Optimizing Federated Agents For Fairness And Privacy In Bandits

Thesis Proposal submitted in partial
fulfillment of the requirements of the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Sambhav Solanki
2018111008
sambhav.solanki@research.iiit.ac.in

Advised by Dr. Sujit P Gujar

Figures/iiit-eps-converted-to.pdf

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
October, 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Optimizing Federated Agents For Fairness And Privacy In Bandits” by Sambhav Solanki, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Sujit P Gujar

To my parents and my sister.

Acknowledgements

I would like to extend my deepest gratitude to my advisor, Dr. Sujit Gujar, whose guidance, encouragement, and insightful feedback have been invaluable throughout the course of this research. Your expertise and dedication have been instrumental in shaping this work, and I am truly grateful for the time and effort you have invested in my academic journey. Additionally, I wish to thank Dr. Shweta Jain for their support. Your mentorship has been a paramount for my development.

I am also indebted to all the colleagues and friends I made during my time at Machine Learning Lab. The discussions with Sankarshan, Samhita, Pronoy, and Debojit on various topics have been crucial in shaping this work. Sanjay, Varul, Shaily, Manisha, Anurag, Shantanu, Parth, Rashid, I thank you for your continuous support and making my time at the lab enjoyable.

I would also like to shoutout Dolton Fernandes and George Tom for being there throughout and supporting me. You guys will always hold a special place in my heart. To all my extremely close friends: Shantanu, Sridhar, Nikunj, Bhuvanesh, Naren, and Ashish - I owe a lot to you guys and all the memories we made together.

Finally, and most importantly, I acknowledge the pivotal role my family played. To my parents and sister, thank you.

Abstract

Federated learning offers a promising approach to learn collaboratively across multiple devices or agents. This collaboration allows leveraging a vast amount of data while preserving privacy by keeping the raw data local. However, this approach is not void of challenges. This thesis explores some of these challenges in the context of multi-armed bandit (MAB) problems.

MAB problems are a type of decision-making problem where an agent repeatedly chooses between multiple actions, each with an unknown reward, and aims to maximize the total reward over time. Federated MABs introduce unique fairness concerns. For instances, for a fair algorithm, the learning process should not favor certain actions over others to an extreme degree, especially if those actions are beneficial to specific sub-populations. In addition to fairness, privacy is another key challenge. While federated learning avoids sharing raw data, the learning process itself can leak information about individual data points.

This thesis proposes two novel algorithms (and variants) to address these challenges. The first algorithm, P-FCB, tackles a specific type of MAB problem with constraints in a federated setting. It promotes fairness by optimizing for the collective benefit of all users. The second algorithm, Fed-FairX-LinUCB, focuses on achieving fairness and privacy guarantees in a more complex scenario where actions have additional context. It ensures that action selection is fair and avoids situations where some actions are never chosen. Additionally, both these algorithms provide differential privacy guarantees, a formal guarantee that ensures the learning process reveals minimal information about any individual data point.

By exploring these algorithms, this thesis aims to contribute to the development of federated learning for MAB problems that is both fair and privacy-preserving.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

This chapter introduces the exploration-exploitation dilemma, a crucial aspect of decision-making that influences choices ranging from everyday activities to high-stake scenarios. Through the simple yet relatable example of choosing a restaurant, we illustrate the balance between exploring new options and exploiting known ones. This narrative sets the stage for the thesis by highlighting the pervasive impact of this trade-off.

1.1 To explore or To exploit

"To explore" or "to exploit"—this dilemma is ever-present and manifests in various aspects of our daily lives. While this trade-off may not be immediately apparent to those unfamiliar with the specific topics we intend to discuss in this work, it underlies many of our everyday decisions.

Consider our protagonist, who, after a long day at work, must decide where to have dinner. This scenario, a familiar one, poses a recurring question: where should he eat? The restaurant down the street is a reliable choice, offering consistently good meals. On the other hand, there is a newly opened Chinese restaurant that presents a more uncertain option. His first visit there was disappointing, but the second experience was outstanding, serving some of the best food he has had in a while. Now, he faces a choice: should he give the Chinese restaurant another chance, hoping the initial bad experience was an anomaly, or should he play it safe and stick with the known quality of the familiar restaurant, even if it may not be the best?

This simple decision encapsulates the essence of the exploration versus exploitation trade-off. It illustrates the broader theme of this thesis: the delicate balance between seeking new opportunities and relying on established ones. As we delve deeper into this topic, we will explore how this fundamental decision-making process impacts various fields and scenarios, revealing its pervasive influence on our lives. Crucially, we aim to uncover strategies for making these decisions more effectively. How do we determine the optimal number of chances to give a new restaurant—or any new opportunity—before

[redacted]

Figure 1.1: Choice of Restaurants

Image credit: www.wsj.com

deciding it's not worth the risk? By addressing these questions, we hope to provide insights that enhance our ability to navigate the complexities of everyday choices.

1.2 Work Together, Learn Faster

In the realm of decision-making, the exploration versus exploitation dilemma is not confined to individuals alone. When multiple agents are involved, their collective experience can significantly influence and enhance decision-making processes. This theme—"Work Together, Learn Faster"—highlights how collaboration can accelerate learning and optimize outcomes, especially in scenarios where exploration and exploitation are critical.

Imagine a community of food enthusiasts, each facing the same decision as our protagonist: whether to stick with a reliable restaurant or try a new one. Individually, each person might spend a considerable amount of time and resources experimenting with different options. However, if these individuals share their experiences and insights, the entire group benefits. Positive experiences at a new restaurant can quickly spread through word of mouth, encouraging others to explore that option with reduced risk.

Conversely, a collective understanding of a consistently poor experience can prevent unnecessary trials, thereby saving time.

This principle extends beyond dining choices to various fields, including business, technology, and scientific research. In the corporate world, companies often face the challenge of innovating (exploration) while maintaining efficiency (exploitation). When organizations collaborate, they can share knowledge about successful innovations and best practices, thereby reducing the uncertainty and costs associated with exploring new ventures [1, 2]. For instance, partnerships between tech companies often lead to faster advancements as they pool resources and insights, navigating the exploration-exploitation trade-off more effectively than they could alone.

In scientific research, collaboration is equally vital. Researchers working in isolation might take years to achieve breakthroughs, as each must independently explore numerous hypotheses and methodologies. However, collaborative networks of scientists can share findings, methodologies, and data, dramatically accelerating the pace of discovery. This collective approach not only fosters innovation but also minimizes redundant efforts, allowing researchers to focus on the most promising avenues of inquiry.

Education and learning environments also benefit from this collaborative dynamic. Students and educators who share resources and insights can collectively explore a wider array of learning techniques and materials. For example, in a classroom setting, group projects and discussions enable students to learn from each other's experiences, balancing the exploration of new ideas with the exploitation of proven concepts. This collaborative learning approach helps students grasp complex concepts more quickly and effectively than if they were studying alone.

Moreover, the digital age has amplified the potential for collaborative exploration and exploitation. Online platforms, social media, and collaborative tools have made it easier than ever for individuals and organizations to share experiences and insights. Crowdsourced reviews, forums, and professional networks allow people to tap into a vast collective knowledge base, reducing the risks associated with exploration and enhancing the efficiency of exploitation. By leveraging these shared resources, individuals and organizations can make more informed decisions, accelerating their learning processes. As we continue to explore this dynamic throughout this work, it becomes clear that the path to optimal decision-making is often best traveled together.

1.3 Can We Be Optimal

While selecting a restaurant might seem like a decision that doesn't require mathematical rigor, higher-stake decisions benefit immensely from finding optimal solutions to the exploration-exploitation dilemma. This necessitates a deeper dive into the literature on choice optimality under uncertainty.

