

Theorem 26 *Let $K(t) = t^z \log(t)$ and $h_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < \frac{1}{D}$. We assume $H(t) + LD^{\frac{\alpha}{2}} h_T^{-\alpha} \leq \frac{\rho t^{\frac{-\epsilon}{\rho}}}{2}$ and $\epsilon \leq \rho$ holds true, where $H(t) = t^{\frac{-z}{2}}$. For all $1 \leq t \leq T$ the total regret $\mathbb{E}[R(T)]$ is bounded by*

$$\mathbb{E}[R(T)] \leq \log(T) T^{z+\gamma D} + T^{\gamma D} + M\left(\frac{\pi^2}{3} + 2LD^{\frac{\alpha}{2}} T^{1-\gamma\alpha} + \frac{\rho^2}{\rho - \epsilon} T^{1-\frac{\epsilon}{\rho}}\right) \tag{5.40}$$

To bound the leading order of Eqn. (5.40) be sublinear with time, we carefully choose parameter, $z = \frac{2\alpha}{3\alpha+D} \in (0, 1)$, and $\gamma = \frac{z}{\alpha} \in (0, \frac{1}{D})$. Then the expected regret reduces to

$$\mathbb{E}[R(T)] \leq \log(T) T^{\frac{2\alpha+D}{3\alpha+D}} + T^{\frac{D}{3\alpha+D}} + M\left(\frac{\pi^2}{3} + LD^{\frac{\alpha}{2}} T^{\frac{2\alpha+D}{3\alpha+D}} + \frac{\rho^2}{\rho - \epsilon} T^{1-\frac{\epsilon}{\rho}}\right) \tag{5.41}$$

Since we know $\epsilon > 0$, the leading order of regret bound is sub-linear and determined by $\max(\log(T)T^{\frac{2\alpha+D}{3\alpha+D}}, \frac{\rho^2}{\rho-\epsilon}T^{1-\frac{\epsilon}{\rho}})$. If $\epsilon \ll 1$, it indicates the privacy budget is very large and the regret performance is hard to be guaranteed under strong privacy-setting. The last term will goes to zero if $\epsilon \to \rho$, which shows that our regret bound caused by differential privacy protection is shrinking. Moreover, when each agent keep update from global model through communication to center ($\rho \to \infty$), the regret bound will goes to infinity because the continual perturbation from the additive privacy noise.

5.7 Experiments and Results

The severe acute respiratory syndrome coronavirus-2 (SARS-CoV2) has caused the current pandemic of coronavirus disease-19 (COVID-19), which first emerged as an outbreak in December 2019. The admission decision of COVID-19 infected patients remains problematic and challenging, although this is to be expected in such a recently emerged disease. According to the guideline [15], the mild infected patients are recommended to recover at home or be admitted to the hospital when condition worsen. However, an severely infected patient can rapidly develop further more severe symptoms which can be life-threatening and require intensive care intervention (ICU) [6]. However, ICU beds are a precious resource in locations where COVID-19 case numbers are high. Therefore, avoiding ICU beds run-out require far-sighted admission decision for infected patients, according to the status of patients.

In this section, to evaluate our proposed centeralized federated bandit algorithm, we consider the application of contextual COVID-19 admission management for multiple resource-constrained hospitals — thoughtful admission decision needed to be made for infected patients based on their body status (context), to maximize the utility.

5.7.1 Contextual COVID-19 Admission Decision

We have collected COVID-19 patients' datasets from the Kaggle online resource [94], which contains the grouped information of previous diseases, blood sample results, vital sign data and admission record of more than 2000 COVID-19 positive patients. One consideration for utilizing this dataset for our federated bandit problem, is that the patient information is location-dependent, which means one hospital may receives patient with similar information (e.g. age). To have a more comprehensive decision schema for patients, multiple hospitals can collaboratively find the optimal decision by sharing their decision knowledge. Due to the patient information is very sensitive and personal, powerful privacy protection is needed when sharing.

We utilize the age, respiration rate and oxygen saturation rate as the contextual information for patients. The action (arm) is to decide the patient should be admitted

Table 5.1: Reward Matrix

	ICU	Not ICU
Admission	0.8	0.2
Not Admission	0.1	0.9

to the hospital ward (0 for no admission, 1 for admission). In this way, the dimension of observed contextual vector is four, which is in the form of <age, respiration rate, oxygen saturation rate, admission decision>. The reward of the admission decision depends on whether the patient is sent to ICU due to the severe conditions. For example, if the patient is admitted to hospital and transferred to ICU later, it indicates that the admission decision is a thoughtful move for this patient and should earn a high reward. Otherwise, denying admission for patients who will be sent to ICU eventually, is not preferred (yielding low reward). The challenge for this problem is that the hospital need to estimate the severity of COVID-19 infected patient according to their contexts, then determine admission decision for further ICU usage.

5.7.2 Simulation Settings

We consider a collaboration center and 4 hospitals ($M = 4$). In each time slot, the hospital learn the local decision schema based on Algorithm 7 and conduct the federated bandit learning in Algorithm 8. The actual non-parameterized reward function is unknown a priori to the each hospital and, in our simulation, is generated based on the reward matrix in Table 5.1.

To evaluate our proposed federated bandit algorithm, we use the following baselines:

(1) **CFB w/o privacy:** Each hospital use Algorithm 8 without additive Laplacian noise for differential privacy and update cut-off..

(2) **CFB w/o cut-off:** Each hospital use Algorithm 8 without update cut-off.