The study of choice optimality under uncertainty is a rich area of research, encompassing various models and theories that aim to balance exploration and exploitation effectively. While we will explore algorithmic formulations in later chapters, it's crucial to recognize that these methods offer structured approaches to making optimal decisions amidst uncertainty. These approaches leverage mathematical models to evaluate potential outcomes and make informed choices that maximize long-term benefits

An important question arises: can the process of achieving optimality for a single learner be directly applied to collaborative learners? Intuitively, one might think that simply pooling information from multiple agents would naturally lead to better decisions. However, this is not always the case. Collaborative decision-making introduces complexities that require specific strategies to ensure optimal outcomes. Factors such as communication overhead, conflicting preferences, and the distribution of knowledge must be carefully managed to avoid suboptimal decisions.

Moreover, while collaborative learning can theoretically enhance optimality, practical considerations might limit its feasibility. For instance, privacy concerns can be a significant barrier. Returning to our protagonist, while sharing dining experiences with a community could improve collective decision-making, he might be reluctant to broadcast his eating habits publicly. Privacy concerns extend beyond personal preferences; in corporate and scientific environments, sensitive information sharing can be problematic due to competitive or ethical reasons. Therefore, even if we achieve theoretically optimal learning under collaboration, we must evaluate its practicality. The theme of this work is centered around these critical questions:

Can we design systems that optimize selection under the exploration-exploitation trade-off? Does collaboration enhance the selection process? Can naive collaborations suffice, or are more complex strategies necessary to achieve better utility? What kind of sensitive information and privacy might one forfeit by collaborating? Are there mechanisms to balance exploration and exploitation in a way that fosters collaborative learning without compromising sensitive information? If not, what level of privacy is achievable, and at what cost?

1.4 Problems Explored

While a plethora of existing works and open-ended questions revolve around the themes discussed above, this thesis focuses on optimization using bandit algorithms [3, 4], which are further discussed in Chapter 2. Specifically, optimization under the constraints of federated learning and varying privacy concerns is given importance, addressing how these methods can be adapted to different constrained environments.

To deepen our intuition, consider a practical example: a network of hospitals aiming to improve patient treatment protocols. Each hospital has its own patient data, treatment methods, and outcomes. The challenge is to optimize treatment protocols across all hospitals by balancing exploration (trying new treatments) and exploitation (using known effective treatments). The hospitals cannot share raw patient data due to privacy regulations, making this a constrained environment that requires sophisticated optimization techniques.

1.4.1 Example: Optimizing Treatment Protocols in a Federated Network of Hospitals

Initial Setup

Imagine a network of hospitals that aim to collaboratively enhance their treatment protocols for a particular medical condition. Each hospital collects data on patient outcomes based on the treatments administered. However, due to strict privacy regulations and varying levels of data sensitivity, these hospitals cannot share raw patient data directly. Instead, they need to collaborate in a manner that allows them to learn from each other's experiences without compromising patient privacy.

[redacted]

Figure 1.2: Collaborative learning in clinical trials

Image credit: www.clinicallab.com

Applying Bandit Algorithms

To tackle this problem, we employ bandit algorithms within a federated learning framework. Bandit algorithms are particularly well-suited for this type of optimization problem because they are designed to balance exploration (testing new treatments) and exploitation (using treatments known to be effective). Each hospital acts as an agent, using a local bandit algorithm to make treatment decisions based on its own data while periodically updating a global model that aggregates insights from all hospitals.

Federated Learning with Privacy Constraints

In federated learning [5, 6], each hospital trains a local model on its own data and only shares model updates with a central server. The central server then aggregates these updates to improve a global model, which is shared back with all hospitals. This approach ensures that raw data never leaves the local hospital, thereby maintaining patient privacy. However, to further enhance privacy, techniques such as differential privacy, discussed in Chapter 3 can be applied to the shared updates, adding noise to the data to prevent the re-identification of individual patients.

Outcome

By using this federated learning approach with bandit algorithms, the network of hospitals can iteratively improve their treatment protocols in a privacy-preserving manner. Each hospital benefits from the collective knowledge without exposing sensitive patient data. The effectiveness of this collaborative learning process can be evaluated through simulation and theoretical bounds.

1.4.2 Two Settings

Our work explores the important underlying contributions in the stated fields. Moreover, two novel settings are studied and analysed in depth.

1. The first setting specifically examines federated optimization with privacy considerations when decisions involve selecting a subset of choices rather than a single option. Referring back to our example, suppose that at time of administration, a combination of multiple treatments could be chosen. These scenarios, where decision-making involves choosing a set of options, are explored in detail in Chapter 5.

2. The other setting delves into the conscious consideration of fairness during optimization. Rather than focusing solely on straightforward reward maximization, this approach examines alternative objectives within federated settings, as discussed in Chapter 6. Recalling our medical research example, it is crucial not to prematurely discontinue trialing treatments that may not be the best currently but still show potential. Purely optimizing for rewards can lead to algorithmic starvation, where promising treatments are overlooked. By incorporating fairness into the optimization process, we ensure a balanced consideration of all potential treatments, fostering a more equitable and effective decision-making framework.

1.5 Organisation

This thesis work has been divided into 7 chapters.

Chapter 1: Introduction

The Introduction chapter sets the stage by providing an intuitive understanding of the domain and the novel problems being explored. It outlines the exploration-exploitation dilemma, emphasizes the significance of optimization in high-stake environments, and introduces the overarching themes of privacy and fairness in federated learning contexts. This chapter serves as a roadmap for the subsequent discussions, highlighting the key questions and objectives of the thesis.

Chapter 2: Multi-Armed Bandits: Overview

Chapter 2 offers a comprehensive description of the Multi-Armed Bandit (MAB) problem. It formally defines the problem and explores its various formulations and complexities. The chapter also reviews popular literature and algorithms in the field, providing a solid foundation for understanding how bandit problems can be applied to real-world scenarios and the challenges they present in optimizing decisions under uncertainty.

Chapter 3: Multi-Armed Bandits: Differential Privacy: Overview

In Chapter 3, the concept of Differential Privacy is introduced and formally defined. The chapter explains the importance of privacy-preserving techniques in data analysis and decision-making processes. It covers fundamental principles, key definitions, and practical implementations of differential privacy, setting the stage for its application in later chapters, particularly in federated learning and multi-agent systems.

Chapter 4: Fairness and Privacy in Multi-Armed Bandits: Literature Review

Chapter 4 presents a literature review focused on the intersection of fairness and privacy within the context of Multi-Armed Bandits. It discusses major and relevant papers, highlighting how these aspects have been addressed in existing research. This chapter provides a critical analysis of the current state of the field, identifying gaps and challenges that the thesis aims to address through its novel contributions.

Chapter 5: Fairness and Privacy in Multi-Armed Bandits: Literature Review

This chapter delves into the first setting described earlier, based on a published paper. It explores the implementation of differentially private federated learning in scenarios where decisions involve selecting a subset of options. The chapter discusses the algorithmic strategies used to balance privacy and optimization and presents experimental results that demonstrate the effectiveness of the proposed approach in constrained environments.

Chapter 6: Exposure Of Fairness in Multi-agent Contextual Bandits with Privacy Guarantees

Chapter 6 focuses on the second setting, based on another published paper. It examines the integration of fairness into multi-agent contextual bandit problems while ensuring privacy guarantees. The chapter details the methods used to achieve exposure of fairness, discusses the trade-offs involved, and provides theoretical and empirical evidences to back the claims.

Chapter 7: Conclusion and Future Work

The final chapter summarizes the key findings and contributions of the thesis. It reflects on the implications of the research for the field of optimization in federated learning environments, particularly concerning privacy and fairness. The chapter also outlines potential directions for future work, suggesting areas where further research could build on the foundations laid by this thesis to address remaining challenges and explore new applications.

Chapter 2

Multi-Armed Bandits: Overview

This chapter provides a comprehensive introduction to Multi-Armed Bandits, offering formal definitions and exploring various variations of the problem. It examines multiple algorithmic solutions, detailing their approaches and effectiveness in addressing the challenges posed in Multi-Armed Bandit settings.

2.1 Motivation

Consider the following examples [7, 8, 9, 10, 11],

- *Online Advertising:* In online advertising, the platform needs to decide which ads to display to users based on their past behavior. Selecting the best ad to display to a user at a given time helps in maximizing click-through rates and revenue.

- *Recommendation systems:* Recommendation systems aim to suggest items (such as movies, products, or news articles) to users that they are likely to find interesting and engage with. Recommendations can be optimized by observing which items the users choose and recording their feedback (such as rating, click-through rate, or dwell time), then recommending the item with the highest expected reward to the next user. However, to ensure the exploration of new items, a small fraction of trials need to be allocated for selecting new items.

- *Procurement:* In procurement, companies often face a large number of suppliers and bids to evaluate and choose from, and the goal is to select the best suppliers and negotiate the best prices and terms. Over time, the goal is to learn from past results and adjust the selection criteria and strategies to optimize the procurement process.

- *Network Optimization:* Network optimization involves the allocation of resources such as bandwidth, power, or computational capacity. By selecting the best configuration based on feedback from the network, the performance can be improved and costs can be reduced.

- *Pricing Optimization:* In e-commerce, dynamic pricing can be used to optimize product revenue. By testing different price points for a product, the algorithm can quickly determine which price is most likely to maximize revenue.

In all of these examples, there is a need to select the best option from a set of choices, where the outcome of each choice is uncertain or probabilistic. MAB algorithms solve this problem by exploring the options to learn more about their outcomes and selecting the option that has the highest expected reward based on the knowledge gained from exploration. Moreover, in all of these examples, the decision-making process is iterative, meaning that the algorithm continuously learns from feedback and adjusts its strategy accordingly.

2.2 Bandit Framework

The MAB framework encapsulates problem settings with exploration-exploitation trade-offs. The framework considers a learning agent who makes choices (noted as arms) from a given decision set, in an interactive manner over a time period. We start by introducing some notation in Section 2.2.1 and then elaborate on the bandit setting in the following sections.

All of the above examples demonstrate the application of Multi-armed bandits (MAB) algorithms, which are a type of reinforcement learning technique that balances the trade-off between exploration and exploitation.

Figure 2.1: Octopus using MAB for slot machines

Image credit: www.towardsdatascience.com

2.2.1 Notations

In the MAB framework, there is a set of arms, denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$, where each arm represents a choice or an action (we use arms and actions interchangeably for the remainder of this work). When an arm is pulled, a reward is observed, often from an unknown distribution. The expected value or mean reward for an arm i is given by μ_i and the set of expected rewards is denoted by $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$. The process of arm selection is carried over T rounds by the learning agents.

2.2.2 Feedback

We denote the reward observed (a.k.a. feedback) on pulling an arm i, at round t, by $r_{i,t}$. The feedback $r_{i,t}$ depends on the reward structure for the setting. Often reward structures lying in $[0, 1]$ are considered for simplicity.

Based on the amount of information revealed in each round, we categorize the bandit feedback in the following manner:

- *Bandit feedback*: Only feedback about the chosen arm(s) is available.
- *Partial feedback*: Some information, in addition to the reward of the chosen arm(s), is available.
- *Full feedback*: The rewards for all arms that could have been chosen is available.

In addition to the quantitative measure of feedback, it is also important to consider the different reward structures. The commonly considered reward structures in bandit literature are Stochastic rewards, adversarial rewards, and Markovian rewards.

In a *stochastic reward setting*, each reward for an arm is sampled independently from a fixed but unknown distribution. In a *adversarial reward setting* on the other hand, the reward is determined by an adversary after observing the agent play. Often, the objective of the adversary is to provide as low of a reward as possible and randomized algorithms are often used in such settings. Lastly, the *markovian reward setting* assumes the association of each arm with a Markov process, each with its own state space and rewards. The reward changes for an arm based on the state space.

For the problem setting considered in this work, bandit feedback in stochastic reward settings are the primary focus. Unless specified otherwise, feedback in the rest of this work can be assumed to be bandit and stochastic in nature.

2.2.3 General Stochastic Bandits

The general stochastic bandit refers to the basic variant of bandits, whereby a single agent aims to identify the arm with the highest expected reward, given uncertainty and stochasticity in the reward distributions of the arms. At each round, the agent selects a single arm to pull, denoted by i_t, based on its current estimate of the expected reward of the arms. After pulling the arm, the agent receives a reward drawn from the distribution of the selected arm, which is stochastic.

The objective of the agent is to minimize the regret, defined in Section 2.2.4. The regret measures the opportunity cost of not selecting the best arm at each round and represents the agent's performance.

There are several algorithms that can be used to solve the general stochastic bandit problem, such as the epsilon-greedy algorithm, the Upper Confidence Bound (UCB) algorithm, and the Thompson Sampling algorithm. Section 2.3 discusses some of these.

2.2.4 Analysis: Regret

Let us consider the online advertisement example where some ads are to be shown to the user. Consider the potential revenue lost by not selecting the best ad at each round, which is a tangible and relevant metric for the website.

Now, let's say an algorithm, $\mathcal{A}_1$, picks the ads in a cyclic manner for the first $T/5$ rounds and then picks the ad with the highest observed reward for the rest of the rounds whereas another algorithm, $\mathcal{A}_2$, picks the ads to display randomly. When optimizing the revenue, it is important to figure out which algorithm amongst $\mathcal{A}_1$ and $\mathcal{A}_2$ performs better.

Definition 2.1: Regret

If $r_{i_t,t}$ denotes the reward observed by pulling arms according to an algorithm $\mathcal{A}$ and $r_{i_\star,t}$ denotes the reward observed by pulling arms according to an optimal algorithm $\mathcal{A}^\star$, then,

$$Reg(\mathcal{A}) = \sum_{t=1}^{T} r_{i_t,t} - r_{i_\star,t} \tag{2.1}$$

Regret of an algorithm $\mathcal{A}$ measures its performance against an optimal algorithm (it is implied that the optimal algorithm has access to the unknown parameters). Note that regret is a random variable that depends on the stochastic nature of the reward distributions of the arms and the random selection of the agent. Therefore, the regret is usually defined in expectation, i.e., as the expected value of the regret over multiple trials or runs of the algorithm. The goal of the agent is to minimize the expected regret, which represents the average opportunity cost of not selecting the best arm over multiple trials.

Definition 2.2: Pseudo Regret

The expected regret of an algorithm $\mathcal{A}$ is,

$$\mathbb{E}[Reg(\mathcal{A})] = \sum_{t=1}^{T} \mu_{i_t} - \mu_{i_\star} \tag{2.2}$$

2.2.5 Variants

Section 2.2.3 outlines the general stochastic setting. But often such a simple model is insufficient for practical applications. The literature on bandit formulations has introduced a wide range of variants/generalizations of the general bandit [12, 13]. Since it is not possible to cover all the variants as part of this work, we introduce some relevant variants here and expand upon them in later sections of this chapter.