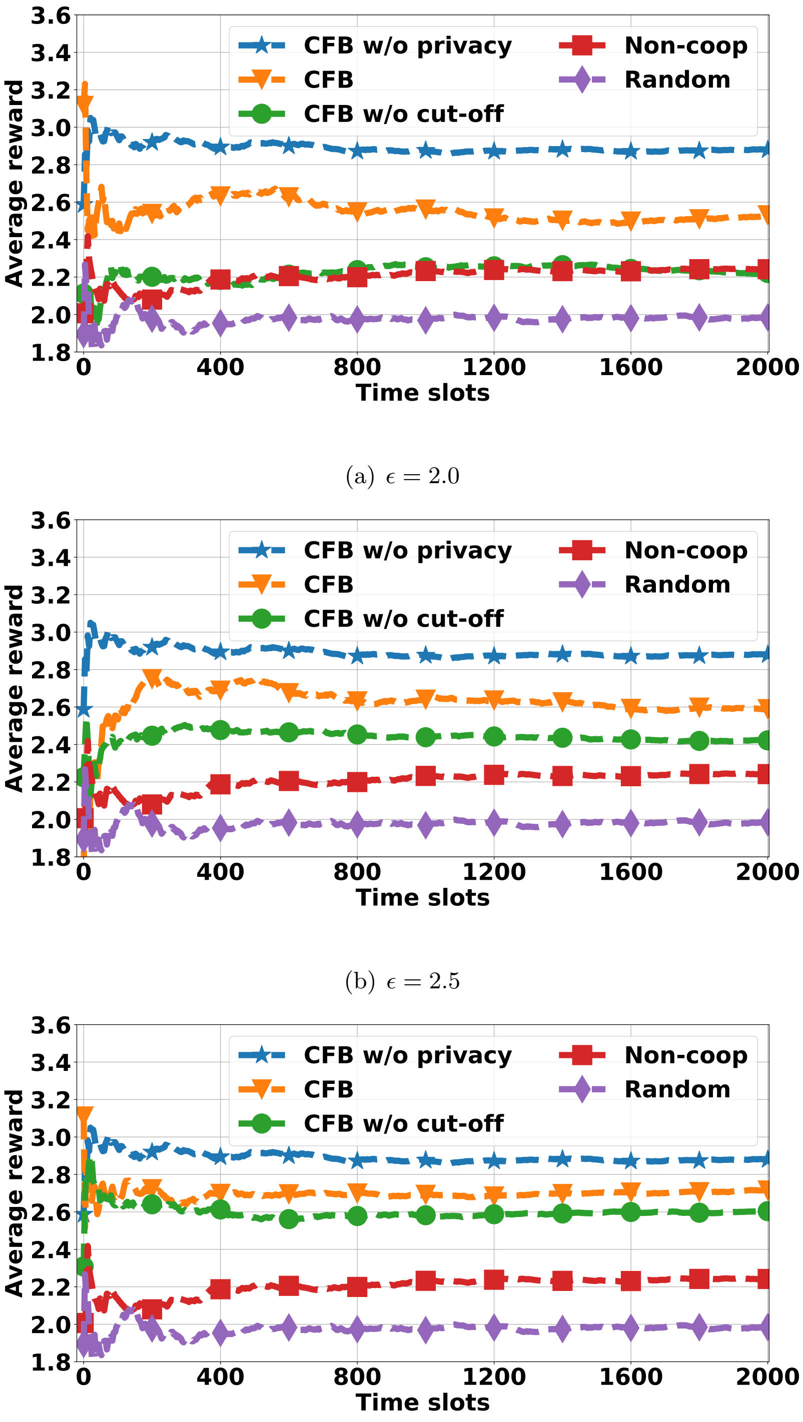
(3) **Non-coop:** The Non-coop algorithm let each hospital use Algorithm 7 to find local decision schema without communication to the center.

(4) **Random:** The Random algorithm decide admissions randomly for every infected patient.

The simulation runtime is set to $T = 2000$ slots, and we set the epoch length $\tau = 50$ slots and cut-off threshold as 10 . We evenly partition patient context (age, respiration rate, oxygen saturation rate) by setting $h_T = 4$ and there are $4^3 \times 2 = 128$ hypercubes (2 for binary arms) in total. We group the dataset by age and hospitals receive each age group in different time order. We test our proposed algorithm with benchmarks with different privacy budget ($\epsilon = 2.0, 2.5, 3.0$). We evaluate the performance using the metric of average reward for 4 hospitals over time.

5.7.3 Simulation Results

The results in Figure 5.1 show that the performance of our proposed algorithm with or without cut-off threshold degrade because of privacy guarantee, compared to the non-privacy setting. Nonetheless, they still have high average reward and outperform the Non-coop and random solutions. More importantly, our algorithm with privacy guarantee is getting even closer to the non-privacy setting as the privacy budget shrinks. This result validates our regret analysis since less budget leads minor noise in global update from the center. Moreover, the performance gap between with and without cut-off decreases as ϵ increases, which indicates that the cut-off threshold is very significant to lower impact of noisy perturbation when strong privacy needed.



(a) $\epsilon = 2.0$

(b) $\epsilon = 2.5$

(c) $\epsilon = 3.0$

Figure 5.1: Average reward for different privacy budget

Chapter 6

Conclusions

This dissertation focused on the contextual bandit learning problem within the framework of imperfect environment. Four emerging issues in terms of the uncertain probabilistic context, missing or delayed reward feedback, adversarial arm removal and collaborative learning with privacy protection were studied under various bandit models.

In the first part of dissertation, we consider a new setting of bandit learning with multiple feedback signals, time-varying utility functions and probabilistic context information. For this setting, we propose a multi-feedback probabilistic kernelized UCB algorithm to choose the optimal arm in order to minimize the expected cumulative regret. We derive an upper bound of the expected cumulative regret incurred by our proposed algorithm, with respect to the best action that maximize the expected reward, and show that the bound grows sub-linearly with time. We apply the proposed algorithm to DNN model selection. The simulation results further validate the sub-linearity of the cumulative regret.

In the second part, we propose an algorithm based on delayed contextual UCB for arm selection, which updates its reward function learning whenever new reward feedback is received, and also establish an upper bound on the cumulative regret. Then, we propose a novel extension by using semi-supervised learning to produce fictitious estimates for delayed or missing rewards. Finally, we apply our algorithms to the problem of an online context-aware news recommendation to find the most preferred articles to users. Our empirical result validates our regret analysis and demonstrates that advantage of the fictitious estimates for decreasing the regret.

In the third part of the dissertation, considering the practical scenario that some selected arms may be deliberately or accidentally nullified, we study a novel and challenging contextual combinatorial bandit setting with arm removal and submodular utility. We propose a novel online bandit algorithm, called R2C2-MAB, to robustly select arms to maximize the worst-case submodular utility while balancing exploration and exploitation. Importantly, we prove that R2C2-MAB achieves a sublinear regret in time compared to an efficient baseline algorithm. To empirically evaluate R2C2-MAB, we consider the wireless sniffer channel assignment problem as a concrete example. Under both stochastic and adversarial arm removals, our simulation results show that R2C2-MAB achieves a total reward close to that of the baseline, while outperforming other existing bandit algorithms that either do not exploit the submodularity structure of the utility function or neglect the presence of arm removal.

In the last part, we consider the federated setting of bandit learning with differential privacy and non-parameterized reward function. For this setting, we propose a centralized federated bandit algorithm to learn the environment collaboratively with private communication, in order to minimize the total cumulative regret. We derive an upper bound of the expected cumulative regret incurred by our proposed algorithm, and show that the bound grows sub-linearly with respect to privacy budget. We apply the proposed algorithm to contextual COVID-19 admission decision problem. The simulation results further validate superiority of proposed federated bandit algorithms.

Bibliography

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NeurIPS*, pages 2312–2320, 2011.

[2] Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *arXiv preprint arXiv:2003.12880*, 2020.

[3] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *International Conference on Machine Learning*, pages 32–40. PMLR, 2017.

[4] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, pages 39–1, 2012.

[5] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, pages 127–135, 2013.

[6] Md Martuza Ahamad, Sakifa Aktar, Md Rashed-Al-Mahfuz, Shahadat Uddin, Pietro Liò, Haoming Xu, Matthew A Summers, Julian MW Quinn, and Mohammad Ali Moni. A machine learning model to identify early stage symptoms of sars-cov-2 infected patients. *Expert systems with applications*, 160:113661, 2020.

[7] Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. A neural networks committee for the contextual bandit problem. In *ICONIP*, pages 374–381. Springer, 2014.

[8] Jason Altschuler, Victor-Emmanuel Brunel, and Alan Malek. Best arm identification for contaminated bandits. *Journal of Machine Learning Research*, 20(91):1–39, 2019.

[9] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[10] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[11] Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *COLT*, 2016.

[12] Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *STOC*, pages 45–53, 2004.

[13] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. Resourceful contextual bandits. In *COLT*, pages 1109–1134, 2014.

[14] Sandilya Bhamidipati, Branislav Kveton, and S Muthukrishnan. Minimal interaction search: Multi-way search with item categories. In *AAAI Workshop*, pages 9–15. AI Access Foundation, 2013.

[15] Adarsh Bhimraj, Rebecca L Morgan, Amy Hirsch Shumaker, Valery Lavergne, Lindsey Baden, Vincent Chi-Chung Cheng, Kathryn M Edwards, Rajesh Gandhi, William J Muller, John C O'Horo, et al. Infectious diseases society of america guidelines on the treatment and management of patients with covid-19. *Clinical Infectious Diseases*, 2020.

[16] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Online exp3 learning in adversarial bandits with delayed feedback. In *NeurIPS*, pages 11345–11354, 2019.

[17] Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. *arXiv preprint arXiv:1706.04918*, 2017.

[18] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[19] Djallel Bouneffouf and Irina Rish. A survey on practical applications of multi-armed and contextual bandits. In *arXiv*, 2019, `https://arxiv.org/abs/1904.10040`.

[20] Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. Context attentive bandits: Contextual bandit with restricted context. *arXiv preprint arXiv:1705.03821*, 2017.

[21] Djallel Bouneffouf, Sohini Upadhyay, and Yasaman Khazaeni. Contextual bandit with missing rewards. *arXiv preprint arXiv:2007.06368*, 2020.

[22] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *SIAM*, pages 1433–1452. SIAM, 2014.

[23] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. " you might also like:" privacy risks of collaborative filtering. In *2011 IEEE symposium on security and privacy*, pages 231–246. IEEE, 2011.

[24] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[25] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In *COLT*, pages 750–773, 2018.

[26] Nicolo Cesa-Bianchi, Claudio Gentile, Yishay Mansour, and Alberto Minora. Delay and cooperation in nonstochastic bandits. In *COLT*, volume 49, pages 605–622, 2016.

[27] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. *arXiv preprint arXiv:1306.0811*, 2013.

[28] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[29] Lixing Chen, Zhuo Lu, Pan Zhou, and Jie Xu. Learning optimal sniffer channel assignment for small cell cognitive radio networks. In *INFOCOM*, pages 656–665. IEEE, 2020.

[30] Lixing Chen, Jie Xu, and Zhuo Lu. Contextual combinatorial multi-armed bandits with volatile arms and submodular reward. *NeurIPS*, 31:3247–3256, 2018.

[31] Lixing Chen, Jie Xu, Shaolei Ren, and Pan Zhou. Spatio–temporal edge service placement: A bandit learning approach. *IEEE Transactions on Wireless Communications*, 17(12):8388–8401, 2018.

[32] Wei Chen, Yihan Du, Longbo Huang, and Haoyu Zhao. Combinatorial pure exploration for dueling bandit. In *ICML*, 2020.

[33] S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. *ICML*, 2017.

[34] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 844–853. JMLR. org, 2017.

[35] W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. *NeurIPS*, 2011.

[36] A. A. Deshmukh, U. Dogan, and C. Scott. Multi-task learning for contextual bandits. *NeurIPS*, 2017.

[37] Abhimanyu Dubey and Alex Pentland. Differentially-private federated linear bandits. *arXiv preprint arXiv:2010.11425*, 2020.

[38] Abhimanyu Dubey and Alex Pentland. Private and byzantine-proof cooperative decision-making. In *AAMAS*, pages 357–365, 2020.

[39] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine learning for healthcare conference*, pages 67–82. PMLR, 2018.

[40] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[41] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[42] Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *FOCS*, 2011.

[43] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. An analysis of approximations for maximizing submodular set functions—ii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

[44] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *2010 IEEE DySPAN*, pages 1–9. IEEE, 2010.

[45] Phanideep Gampa and Sumio Fujita. Banditrank: Learning to rank using contextual bandits. *arXiv preprint arXiv:1910.10410*, 2019.

[46] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. In *NeurIPS*, 2019.

[47] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[48] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *SIAM*, pages 1098–1116. SIAM, 2011.

[49] Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*, 2014.

[50] Ziwei Guan, Kaiyi Ji, Donald J Bucci Jr, Timothy Y Hu, Joseph Palombo, Michael Liston, and Yingbin Liang. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *AAAI*, 2020.

[51] Xiaoguang Huo and Feng Fu. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11):171377, 2017.

[52] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

[53] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[54] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu. Adversarial attacks on stochastic bandits. In *NIPS*, 2018.

[55] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[56] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust bayesian optimization. In *AISTATS*, 2020.

[57] Johannes Kirschner and Andreas Krause. Stochastic bandits with context distributions. In *NeurIPS*, 2019.

[58] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[59] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[60] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[61] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference (ECC)*, pages 243–248. IEEE, 2016.

[62] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *NeurIPS*, pages 817–824, 2008.

[63] Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *STOC*, pages 323–332, 2009.

[64] Fengjiao Li, Jia Liu, and Bo Ji. Combinatorial sleeping bandits with fairness constraints. In *INFOCOM*, 2019.

[65] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670. ACM, 2010.

[66] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *ICML*, volume 16, pages 1245–1253, 2016.

[67] Tan Li, Linqi Song, and Christina Fragouli. Federated recommendation system via differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2592–2597. IEEE, 2020.

[68] Baihan Lin, Guillermo Cecchi, Djallel Bouneffouf, and Irina Rish. Adaptive representation selection in contextual bandit. *arXiv preprint arXiv:1802.00981*, 2018.

[69] Christopher H Lin, Ece Kamar, and Eric Horvitz. Signals in the silence: Models of implicit feedback in a recommendation system for crowdsourcing. In *AAAI*. Citeseer, 2014.