The most common bandit variants are based on the reward setting. These have already been discussed in Section 2.2.2. In addition, we look at the contextual, combinatorial bandits, and sleeping bandits variants as those are relevant for the rest of this work.

- *Combinatorial Bandits:* Combinatorial bandits are a variant of multi-armed bandits where the arms are not simple actions, but rather combinations of actions or arms that can be chosen together as a group. In other words, the agent needs to choose a set of arms or a “combination" of arms to take action rather than choosing a single arm. We define the setting more formally and outline some popular algorithms in Section 2.4.
- *Contextual Bandits:* In contextual bandits, the agent observes some context or state before choosing an action. The reward distribution of each arm depends on the context, and the agent has to learn a mapping between contexts and arms to maximize its cumulative reward. Contextual Bandits are defined in more detail in Section 2.5.
- *Sleeping Bandits:* Unlike classical bandits, only a subset of all arms are available in any round. The rest of the arms are considered to be unavailable, or “asleep". At each instant t, the algorithm receives the set of awake arms as input and needs to select an arm to maximize cumulative reward over all rounds.

2.3 Popular Bandit Algorithms

In this section, we look at two popular algorithms for the general stochastic bandit framework defined in Section 2.2.3, namely UCB1 and Thompson sampling.

2.3.1 UCB1

UCB1 is a widely used algorithm for solving the general stochastic bandit problem. The main idea behind the algorithm is to form a confidence region around empirical reward estimates and pick the arm with the highest upper confidence bound. It balances the exploration-exploitation trade-off well. If an arm is pulled for very few rounds, the confidence interval is loose and the upper confidence bound will be high. Pulling an arm more tightens the confidence bound. So in the event of over-exploration, the arm with a higher reward estimate is picked more often, i.e., we see the exploitation term take over.

Algorithm 2.1: UCB1

```
Input $\alpha > 0$
for $t \in \{1 \dots K\}$ do
    Select an arm, $i_t \leftarrow t$
    Observe the reward $r_{i_t,t}$
    Set initial $\hat{\mu}_i \leftarrow r_{i_t,t}$
end for
for $t \in \{K + 1 \dots T\}$ do
    Select arm $i_t \in \arg\max_{i \in [K]} \left( \hat{\mu}_{i,t} + \sqrt{\frac{\alpha \ln T}{N_i(t)}} \right)$
    Observe reward $r_{i_t,t}$
    Update $\hat{\mu}_{i_t}$
end for
```

In Algorithm 2.1, $\mathcal{N}_i(t)$ denotes the number of times the arm i has been pulled till round t. Additionally, the term $\left(\hat{\mu}_{i,t} + \sqrt{\frac{\alpha \ln T}{N_i(t)}} \right)$ denotes the upper confidence term for arm i, where $\hat{\mu}_{i,t}$ leads exploitation whereas $\sqrt{\frac{\alpha \ln T}{N_i(t)}}$ is the exploration term.

UCB1 achieves a regret of $O(\log T)$ which has been proved optimal for the general stochastic bandit setting. Further, a lot of bandit literature derives from the ideas presented in UCB1, for both general stochastic as well as other variants of the problem.

Note: The regret term for UCB1 contains a Δ_i^{-1} term, which denotes the difference in the expected difference between the i^{th} arm and the best arm. In a gap-agnostic scenario, UCB1 achieves a regret of order $O(\sqrt{T} \log T)$.

2.3.2 Thompson Sampling

Thompson Sampling is a probabilistic algorithm for the general stochastic bandit problem [14]. Thompson Sampling uses Bayesian inference to choose which arm to pull. At each iteration, the algorithm assigns a prior distribution over the reward distribution of each arm. Then, it samples a value from each arm's prior distribution and selects the arm with the highest sampled value. After observing the reward for the selected arm, the algorithm updates the posterior distribution for that arm, incorporating the new information.

In the Algorithm 2.2, ζ represents the prior for the reward distributions. The algorithm updates the posterior distribution of the reward distribution for the selected arm based on the observed reward.

Algorithm 2.2: Thompson Sampling

```
Input: Prior for the reward distributions $\zeta$
for $t \in [T]$ do
    for $k = 1, 2, ..., K$ do
        Sample $\{\hat{\mu}_i\}_{i \in [K]} \sim \zeta$
    end for
    Select arm $i_t = \arg\max_{i \in K} \mathbb{E}_{\hat{\mu}_i}[r_{i,t}]$
    Observe reward $r_{i_t,t}$
    Update $\zeta$ using $(r_{i_t,t}, i_t)$
end for
```

Thompson sampling incurs a regret of $O(\sqrt{T \log T})$, which is similar to $UCB1$ in a gap-agnostic scenario.

2.4 Combinatorial Bandits

In a combinatorial bandit problem (CMAB), the decision-maker needs to select a subset of arms, rather than selecting a single arm.

Definition 2.3: Superarm

Let the set of all possible sets (power set) over the arms be denoted by $\mathcal{S}$. Then any subset of arms, $S \in \mathcal{S}$, is called a superarm.

In CMAB, the agents select a super arm in each round and every arm $k \in S$ is triggered and played as a result. The superarm is considered the unit of selection rather than individual arms since the reward is observed with respect to the superarm, rather than the individual arms in the superarm.

More formally, the CMAB problem consists of K arms associated with a vector of expected rewards, $\mu = (\mu_1, \mu_2, \ldots, \mu_K)$. In each round $t \in T$, the agent selects a superarm S, and all arms $i \in S$ are played. The reward, $r_{S,t}$ for the superarm is observed. Some works consider this as a cumulative of the rewards observed by the arms part of the super arm, whereby the reward variables of different arms may be dependent.

While there has been a plethora of work done in the last decade on CMAB problems, we focus on and outline two works that are significant for the rest of this work. The algorithm CUCB is presented in Section 2.4.1 and algorithm SS-UCB is presented in Section 2.4.2. Before presenting the two algorithms, we also introduce the notion of an oracle.

Definition 2.4: Oracle

An oracle takes the expectation vector, μ, and outputs a superarm $S \in \mathcal{S}$.

Definition 2.5: α-Approximation Oracle

An α-approximation oracle takes the expectation vector, μ, and outputs a superarm $S \in \mathcal{S}$ such that $r_S > \alpha \cdot r_{S^*}$. Here S^* is the optimal superarm, i.e., the superarm with the highest expected reward.

Often, even with a known expectation vector, the computation of optimal superarm is a computationally hard problem, and thus we defer to efficient approximation oracles.

2.4.1 CUMB

In the CUMB algorithm, presented in Chen et al. [15], the authors consider the CMAB setting with an oracle present. The following restrictions on the rewards of the superarm are considered,

- **Monotonicity.** Expected reward of playing any super arm $S \in \mathcal{S}$ is monotonically non-decreasing with respect to the expectation vector, i.e., if $\forall i \in [K],\ \mu_i \leq \mu'_i$, then, $\mathbb{E}[r_S(\mu)] \leq \mathbb{E}[r_S(\mu')],\ \forall S \in \mathcal{S}$.
- **Bounded smoothness.** There exists a continuous, strictly increasing (and thus invertible) function $f(.)$ with $f(0) = 0$, called bounded smoothness function, such that for any two expectation vectors μ and μ' and for any $\Lambda > 0$, we have $|r_S(\mu) - r_S(\mu'| \leq f(\Lambda)$ if $\max_{i \in S} |\mu_i - \mu'_i| \leq \Lambda$.

The CUMB algorithm is similar to UCB1, whereby the confidence bounds for each arm are calculated and then passed on to the oracle for superarm selection.