[70] Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *ICML*, 2019.

[71] Bingqian Lu, Jianyi Yang, Lydia Y Chen, and Shaolei Ren. Automating deep neural network model selection for edge inference. In *CogMI*, pages 184–193. IEEE, 2019.

[72] Shiyin Lu, Guanghui Wang, Yao Hu, and Lijun Zhang. Multi-objective generalized linear bandits. *arXiv preprint arXiv:1905.12879*, 2019.

[73] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *STOC*, 2018.

[74] Mohammad Malekzadeh, Dimitrios Athanasakis, Hamed Haddadi, and Benjamin Livshits. Privacy-preserving bandits. *arXiv preprint arXiv:1909.04421*, 2019.

[75] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. Decentralized cooperative stochastic multi-armed bandits. *arXiv preprint arXiv:1810.04468*, 2018.

[76] Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits and recommender systems. In *LOD*, pages 325–336. Springer, 2015.

[77] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[78] Nikita Mishra and Abhradeep Thakurta. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 592–601, 2015.

[79] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical programming*, 14(1):265–294, 1978.

[80] G. Neu and J. Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. *arXiv preprint arXiv:2002.00287*, 2020.

[81] Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. Online markov decision processes under bandit feedback. In *NeurIPS*, pages 1804–1812, 2010.

[82] James B Orlin, Andreas S Schulz, and Rajan Udwani. Robust monotone submodular function maximization. *Mathematical Programming*, 172(1-2):505–537, 2018.

[83] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

[84] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *SIAM*, pages 461–469. SIAM, 2014.

[85] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[86] Wenbo Ren, Xingyu Zhou, Jia Liu, and Ness B Shroff. Multi-armed bandits with local differential privacy. *arXiv preprint arXiv:2007.03121*, 2020.

[87] Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic DecisionTheory*, pages 18–34. Springer, 2017.

[88] Aadirupa Saha, Pierre Gaillard, and Michael Valko. Improved sleeping bandits with stochastic actions sets and adversarial rewards. *arXiv preprint arXiv:2004.06248*, 2020.

[89] Vidit Saxena, Joakim Jaldén, Joseph E Gonzalez, Mats Bengtsson, Hugo Tullberg, and Ion Stoica. Contextual multi-armed bandits for link adaptation in cellular networks. In *Proceedings of the 2019 Workshop on Network Meets AI & ML*, pages 44–49, 2019.

[90] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. *arXiv preprint arXiv:1810.00068*, 2018.

[91] Chengshuai Shi, Cong Shen, and Jing Yang. Federated multi-armed bandits with personalization. *arXiv preprint arXiv:2102.13101*, 2021.

[92] Dong-Hoon Shin, Saurabh Bagchi, and Chih-Chun Wang. Distributed online channel assignment toward optimal monitoring in multi-channel wireless networks. In *INFOCOM*, pages 2626–2630. IEEE, 2012.

[93] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. *ICML*, 2010.

[94] Sírio-Libanês. Clinical data to assess diagnosis. *Kaggle*. *https://www.kaggle.com/S%C3%ADrio-Libanes/covid19*, 2020.

[95] V. Syrgkanis, A. Krishnamurthy, and R. Schapire. Efficient algorithms for adversarial contextual learning. *ICML*, 2016.

[96] Sho Takemori, Masahiro Sato, Takashi Sonoda, Janmajay Singh, and Tomoko Ohkuma. Submodular bandit problem under multiple constraints. In *UAI*, 2020.

[97] Jing Tang, Xueyan Tang, and Junsong Yuan. An efficient and effective hop-based approach for influence maximization in social networks. *Social Network Analysis and Mining*, 8(1):10, 2018.

[98] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *CIKM*, pages 1587–1594. ACM, 2013.

[99] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. *arXiv preprint arXiv:1906.00670*, 2019.

[100] Aristide Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[101] Aristide Tossou and Christos Dimitrakakis. Achieving privacy in the adversarial multi-armed bandit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[102] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. *UAI*, 2013.

[103] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[104] Claire Vernade, Alexandra Carpentier, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Contextual bandits under delayed feedback. *arXiv preprint arXiv:1807.02089*, 2018.

[105] Sofía S Villar, Jack Bowden, and James 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*, 30(2):199, 2015.

[106] Eric A Wan. Neural network classification: A bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(4):303–305, 1990.

[107] H. Wang, Q. Wu, and H. Wang. Learning hidden features for contextual bandits. *CIKM*, 2016.

[108] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *CIKM*, pages 1633–1642. ACM, 2016.

[109] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: How much communication is needed to achieve (near) optimal regret. *arXiv preprint arXiv:1904.06309*, 2019.

[110] Nirandika Wanigasekara, Yuxuan Liang, Siong Thye Goh, Ye Liu, Joseph Jay Williams, and David S Rosenblum. Learning multi-objective rewards and user utility function in contextual bandits for personalized ranking. In *IJCAI*, pages 3835–3841. AAAI Press, 2019.

[111] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538, 2016.

[112] Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *ICML*, 2016.

[113] Saba Q Yahyaa, Madalina M Drugan, and Bernard Manderick. Thompson sampling in the adaptive linear scalarized multi objective multi armed bandit. In *ICAART (2)*, pages 55–65, 2015.

[114] J. Yang and S. Ren. Bandit learning with predicted context: Regret analysis and selective context query. In *INFOCOM*, 2021.

[115] J. Yang and S. Ren. Robust bandit learning with imperfect context. In *AAAI*, 2021.

[116] L. Yang, J. Yang, and S. Ren. Multi-feedback bandit learning with probabilistic contexts. In *IJCAI*, 2020.

[117] Lin Yang, Mohammad Hajiesmaili, Mohammad Sadegh Talebi, John C. S. Lui, and Wing Shing Wong. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In *NeurIPS*, 2020.

[118] Se-Young Yun, Jun Hyun Nam, Sangwoo Mo, and Jinwoo Shin. Contextual multi-armed bandits under feature uncertainty. *arXiv preprint arXiv:1703.01347*, 2017.

[119] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 2025–2034, 2016.

[120] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[121] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In *NeurIPS*, pages 5198–5209, 2019.

[122] Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–29, 2021.

Appendix A

Supplemental Proof

A.0.1 Proof of Lemma 8

Proof. Since Eqn. (3.8) has another expression by Eqn. (3.6), we have

$$\begin{aligned}|\hat{g}_{a,t} - g_{a,t}| &= |\phi\left(x_{a,t}\right)^{\top}\theta - \phi(x_{a,t})^{\top}\mathbf{C}_t{}^{-1}\mathbf{\Phi}_t\mathbf{y}_t| \\ &= |\lambda\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-1}\theta| + |\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-1}\mathbf{\Phi}_t\left(\mathbf{y}_t - \Phi_t^{\top}\theta\right)| \\ &\leq \lambda\|\mathbf{C}_t^{-1}\phi\left(x_{a,t}\right)\| + |\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-1}\mathbf{\Phi}_t\left(\mathbf{y}_t - \Phi_t^{\top}\theta\right)|\end{aligned} \tag{A.1}$$

where the last inequality comes from Cauchy-Schwartz inequality.