Algorithm 2.3: CUMB

```
for $t \in [T]$ do
    $[\forall i \in [K]]$, calculate the upper confidence bounds, $\bar{\mu}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{3\ln(T)}{2N_i(t)}}$
    Select superarm $S = ORACLE(\bar{\mu})$
    Update $\hat{\mu}$
end for
```

CUMB algorithm achieves a distribution-dependent regret of $O(\log(T))$.

2.4.2 SS-UCB

The authors of Deva et al. [16] consider a further variant of combinatorial bandits where each arm also has a known cost associated with it. The concept of quality, defined as the difference between scaled (using a pre-defined constant) reward observed on choosing an arm and the cost associated, is used to constraint the $CMAB$ setting. The agent's objective is to select sets of arms (superarms) that minimize regret while guaranteeing a minimum level of quality.

Algorithm 2.4: SS-UCB

```
Input: Minimum quality threshold $\alpha$, costs $(c_i)_{i\in[K]}$, constant $QR$, error margin $\epsilon_2$
$\tau = \frac{3\ln T}{2\epsilon_2}$
for $t \in [1, \tau]$ do
    Select superarm $S = K$
    Update $\hat{\mu}$
end for
for $t \in [\tau, T]$ do
    $[\forall i \in [K]]$, calculate the upper confidence bounds, $\bar{\mu}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{3\ln(T)}{2N_i(t)}}$
    Select superarm $S = ORACLE(\bar{\mu}, QR, \alpha + \epsilon_2)$
    Update $\hat{\mu}$
end for
```

Deva et al. [16] proposes two oracles, namely $DPSS$ and GSS. $DPSS$ formulates the superarm selection as an integer linear programming problem while GSS provides an approximation using greedy approach. $SS - UCB$ achieves a regret of $O(\log(T))$ with high probability under certain distribution conditions.

2.5 Contextual Bandits

In the Contextual Bandit setting, the agent interacting with the arms generally observes additional information (context) for each of the arms at the start of every round. The rewards associated with each arm depend on this context. The player's objective would involve learning the mapping from a set of contexts to rewards so the optimal arm given a particular context can be chosen.

More formally, at the start of each round, t, the agents observe the context vectors, $x_t(i) \in \mathbb{R}^d$, for each arm $i \in [K]$. We denote the set of context vectors at time t by $\mathcal{X}_t = \{x_t(1), x_t(2), \ldots, x_t(K)\}$. There exists a deterministic but unknown mapping $\mathcal{F}(.) : \mathbb{R}^d \rightarrow \mathbb{R}$, from the context vector to the feedback, i.e., for any arm k, $r_{k,t} = \mathcal{F}(x_t(k))$. We note it as the payoff function. We observe that a stochastic setting is assumed in most contextual bandits. Thus the reward is given by $r_{k,t} = \mathcal{F}(x_t(k)) + \eta_t(k)$, whereby $\eta_t(k)$ is a randomly sampled noise term.

Additionally, we want to note that we have defined a shared payoff function for all arms, but this is not always the case. Contextual bandits can be of the following formulations,

- *Shared Model* Different arms share the payoff function. In this case, the payoff function is defined irrespective of the arm and only takes the context vector as the input.
- *Disjoint Model* Different arms **do not** share the payoff function. In this case, each arm has its payoff function and we denote the set of payoff functions by $\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_K\}$.

Often a linear payoff function is assumed, i.e., $\mathcal{F}(x_t(k)) = \theta \cdot x_t(k)$, where θ is the coefficient vector. In the case of a disjoint linear model, θ represents the set of coefficient vectors, $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$.

2.5.1 LinUCB

The authors of Li et al. [17] provided some of the premiere work for the contextual bandit problems. Algorithms for a disjoint linear model as well as a hybrid linear model (shared + disjoint parameters) are given. They adopted a ridge regression-based mean estimation with confidence bounds techniques to provide an optimistic arm selection algorithm. We note the algorithm for the disjoint model below. We refrain from explicitly stating the algorithm for the hybrid model but state the reward formulation.

For ridge regression, the history of the selected context and feedback received is used as the training data. Let $\mathcal{D}_i$ be a design matrix of dimension $\mathcal{N}_i(t) \times d$ at trial t, whose rows correspond to the $\mathcal{N}_i(t)$ training inputs (the context vectors whenever arm i is selected). Let y_i be a vector of length $\mathcal{N}_i(t)$ at time t noting the corresponding feedback observed. Then the ridge regression solution for calculating estimated θ_i is given as follows,

$$\hat{\theta}_i = \left(\mathcal{D}_i^\intercal \cdot \mathcal{D}_i + I_d\right)^{-1} \mathcal{D}_i^\intercal y_i$$

where I_d is the $d \times d$ identity matrix. For conciseness, we use $V_i = \left(\mathcal{D}_i^\intercal \cdot \mathcal{D}_i + I_d\right)$ and $b_i = \mathcal{D}_i^\intercal y_i$. Note that the t term is omitted as these terms are a single variable being continuously updated. Later in this work, we may use a subscript notation including t (e.g. $V_{i,t}$) to note the value of the variable at the time t.

Algorithm 2.5: LinUCB - Disjoint linear models

```
Input: $\alpha > 0$
for all $i \in [K]$ do
    $V_i = I_d$
    $b_i = 0_{d\times 1}$
end for
for $t \in [T]$ do
    Observe context for all arms $i \in [K]$, $\mathcal{X}_t$
    for $i \in [K]$ do
        $\hat{\theta}_i = V_i^{-1} b_i$
    end for
    Select arm $i_t = \arg\max_{i\in[K]} \hat{\theta}_i + \alpha\sqrt{x_t(i)^\intercal V_i^{-1} x_t(i)}$
    Observe reward $r_{i_t,t}$
    $V_{i_t} = V_{i_t} + x_t(i_t) \cdot x_t(i_t)^\intercal$
    $b_{i_t} = b_{i_t} + r_{i_t,t} x_t(i_t)$
end for
```

While no formal regret guarantees were provided for this work, subsequent based works provide regret upper bounds.

For the hybrid linear model, the reward is determined by a combination of shared and local parameters,

$$E[\hat{r}_{i_t,t}|x_t(i_t)] = z_t(i_t)^\intercal \beta + x_t(i_t)^\intercal \theta_{i_t}$$

where $z_t(i_t)$ is a subset of $x_t(i_t)$ denoting the shared parameters and β is the corresponding shared coefficient vector.

2.5.2 SupLinUCB

Building on Algorithm 2.5, the authors of Chu et al. [18] introduces a modified version of LinUCB and prove regret upper bounds for it. The same setting as LinUCB, while considering a shared model, with the added assumption of statistical independence of samples.