For the first term in Eqn. (A.1), since $\mathbf{C}_t$ is a positive definite matrix, we have

$$\begin{aligned}\|\mathbf{C}_t^{-1}\phi\left(x_{a,t}\right)\| &= \sqrt{\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-2}\phi\left(x_{a,t}\right)} \\ &\leq \sqrt{\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-1}\phi\left(x_{a,t}\right)} = w_{a,t}.\end{aligned}$$

For the second term in Eqn. (A.1), since $\mathbb{E}\left[\left(\mathbf{y}_t - \Phi_t^{\top}\theta\right)\right] = 0$, by Azuma's inequality,

$$\begin{aligned}&\Pr\left(|\phi\left(x_{a,t}\right)^{\top}\mathbf{C}_t^{-1}\mathbf{\Phi}_t\left(\mathbf{y}_t - \Phi_t^{\top}\theta\right)| \geq \alpha w_{a,t}\right) \\ &\quad\leq 2\exp\left(-\frac{2\alpha^2 w_{a,t}^2}{\|\mathbf{\Phi}_t^{\top}\mathbf{C}_t^{-1}\phi\left(x_{a,t}\right)\|^2}\right) \leq 2\exp\left(-2\alpha^2\right)\end{aligned}$$

where the last inequality is because

$$\begin{aligned} w_{a,t}^2 &= \phi\left(x_{a,t}\right)^\top \mathbf{C}_t^{-1} \phi\left(x_{a,t}\right) \\ &= \phi\left(x_{a,t}\right)^\top \mathbf{C}_t^{-1} \left(\mathbf{\Phi}_t \mathbf{\Phi}_t{}^\top + \lambda \mathbf{I}\right) \mathbf{C}_t^{-1} \phi\left(x_{a,t}\right) \\ &\geq \phi\left(x_{a,t}\right)^\top \mathbf{C}_t^{-1} \mathbf{\Phi}_t \mathbf{\Phi}_t{}^\top \mathbf{C}_t^{-1} \phi\left(x_{a,t}\right) \\ &= \|\mathbf{\Phi}_t^\top \mathbf{C}_t^{-1} \phi\left(x_{a,t}\right)\|^2 \end{aligned}$$

Letting $\frac{\delta}{TK} = 2\exp\left(-2\alpha^2\right)$, with probability at least $1-\frac{\delta}{T}$, we have $|\phi\left(x_{a,t}\right)^\top \mathbf{C}_t^{-1} \mathbf{\Phi}_t \left(\mathbf{y}_t - \Phi_t^\top \theta\right)| \leq \alpha w_{a,t}$. This completes the proof. ■

A.0.2 Proof of Lemma 10

Proof. By the definition of instant regret in Eqn. (3.3), we have

$$\begin{aligned} reg_t &= \mathbb{E}\left[y_{a_t^*,t} - y_{a_t,t}\right] \\ &= g_{a_t^*,t} - \hat{g}_{a_t^*,t} + \hat{g}_{a_t^*,t} - g_{a_t,t} \\ &\leq \left(\alpha + \lambda\right) w_{a_t^*,t} + \hat{g}_{a_t^*,t} - g_{a_t,t} \\ &\leq \left(\alpha + \lambda\right) w_{a_t,t} + \hat{g}_{a_t,t} - g_{a_t,t} \\ &\leq 2\left(\alpha + \lambda\right) w_{a_t,t} \end{aligned}$$

where the first inequality and the third inequality hold based on Lemma 8 which bounds the reward estimation error for all arms, and the second inequality comes from the UCB-based arm selection in Algorithm 2. ■

A.0.3 Proof of Lemma 11

Proof. Divide the $T-\tau_{max}$ rounds into τ_{max} groups, each with m elements. In this way, the p-th round set, $p \in \mathbb{Z}^+, p \in [1, \tau_{max}]$, is $\Omega^p = \{\tau_{max} + p, 2\tau_{max} + p, \cdots, m\tau_{max} + p\}$.

Correspondingly, the contexts with respect to the selected arms are also divided into τ_{max} groups, each group with m elements. For example, in the pth context group , $p \in \mathbb{Z}^+, p \in [1, \tau_{max}]$, the contexts are $\{\phi(\bar{x}_{\tau_{max}+p}), \phi(\bar{x}_{2\tau_{max}+p}), \cdots, \phi(\bar{x}_{m\tau_{max}+p})\}$ where $\bar{x}_{s\tau_{max}+p} = x_{s\tau_{max}+p, a_{s\tau_{max}+p}}$.

Recall that $\mathbf{C}_t$ in Eqn. (3.4.1) can also be written as $\mathbf{C}_t = \sum_{s=1}^{t-1} \phi(\bar{x}_s)\phi(\bar{x}_s)^\top + \lambda\mathbf{I}$. For each group, we construct m matrices with the similar form as $\mathbf{C}_t$, which are

$$W_i^p = \lambda\mathbf{I} + \sum_{s=1}^{i-1} \phi(\bar{x}_{s\tau_{\max}+p})\phi(\bar{x}_{s\tau_{\max}+p})^\top,$$

$$i, p \in \mathbb{Z}^+, \quad p \in [1, \tau_{max}], \quad i \in [1, m].$$

By using Lemma 11 in [1], we have

$$\sum_{s=1}^{m} \|\phi(\bar{x}_{s\tau_{max}+p})\|^2_{(W_s^p)^{-1}} \le 2\log\frac{\det(W_m^p)}{\det(\lambda\mathbf{I})} \tag{A.2}$$

and

$$\sum_{s=1}^{m} \|\phi(\bar{x}_{s\tau_{max}+p})\|^2_{(W_s^p)^{-1}} \le 2\log\frac{\det(W_m^p)}{\det(\lambda\mathbf{I})}.$$

Since the feedback delay d_t is no larger than τ_{max}, the reward for arm selected at round t must be fed back at round $t + \tau_{\max}$. Thus, we have $\forall t > \tau_{max}, \mathcal{T}_{t-\tau_{max}} \subseteq \mathcal{T}_t$. To help analyze, we let $\Omega_i^p = \{\tau_{max} + p, 2\tau_{max} + p, \cdots, i\tau_{max} + p\}$ for $i, p \in \mathbb{Z}^+, i \le m$. If $t = i\tau_{max} + p$, then $\Omega_{i-1}^p \subset \mathcal{T}_{t-\tau_{max}} \subseteq \mathcal{T}_t$. Since $\mathbf{C}_t$ and W_i^p are both positive-definite matrices, for any $t = i\tau_{max} + p$, we have

$$\begin{aligned}
&\phi(\bar{x}_t)^\top \mathbf{C}_t^{-1} \phi(\bar{x}_t) \\
=&\phi(\bar{x}_t)^\top \left(\lambda \mathbf{I} + \sum_{s \in \mathcal{T}_t} \phi(x_{s,a_s}) \phi(x_{s,a_s})^\top \right)^{-1} \phi(\bar{x}_t) \\
\leq&\phi(\bar{x}_t)^\top \left(\lambda \mathbf{I} + \sum_{s \in \Omega_{i-1}^p} \phi(x_{s,a_s}) \phi(x_{s,a_s})^\top \right)^{-1} \phi(\bar{x}_t) \\
=&\phi(\bar{x}_t)^\top \left(W_{i-1}^p \right)^{-1} \phi(\bar{x}_t).
\end{aligned}$$

Therefore, the sum of $w_{a_t,t}^2$ can be expressed as

$$\begin{aligned}
\sum_{t=\tau_{max}+1}^{T} ||\phi(x_{a,t})||_{\mathbf{C}_t^{-1}}^2 &= \sum_{p=1}^{\tau_{max}} \sum_{s=1}^{m} \|\phi(\bar{x}_{s\tau_{max}+p})\|_{\left(W_{s-1}^p\right)^{-1}} \\
&\leq 2 \sum_{p=1}^{\tau_{max}} \log \frac{\det\left(W_m^p\right)}{\det\left(\lambda \mathbf{I}\right)}.
\end{aligned}$$

Let K_m^p be the $m \times m$ kernel matrix with respect to the pth context group. By Sylvester's determinant theorem, we have

$$\begin{aligned}
\log \frac{\det\left(W_m^p\right)}{\det\left(\lambda \mathbf{I}\right)} &= \log \frac{\det\left(\mathbf{I} + K_m^p\right)}{\det\left(\lambda \mathbf{I}\right)} \\
&\leq d_p \log(1 + \frac{mc_k}{d_p \lambda}),
\end{aligned}$$

where d_p is the rank of K_m^p. Define the effective dimension as $d = \arg\max_{d_p = d_1, \cdots, d_{\tau_{\max}}} (d_p \log(1 + \frac{mc_k}{d_p \lambda}))$. Then sum of $w_{a_t,t}^2$ can be bounded as

$$\begin{aligned}
\sum_{t=\tau_{max}+1}^{T} ||\phi(x_{a,t})||_{\mathbf{C}_t^{-1}}^2 &\leq 2 \sum_{p=1}^{\tau_{max}} d_p \log(1 + \frac{mc_k}{d_p \lambda}) \\
&\leq 2\tau_{max} d \log(1 + \frac{mc_k}{d\lambda}).
\end{aligned}$$

■

A.0.4 Proof of Lemma 12

Proof. It is same as proof of Lemma 8, we have

$$\begin{aligned}|\bar{g}_{a,t} - g_{a,t}| &= |\phi\left(x_{a,t}\right)^{\top}\theta - \phi(x_{a,t})^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\bar{\mathbf{y}}_t| \\ &\leq \lambda\|\bar{\mathbf{C}}_t^{-1}\phi\left(x_{a,t}\right)\| + |\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)|\end{aligned} \tag{A.3}$$

where the last inequality comes from Cauchy-Schwartz inequality.

For the first term in Eqn. (A.3), since $\mathbf{C}_t$ is a positive definite matrix and $\lambda \geq 1$, we have

$$\|\bar{\mathbf{C}}_t^{-1}\phi\left(x_{a,t}\right)\| \leq \sqrt{\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\phi\left(x_{a,t}\right)} = \bar{w}_{a,t}.$$

For the second term in Eqn. (A.3), different from Lemma 1, since $\mathbb{E}\left[\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)\right] \neq 0$, by Azuma's inequality,

$$\begin{aligned}&\Pr\left(|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)| - |\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\mathbb{E}\left[\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)\right]| \geq \alpha\bar{w}_{a,t}\right) \\ &\leq 2\exp\left(-\frac{2\alpha^2\bar{w}_{a,t}^2}{\|\bar{\mathbf{\Phi}}_t^{\top}\bar{\mathbf{C}}_t^{-1}\phi\left(x_{a,t}\right)\|^2}\right) \leq 2\exp\left(-2\alpha^2\right)\end{aligned}$$

where the last inequality is because

$$\bar{w}_{a,t}^2 \geq \|\bar{\mathbf{\Phi}}_t^{\top}\bar{\mathbf{C}}_t^{-1}\phi\left(x_{a,t}\right)\|^2.$$

As for $|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\mathbb{E}\left[\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)\right]|$, we assume entry in vector $\phi(x_{a,t})^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t$ is bounded by $|V_{max}|$ and we have

$$
\begin{aligned}
|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\mathbb{E}\left[\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)\right]| &\leq |V_{max}\sum_{\bar{y}^k\in\bar{\mathbf{y}}_t}\mathbb{E}\left[\bar{y}^k - \phi(x^k)^{\top}\theta\right]| \\
&= |V_{max}\sum_{\hat{y}^k\in\hat{\mathbf{y}}_t}\mathbb{E}\left[\hat{y}^k - \phi(x^k)^{\top}\theta\right]| \\
&\leq |V_{max}\sum_{\hat{y}^k\in\hat{\mathbf{y}}_t}\left(\mathbb{E}\left[\tilde{y}^i - \phi(x^k)^{\top}\theta\right] + L\kappa_t^{\beta}\right)| \qquad \text{(A.4)} \\
&\leq |V_{max}||\hat{\mathbf{y}}_t|L\kappa_t^{\beta} \\
&\leq L|V_{max}|t\kappa_t^{\beta}.
\end{aligned}
$$

The second inequality in Eqn. (A.4) is from the fictitious feedback criteria in Eqn. (3.12) and further we have,

$$
\begin{aligned}
&\Pr\left(|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)| - L|V_{max}|t\kappa_t^{\beta} \geq \alpha\bar{w}_{a,t}\right) \\
&\leq \Pr\left(|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)| - |\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\mathbb{E}\left[\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)\right]| \geq \alpha\bar{w}_{a,t}\right).
\end{aligned}
$$