Algorithm 2.6: BaseLinUCB (at time step t)

```
Inputs: $\alpha > 0$, $\Psi_t \subseteq \{1, 2, \ldots, t-1\}$
$V = I_d + \sum_{\tau\in\Psi_{i,t}} x_\tau(i_\tau)^\top x_\tau(i_\tau)$
$b = \sum_{\tau\in\Psi_t} r_{i_\tau,\tau} x_\tau(i_\tau)$
$\theta = V^{-1}b$
for $i \in [K]$ do
    $w^s_{i,t} = \alpha\sqrt{x_t(i)^\top V_i^{-1} x_t(i)}$
    $\hat{\mu}^s_{i,t} = \theta^\top x_t(i)$
end for
```

Algorithm 2.7: SupLinUCB

```
Initialization: $S = \ln T$, $\Psi^s_t = \phi$ for all $\forall s \in [T]$
for t = 1,2,. . . , T do
    $s = 1$ and $\hat{A}_1 = [K]$
    $j = 1 + (t \bmod K)$
    repeat
        Use BaseLinUCB with $\Psi^s_t$ and context vector $X_t$ to calculate the width $w^s_{i,t}$ and upper
        confidence bound $\bar{\mu}^s_{i,t} = (\hat{\mu}^s_{i,t} + w^s_{i,t}), \forall i \in \hat{A}_s$
        if $w^s_{i,t} \leq \frac{1}{\sqrt{T}}, \forall i \in \hat{A}_s$ then
            Choose $i_t = argmax_{i\in\hat{A}_s} \bar{\mu}^s_{i,t}$
            Keep the same index sets at all levels:
            $\Psi^{s'}_{t+1} = \Psi^{s'}_t, \forall s' \in [S]$
        else if $w^s_{i,t} \leq 2^{-s}, \forall i \in \hat{A}_s$ then
            $\hat{A}_{s+1} = \{i \in \hat{A}_s \mid \bar{\mu}^s_{i,t} \geq \max_{i'\in\hat{A}_s}(\bar{\mu}^s_{i',t}) - 2^{1-s}\}$
            $s = s + 1$
        else
            Select $i_t \in \hat{A}_s$ such that $w^s_{i_t,t} > 2^{-s}$.
            Update the index sets at all levels:
            $\Psi^{s'}_{t+1} = \begin{cases} \Psi^{s'}_t \cup \{t\} & \text{if } s = s' \\ \Psi^{s'}_t & \text{otherwise} \end{cases}$
        end if
    until an arm $i_t$ is found
end for
```

Algorithm 2.7 adapts a ridge regression-based solution as well. The main advantage of 2.7 over 2.5 is that the authors were able to show a regret bound of $\bar{O}(\sqrt{Td})$ while being simple and robust in practice.

2.6 Notational Summary

For ease of reference, Table 2.1 lists some important notations introduced in this chapter.

Symbol	**Description**
K	Number of arms
$\mathcal{K} = [K] = \{1, 2, \ldots, K\}$	Set of arms
T	Number of rounds
i_t	Arm chosen at round t
μ_i	Expected reward of arm i
$\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$	Set of expected rewards
$\hat{\mu}_{i,t}$	Empirical estimate of the expected reward of arm i at round t
$\bar{\mu}_{i,t}$	UCB for arm i at round t
$r_{i,t}$	Reward observed for arm i at round t
$Reg(\mathcal{A})$	Regret for algorithm $\mathcal{A}$
$\mathcal{N}_i(t)$	Number of times arm i has been pulled till round t
ζ	Prior for the reward distribution
S	Superarm
$\mathcal{S}$	Set of all possible superarms
$S^\star$	Optimal superarm
α	Minimum quality threshold

Table 2.1: Multi-armed bandit: Important Notations

Chapter 3

Differential Privacy: Overview

This chapter introduces differential privacy, emphasizing the importance and necessity of this concept in data security and privacy. It provides formal definitions of key terms to establish a foundational understanding. Various techniques for ensuring differential privacy are presented, alongside a detailed exploration of the properties of differential privacy mechanisms.

3.1 Motivation

With the increasing amount of data being generated by various sources, organizations can leverage this data to gain insights into their operations and make informed decisions. But, this advent of data-based analytic and prediction modeling comes with its own caveats. Consider the following examples,

- *Medical Research:* Researchers may want to study the relationship between certain genetic markers and the likelihood of developing a particular disease. Release of this information could reveal sensitive information about individuals, such as their risk of developing certain diseases or their family's medical history.

- *Ride-Sharing:* Ride-sharing companies often collect a large amount of data about their users, including their pickup and drop-off locations, trip duration, and payment information. This data, without proper protection, could be used to determine the individuals' movement and habit patterns.

- *Census Data:* Governments collect census data to help allocate resources and make policy decisions. This data contains information about individuals, such as their race, ethnicity, and income.

- *Smart Grids:* Smart grids use sensors and data analytics to optimize energy usage and distribution. However, the data about energy usage can also be used to know about individuals' lifestyles and habits.

- *Online Advertising:* Advertisers often collect data on users' online behavior to target them with personalized ads. Users' interests and behavior patterns are collected, which can have many adversarial uses other than ads.

- *Social Media Analysis:* Social media platforms collect a large amount of data on users' behavior and interests. This data can be used for various purposes, such as targeted advertising and trend analysis. Without proper moderation, sensitive information about individuals' personal lives and preferences can be used for radical manipulation.

- *Financial Analysis:* Financial institutions often collect data on customers' financial transactions and behavior to detect fraud and make credit decisions. However, sharing this data could reveal sensitive information about individuals' financial situations and behavior.

The common denominator amongst the above examples is the need to protect individuals' sensitive information and preserve their privacy while still allowing useful insights to be gained from the data. In each example, sharing the raw data could potentially reveal sensitive information about individuals' behaviors, habits, and preferences.

Differential privacy is a mathematical framework for protecting individuals' privacy in data analysis. It works by adding random noise to data before it is released or analyzed, in a way that guarantees that an individual's data cannot be distinguished from any other individual's data, even if an attacker has access to other publicly available information.

The core idea behind differential privacy is to provide a formal definition of privacy that is based on the concept of indistinguishability. The definition states that a query applied to a data set should produce essentially the same result irrespective of any particular data point being included in the data set. In other words, the result of a query should be “indistinguishable" from what it would be if an individual data point were excluded from the data set.

Differential privacy can be applied to a wide range of data analysis tasks, including database queries, machine learning, and statistical modeling [19]. It provides a principled and rigorous approach to privacy protection that a solid mathematical foundation backs. Differential privacy is increasingly being used in a variety of applications, including healthcare, social science, finance, and marketing, where it is essential to protect individuals' sensitive information while still allowing valuable insights to be gained from the data.

Note that while we introduce several notations in this chapter, they are independent of other chapters of this thesis and any notation from this chapter used in later chapters will be explicitly re-stated.

3.2 Mathematical Formalization

We start this section by first introducing some formal definitions (with respect to our context) of commonly used terms such as database distance and randomized mechanisms. We then quantify the notion of privacy discussed in Section 3.2.2. Finally, we define differential privacy in Section 3.2.3.

3.2.1 Preliminary Definitions

Differential privacy provides privacy as a process. Here, privacy guarantees plausible deniability of any outcome, and thus randomization is essential for differential privacy. Before we start defining privacy loss and the formal definition of differential privacy, we introduce the notion of database and randomized algorithms, which are essential prerequisites.

Definition 3.1: Database

We consider databases, x, as being collections of records from a universe $\mathcal{X}$. We will often represent the database by their histograms, whereby $x \in \mathbb{N}^{|\mathcal{X}|}$ and x_i represents the number of elements of type $i \in \mathcal{X}$ in the database x.

Definition 3.2: Randomised Algorithm

For a randomized algorithm M that takes inputs from domain A and produces outputs in the discrete range B, there exists a mapping $M : A \longrightarrow \Delta(B)$. Given an input $a \in A$, the probability of the algorithm M outputting $M(a) = b$ is $(M(a))_b$ for all $b \in B$.

A non-trivial privacy guarantee that holds true regardless of all current or even potential sources of auxiliary information, including other databases, requires randomization. Suppose, for the sake of contradiction, that we have a non-trivial deterministic algorithm. According to non-triviality, there is a query and two databases that each produce unique results for this query. By altering each row individually, we can show that there are two databases, which differ only in the value of a single row, and for which the identical query produces different results. The value of the data in the unidentified row can be discovered by an enemy who is aware that the database is one of these two almost identical databases.

In the above-given representation, the l_1 distance between two databases x and y will be a natural measure of their separation.

Definition 3.3: Distance between Databases

The l_1 norm of a database x is represented by $\|x\|_1 = \sum_{i=1}^{|X|} |x_i|$ and the l_1 distance between two databases x and y is given by $\|x - y\|_1$.

Here, the size of a database x (i.e., the number of records it includes) is measured by $\|x\|_1$, and the number of records that differ between x and y is given by $\|x - y\|_1$.

3.2.2 Privacy Loss

We now quantify the privacy loss for any randomized algorithm M. This allows us to define an optimal privacy scenario.

Definition 3.4: Privacy Loss

Given any output, $O \in M(x)$, for a query on database x, the privacy loss with respect any other database y is given by,

$$\mathcal{L}_{x||y}(O) = \ln\left(\frac{Pr[M(x) = O]}{Pr[M(y) = O]}\right) \tag{3.1}$$

Here, if the privacy loss is positive, it indicates that the probability of observing output O is more likely when database x is queried. On the other hand, a negative loss represents that the probability of observing output O is more likely when database y is queried. So based on the query output, one can make an enhanced prediction on the database queried unless the privacy loss is zero.

The ideal notion of privacy would want zero privacy loss irrespective of output. But this is only possible if both the databases in question produce identical outputs for all possible queries or the randomized algorithm used completely ignores any information gained from the databases and instead just outputs pure noise. In the former scenario, the databases are functionally the same (with respect to queries). In the latter case, the queries lack any information about the database, which defeats the purpose of querying the database. We look at a relaxed version of the privacy notion, where privacy loss is not zero but is bounded.

3.2.3 Differential Privacy

With a grasp of the motivating examples and key concepts, we can now formalize the notion of differential privacy. It hinges on the idea that a randomized algorithm should produce similar outputs, even when operating on databases that differ minimally. This ensures that an individual's presence or absence in the database has a negligible impact on the algorithm's results, effectively protecting their privacy.

Definition 3.5: ϵ-Differential Privacy

A randomized algorithm, M, with input domain $\mathbb{N}^{|\mathcal{X}|}$, is ϵ-differentially private if for any two databases x and y differing in at most one element, and for any set of possible outputs O in the range of M the following inequality holds:

$$Pr[M(x) \in O] \leq e^{\epsilon} Pr[M(y) \in O] \tag{3.2}$$

Here, ϵ is a privacy parameter that controls the degree of privacy offered. Smaller values of ϵ correspond to stronger privacy guarantees.

Definition 3.6: (ϵ, δ)-Differential Privacy

A randomized algorithm, M, with input domain $\mathbb{N}^{|\mathcal{X}|}$, is (ϵ, δ)-differentially private if for any two databases x and y differing in at most one element, and for any set of possible outputs O in the range of M the following inequality holds:

$$Pr[M(x) \in O] \leq e^{\epsilon} Pr[M(y) \in O] + \delta \tag{3.3}$$

Here, δ is a small probability value that allows for a slight relaxation of the privacy guarantee. In practice, δ is often chosen to be very small (e.g., $\delta < \frac{1}{|\mathcal{X}|}$).

3.3 Properties of Differential Privacy

Differential privacy's robustness extends beyond individual algorithms, offering valuable properties that enhance its practical utility and flexibility. We delve into two key properties: composition, which empowers the construction of intricate differentially private systems, and post-processing, which grants freedom to analyze privacy-protected data without jeopardizing its privacy guarantees.These properties play pivotal roles in unlocking the full potential of differential privacy in real-world applications.

3.3.1 Composition Theorem

The composition theorem is a crucial property that enables building complex differentially private systems out of simpler components. It guarantees that if multiple independently run, differentially private algorithms are applied sequentially, the overall system also remains differentially private. This is vital for real-world applications where complex analyses often involve chaining multiple privacy-preserving mechanisms.

The specific composition theorem used depends on the chosen definition of differential privacy. For example, for ϵ-DP:

- *Sequential Composition:* If M_1 is ϵ_1-DP and M_2 is ϵ_2-DP, then their sequential composition is $(\epsilon_1 + \epsilon_2)$-DP.

- *Parallel Composition:* If M_1 is ϵ_1-DP and M_2 is ϵ_2-DP, and their outputs do not depend on each other, then their parallel composition is $\max(\epsilon_1, \epsilon_2)$-DP.

These guarantees allow chaining multiple differentially private mechanisms while still maintaining a bound on the overall privacy loss.

3.3.2 Post-processing

Another important property is the post-processing property. It states that any deterministic function applied to the output of a differentially private mechanism will remain differentially private. This allows developers to perform further analysis on the privacy-protected data without compromising the privacy guarantee. For example, you can calculate statistics, create visualizations, or apply machine learning algorithms to the output of a differentially private mechanism, as long as these operations are deterministic.

3.4 Crafting Privacy-Preserving Algorithms: Techniques for Differential Privacy

With a clear understanding of differential privacy's foundations, we now turn our attention to the techniques that bring its theoretical guarantees to life. This section unveils three commonly employed tools that enable designing of differentially private algorithms, expertly balancing utility with privacy:

3.4.1 Randomized Response

Surveys and questionnaires often probe sensitive topics, requiring careful privacy protection. Randomized response techniques elegantly address this challenge by introducing a veil of randomness, safeguarding individual responses while still permitting meaningful statistical analysis

Mechanism

- Individuals employ a randomizing device (e.g., a coin or a die) to introduce uncertainty into their answers.

- The device's instructions determine whether a truthful response or a strategically altered response is provided, obscuring individual truths.

Illustrative Example

Consider a survey exploring past criminal activity. The randomized response mechanism ensures privacy while enabling crime prevalence estimation:

- Each participant privately flips a coin.
- Heads prompts a truthful response; tails triggers a response guided by the randomizing device, intentionally obscuring the truth.

Formal Modeling

The technique's effectiveness stems from its probabilistic nature. The probability distribution of randomized responses, modeled using conditional probabilities, delicately balances privacy and utility. For the illustrated example, the randomized response can be modeled using conditional probabilities. Let R be the randomized response, where $R = 1$ represents a "yes" response and $R = 0$ represents a "no" response. We can define the conditional probabilities as follow,

$$Pr[R = x|x] = p \qquad \text{(the probability of responding truthfully)}$$

$$Pr[R \neq x|x] = 1 - p \qquad \text{(the probability of responding untruthfully)}$$

where p is the parameter that determine the level of noise introduced by the mechanism.

The randomized response mechanism ensures that the individuals' true answers are protected because the noise introduced by the randomizing device makes it difficult to determine the true response for a specific individual. However, statistical analysis can still be performed on the aggregated randomized responses to estimate the proportion of individuals with a “yes" answer while preserving privacy.

3.4.2 Laplace Mechanism

The Laplace mechanism injects carefully calibrated noise, drawn from the Laplace distribution, into the query output, effectively cloaking individual contributions.

Mechanism

Given a query function $f(x)$ that calculates some statistic over the dataset x, and a sensitivity of Δf, which captures the maximum change in $f(x)$ due to a single individual's data, the Laplace mechanism adds noise drawn from the Laplace distribution with scale parameter $b = \Delta f/\epsilon$.

Key Considerations

The magnitude of noise added is governed by two factors:

- *Sensitivity (Δf):* Reflects how much the query output could change due to a single individual's data. Higher sensitivity necessitates more noise to maintain privacy.

- *Privacy budget (ϵ)*: Dictates the strength of privacy protection. A lower ϵ requires more noise to ensure stronger privacy guarantees.

Formal Modeling

- Let $f(x)$ be the query function and o be the true output.
- The output of the Laplace mechanism is $O = f(x) + Lap(\Delta f/\epsilon)$, where Lap($\lambda$) represents a random variable drawn from the Laplace distribution with scale parameter λ.