Letting $\frac{\delta}{TK} = 2\exp\left(-2\alpha^2\right)$, with probability at least $1-\frac{\delta}{T}$, we have $|\phi\left(x_{a,t}\right)^{\top}\bar{\mathbf{C}}_t^{-1}\bar{\mathbf{\Phi}}_t\left(\bar{\mathbf{y}}_t - \bar{\mathbf{\Phi}}_t^{\top}\theta\right)|$ $\alpha\bar{w}_{a,t} + L|V_{max}|t{\kappa_t}^{\beta}$. This completes the proof. ■

A.0.5 Proof of Theorem 13

Proof. Since the instant regret from Lemma 12 is bounded with probability $1-\frac{\delta}{T}$

$$
reg_t \leq 2(\alpha+\lambda)\bar{w}_{a,t} + 2L|V_{max}|t{\kappa_t}^{\beta},
$$

then the total cumulative regret by Algorithm 3 is bounded up to time T, with probability $1-\delta$

$$
R_T \leq 2(\alpha+\lambda)\sum_{t=1}^{T}\bar{w}_{a,t} + 2L|V_{max}|\sum_{t=1}^{T}t{\kappa_t}^{\beta}.
$$

For the first part $\sum_{t=1}^{T}(\alpha + \lambda)\bar{w}_{a,t}$, based on [36], we have

$$\sum_{t=1}^{T} \bar{w}_{a,t} \leq \sqrt{2Td \log\Big(1 + \frac{c_k}{d\lambda}T\Big)}.$$

For the rest, by letting ${\kappa_t}^{\beta} = \zeta t^{\frac{-3}{\beta}}$, $\sum_{t=1}^{T} t{\kappa_t}^{\beta}$ is bounded by

$$\sum_{t=1}^{T} t{\kappa_t}^{\beta} = \sum_{t=1}^{T} t \cdot \zeta(t^{\frac{-3}{\beta}})^{\beta} = \zeta \sum_{t=1}^{T} t^{-2} \leq \zeta \sum_{t=1}^{\infty} t^{-2} = \zeta \frac{\pi^2}{6}.$$

This completes the proof. ■

A.0.6 Proof of Lemma 17

Proof. In R2C2-MAB, when V_{sub}^{t} occurs, it indicates that the utility of selecting arms in $\tilde{\mathcal{N}}_{sub}^{t}$ is higher than the utility of selecting arms in $\tilde{\mathcal{N}}_{g}^{t}$. Thus, we have

$$Pr\{V_{sub}^{t}\} = Pr\{u(\hat{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) \geq u(\hat{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t})\}. \tag{A.5}$$

The right side of Equation (A.5) indicates that at least one of three following events happens when $H(t) \geq 0$:

$$\begin{aligned}
E_1 =& \{u(\hat{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) \geq u(\bar{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) + H(t)\} \\
E_2 =& \{u(\hat{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t}) \leq u(\underline{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t}) - H(t)\} \\
E_3 =& \{u(\hat{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) \geq u(\hat{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t}), \\
& u(\hat{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) < u(\bar{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) + H(t), \\
& u(\hat{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t}) > u(\underline{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t}) - H(t)\}.
\end{aligned} \tag{A.6}$$

Hence, we have

$$\{u(\hat{\boldsymbol{d}}_{sub}^{t}, \tilde{\mathcal{N}}_{sub}^{t}) \geq u(\hat{\boldsymbol{d}}_{g}^{t}, \tilde{\mathcal{N}}_{g}^{t})\} \subseteq E_1 \cup E_2 \cup E_3. \tag{A.7}$$

The next step is to bound the probability of E_1, E_2 and E_3 separately. For $Pr\{E_1\}$, we have

$$\begin{aligned}
Pr\{E_1\} &= Pr\{u(\hat{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) \geq u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) + H(t)\} \\
&\leq Pr\{\hat{d}(p^t_n) \geq \bar{d}(p^t_n) + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\} \\
&\leq Pr\{\hat{d}(p^t_n) \geq \mathbb{E}[\hat{d}(p^t_n)] + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\} \\
&= \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} Pr\{\hat{d}(p^t_n) \geq \mathbb{E}[\hat{d}(p^t_n)] + \frac{H(t)}{\tilde{b}}\} \\
&\leq \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} \exp\left(\frac{-2C^t(\hat{p}^t_n)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \sum_{n \in \tilde{\mathcal{N}}^t_{sub}} \exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \tilde{b}\exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right)
\end{aligned} \tag{A.8}$$

The first inequality of Equation (A.8) comes from $\{u(\hat{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) \geq u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) + H(t)\} \subseteq \{\hat{d}(p^t_n) \geq \bar{d}(p^t_n) + \frac{H(t)}{\tilde{b}}, \exists n \in \tilde{\mathcal{N}}^t_{sub}\}$, which can be proved by *reductio ad absurdum*. The last three steps of Equation (A.8) utilize the Chernoff-Hoeffding bound. Since this is in the exploitation phase, there are least $t^z\log(t)$ times counted in $C(p), \forall p \in \mathcal{P}_t$. To bound $Pr\{E_1\}$, we choose $H(t) = \tilde{b}d_{max}t^{\frac{-z}{2}}$ and then have

$$\begin{aligned}
Pr\{E_1\} &\leq \tilde{b}\exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\
&= \tilde{b}\exp\left(\frac{-2t^z\log(t)(\tilde{b}d_{max}t^{\frac{-z}{2}})^2}{(\tilde{b}d_{max})^2}\right) \\
&\leq \tilde{b}\exp(-2\log(t)) \\
&\leq \tilde{b}t^{-2}.
\end{aligned} \tag{A.9}$$

Similarly, the $Pr\{E_2\}$ can be bounded in the same way.

$$\begin{aligned} Pr\{E_2\} &\leq \tilde{b}\exp\left(\frac{-2t^z\log(t)H(t)^2}{(\tilde{b}d_{max})^2}\right) \\ &\leq \tilde{b}t^{-2}. \end{aligned} \tag{A.10}$$

Finally, to bound $Pr\{E_3\}$, we define the best and worst context for hypercube p as $\bar{x}(p) := \arg\max_{x\in p} d(x)$ and $\underline{x}(p) := \arg\min_{x\in p} d(x)$, respectively. Thus, we redefine $\bar{d}(p)$ and $\underline{d}(p)$ as:

$$\begin{aligned} \bar{d}(p) &= \frac{1}{C^t(p)}\sum_{\tau:x_\tau\in p} d(\bar{x}_\tau(p)) \\ \underline{d}(p) &= \frac{1}{C^t(p)}\sum_{\tau:x_\tau\in p} d(\underline{x}_\tau(p)) \end{aligned} \tag{A.11}$$

where x_τ are the contexts falling into the hypercube p before time slot t. Further, we have $\forall p \in P^t$:

$$\begin{aligned} \bar{d}(p) - \hat{d}(p) &\leq \frac{1}{C^t(p)}\sum_{\tau:x_\tau\in p} d(\bar{x}_\tau(p)) - d(x_\tau(p)) \\ &\leq \frac{1}{C^t(p)}\sum_{\tau:x_\tau\in p} LD^{\frac{\alpha}{2}}h_T^{-\alpha} \\ &\leq LD^{\frac{\alpha}{2}}h_T^{-\alpha} \end{aligned} \tag{A.12}$$

where $LD^{\frac{\alpha}{2}}h_T^{-\alpha}$ comes from the Hölder condition defined in Equation (4.4) and the fact $||x - x'|| \leq D^{\frac{1}{2}}h_T^{-1}$ due to uniform hypercube partition. Likewise, we have

$$\hat{d}(p) - \underline{d}(p) \leq LD^{\frac{\alpha}{2}}h_T^{-\alpha}. \tag{A.13}$$

Considering the arms in $\tilde{N}^t_{sub}$, by the greedy algorithm, we have:

$$\begin{aligned} u(\bar{\boldsymbol{d}}^t_{sub}, \tilde{\mathcal{N}}^t_{sub}) - u(\hat{\boldsymbol{d}}^t_{sub}\tilde{\mathcal{N}}^t_{sub}) &\leq \sum_{n\in\tilde{\mathcal{N}}^t_{sub}} \hat{d}(p^t_n) - \underline{d}(p^t_n) \\ &\leq \tilde{b}LD^{\frac{\alpha}{2}}h_T^{-\alpha} \end{aligned} \tag{A.14}$$

Similarly, for $\tilde{\mathcal{N}}_g^t$, we have

$$u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) \leq \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha}. \tag{A.15}$$

Next, we analyze the three components of E_3 separately. For the first component, according to the definition in Equation (A.11), we have

$$\{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t)\} \subseteq \{u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t)\} \tag{A.16}$$

For the second part of E_3, by Equation (A.12), we have

$$\begin{aligned} &\{u(\hat{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) < u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) + H(t)\} \\ &\subseteq \{u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) - \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha} < u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) + H(t)\} \\ &= \{H(t) + \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha} > 0\}. \end{aligned} \tag{A.17}$$

For the last component of E_3, using Equation (A.12) again, we have

$$\begin{aligned} &\{u(\hat{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) > u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - H(t)\} \\ &\subseteq \{u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) + \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha} > u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - H(t)\} \\ &= \{H(t) + \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha} > 0\}. \end{aligned} \tag{A.18}$$

Combining Equations (A.16), (A.17) and (A.18), the probability of E_3 is bounded by

$$Pr\{E_3\} \leq Pr\{u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t), H(t) + \tilde{b} L D^{\frac{\alpha}{2}} h_T^{-\alpha} > 0\}. \tag{A.19}$$

Since the utility function satisfies

$$u(\underline{\boldsymbol{d}}_g^t, \tilde{\mathcal{N}}_g^t) - u(\bar{\boldsymbol{d}}_{sub}^t, \tilde{\mathcal{N}}_{sub}^t) \geq A t^{\theta} \tag{A.20}$$

where $A > 0$ and $\theta < 0$, it follows that the right side of Inequality (A.19) contradicts with (A.20), which means that $Pr\{E_3\} = 0$ under the condition specified by (A.20).

Therefore, combining (A.9) and (A.10), we have

$$\begin{aligned} Pr\{V_{sub}^t\} &\leq Pr\{E_1 \cup E_2 \cup E_3\} \\ &\leq Pr\{E_1\} + Pr\{E_2\} + Pr\{E_3\} \\ &= 2 \cdot \tilde{b}t^{-2}. \end{aligned} \tag{A.21}$$

This completes the proof. ■

A.0.7 Proof of Proposition 20

Proof. To demonstrate the applicability of R2C2-MAB for the sniffer assignment problem formulated in Equation (4.28), it suffices to show that the utility function defined in Equation (4.27) is monotone and submodular.

First, we let $\mathcal{N}_1 \subseteq \mathcal{N}, \mathcal{N}_2 \subseteq \mathcal{N}$ and $\mathcal{N}_1 \subseteq \mathcal{N}_2$. Thus, for $\mathcal{N}_2$, we have

$$\begin{aligned} \prod_{n \in \mathcal{N}_2} (1 - Pr_n^t) &= \prod_{n' \in \mathcal{N}_1} (1 - Pr_{n'}^t) \prod_{n^\dagger \in \mathcal{N}_2 \backslash \mathcal{N}_1} (1 - Pr_{n^\dagger}^t) \\ &\leq \prod_{n' \in \mathcal{N}_1} (1 - Pr_{n'}^t) \end{aligned} \tag{A.22}$$

Then, it follows that $1 - \prod_{n \in \mathcal{N}_2} (1 - Pr_n^t) \geq 1 - \prod_{n' \in \mathcal{N}_1} (1 - Pr_{n'}^t)$, proving monotonicity.

Next, for submodularity, we let $\tilde{n} \in \mathcal{N} \backslash \mathcal{N}_2$ and the marginal utility $\Delta(\mathcal{N}_2, \tilde{n})$ is

$$\begin{aligned} \Delta(\mathcal{N}_2, \tilde{n}) &= \left(1 - \prod_{n \in \mathcal{N}_2 \cup \{\tilde{n}\}} (1 - Pr_n^t)\right) - \left(1 - \prod_{n \in \mathcal{N}_2} (1 - Pr_n^t)\right) \\ &= \prod_{n \in \mathcal{N}_2} (1 - Pr_n^t) - \prod_{n \in \mathcal{N}_2 \cup \{\tilde{n}\}} (1 - Pr_n^t) \\ &= (1 - Pr_{\tilde{n}}^t) \prod_{n \in \mathcal{N}_2} (1 - Pr_n^t) \end{aligned} \tag{A.23}$$

Next, we show

$$\begin{aligned}&\Delta(\mathcal{N}_1, \tilde{n}) - \Delta(\mathcal{N}_2, \tilde{n})\\=&(1 - Pr^t_{\tilde{n}}) \prod_{n' \in \mathcal{N}_1} (1 - Pr^t_{n'}) - (1 - Pr^t_{\tilde{n}}) \prod_{n \in \mathcal{N}_2} (1 - Pr^t_{n})\\=&(1 - Pr^t_{\tilde{n}}) \left(\prod_{n' \in \mathcal{N}_1} (1 - Pr^t_{n'}) - \prod_{n \in \mathcal{N}_2} (1 - Pr^t_{n}) \right) \geq 0\end{aligned} \tag{A.24}$$

where the last inequality is from Equation (A.22). Thus, this complete the submodularity proof. ■