Intuitively the Laplace distribution assigns high probabilities to small noise values and progressively lower probabilities to larger values. This helps mask the influence of individual data points while still allowing for meaningful statistical analysis.

3.4.3 Gaussian Mechanism

When releasing real-valued query results, the Gaussian mechanism emerges as a powerful alternative to laplacian mechanism. It strategically adds noise drawn from the Gaussian distribution, offering distinct advantages in certain scenarios.

Mechanism

Similar to the Laplace mechanism, the Gaussian mechanism adds noise drawn from the Gaussian distribution to the query output. However, the noise scale σ depends on both the sensitivity and the desired privacy level:

$$\sigma = (2\Delta f^2/\epsilon^2) * (\log(1.25/\delta))$$

Distinctive Advantages

- *Lower Error Rates:* Often yields lower error rates compared to the Laplace mechanism for queries with low or moderate sensitivity. This is because the Gaussian distribution concentrates most of its probability mass around the mean, leading to less distortion on average.
- *Suitable for specific tasks:* Particularly beneficial for tasks involving linear queries or those with Gaussian-distributed errors, as the added noise preserves these properties.

Formal Modeling

- Let $f(x)$ be the query function and o be the true output.
- The output of the Gaussian mechanism is $O = f(x) + N(0, \sigma)$, where $N(\mu, \sigma^2)$ represents a random variable drawn from the Gaussian distribution with mean μ and standard deviation σ.

Similar to the Laplace mechanism, the noise scale σ balances privacy and utility. A larger σ implies stronger privacy guarantees but potentially higher error rates in the analysis.

Choosing the most suitable mechanism among these three depends on various factors, including the data type, query sensitivity, desired privacy level, and specific application requirements. Understanding the strengths and limitations of each technique empowers developers to design differentially private systems that effectively protect sensitive information while enabling valuable insights.

Chapter 4

Fairness and Privacy in Multi-Armed Bandits: Literature Review

The chapter discusses various approaches to define and incorporate fairness in MAB algorithms, such as minimum share, and fairness of exposure. By reviewing existing fairness-aware MAB algorithms and their challenges, the chapter aims to contribute to the development of fair MAB systems. Additionally, it delves into differentially private MAB settings, exploring key papers that address the trade-off between privacy and utility, such as LinUCB with noise and FedUCB.

4.1 Notion Of Fairness in Bandits

Traditional bandits inherently converge to the best arm, adapting a ’winner takes all’ recommendation. In addition, widespread adoption of MABs have raised concerns about potential biases and unfairness in their decision-making processes. These concerns stem from several factors:

- *Data Bias:* Reward feedback used may inherently contain biases reflecting societal inequalities or historical discrimination. These biases can be inadvertently perpetuated by the algorithms, leading to unfair outcomes for certain groups or individuals.

- *Algorithmic Bias:* Traditional bandit algorithms optimise for convergence to the arm with highest expected reward. This leads to a ’winner takes all’ where other arms can suffer from starvation. Also, the design choices and assumptions embedded within MAB algorithms themselves can unintentionally introduce biases. For instance, regret definitions prioritizing short-term rewards might overlook options that are initially less appealing but have the potential for greater long-term benefits for specific user groups.

- *Lack of Traceability:* Often, bandit algorithms use randomness to drive exploration, making it challenging to understand their decision-making rationale and identify potential biases. This lack of traceability can hinder user trust and accountability.

Unmitigated fairness concerns can have significant societal and ethical implications. For example, biased MAB algorithms in recommender systems could disproportionately disadvantage certain groups in terms of job opportunities or access to educational resources. Similarly, biased ad placement mechanisms could perpetuate harmful stereotypes or unfairly target specific demographics with predatory products or practices.

4.1.1 Addressing Fairness Challenges in MABs

In response to these concerns, a growing body of research is focused on incorporating fairness considerations into the design and implementation of MAB algorithms. This chapter delves into this evolving field by exploring various approaches to promote fairness in MAB settings. This involves two major challenges,

- *Defining Fairness in MABs:* Establishing a clear definition of fairness in the context of MABs is crucial. This involves specifying the desired properties and identifying the potential groups or individuals at risk of unfair treatment. Various fairness notions exist,
  - *Minimum share:* This fairness constraint specifies that each arm should have certain fraction of total selection (specified beforehand to the algorithm).
  - *Group fairness:* Under this fairness concern, commonly associated with contextual bandits, the ideal output of the bandit algorithm is insensitive to a protected information of the arms. The aim is that algorithm shouldn't show bias towards/against any particular group (sharing same protected attribute) of arms.
  - *Group fairness in agents:* In multi-agent cooperative bandits, different learners may end up with different rewards. The fairness constraints put forth here stipulates that agents with similar traits shouldn't experience large differences in rewards, even if they belong to different groups.
  - *Fairness of exposure:* Fairness of exposure stipulates that the number of selections for any arm should be proportionate to its reward (relative to other arms).
- *Fairness-Aware Algorithm Design:* Algorithms need to leverage various techniques to achieve fairness goals. Some explored techniques include incorporating fairness penalties in the reward function, setting minimum exploration guarantees for specific groups, or utilizing counterfactual reasoning to adjust rewards and promote equalized odds.

This chapter will explore these aspects in further detail, examining specific fairness notions, reviewing existing fairness-aware MAB algorithms, and discussing the challenges and limitations of achieving fairness in MAB settings. By critically analyzing these approaches, we aim to contribute to the ongoing efforts towards developing fair, transparent, and ethical MAB algorithms for real-world applications.

4.1.2 Existing Algorithms

This section presents a discussion on some algorithms proposed for achieving arm fairness in MAB settings. The first algorithm looks at ensuring minimum share while the second algorithm optimises for ensuring minimum exposure for the arms.

4.1.2.1 FAIR-LEARN

Patil et al. [20] propose an interesting variant of the stochastic multi-armed bandit problem, which they call the FAIR-MAB problem. In addition to the objective of maximizing the sum of expected rewards, the agent also needs to ensure that at any time, each arm is pulled at least a pre-specified fraction of times. They define a fairness-aware regret, which they call $r - Regret$, that takes into account the above fairness constraints and extends the conventional notion of regret in a natural way.

Algorithm 4.1: FAIR-LEARN

```
**Input:** $(\delta_i)_{i \in K}$, error tolerance $\epsilon_2 \geq 0$, LEARN(.)
**for** $t \in [T]$ **do**
    $A(t) = i | \delta_i(t-1) - \mathcal{N}_i(t) > \epsilon_2$
    Select arm $i_t = \begin{cases} argmax_{i \in K}(\delta_i(t-1) - \mathcal{N}_i(t)) & \text{if } A(t) \neq \emptyset \\ LEARN(.) & \text{otherwise} \end{cases}$
    Update estimate
**end for**
```

Here, $(\delta_i)_{i \in K}$ represent the pre-specified fraction vector and LEARN(.) refers to any learning algorithm. FAIR-LEARN with UCB learning algorithm achieves a $r - Regret$ of $O(\log T)$.

4.1.2.2 FairX-LinUCB

Wang et al. [21] proposes a novel bandit objective that guarantees fairness of exposure to arms. This is captured by fairness regret defined as the difference between a optimally fair arm selection distribution and the picked distribution. The work also studies the optimisation of utility (for learner) and fairness simultaneously, which is captured by reward regret. The central idea is to construct a confidence region, CR_t, at every round t, containing the true parameters with high probability. The proposed algorithm then optimistically selects parameters from the confidence region so as to minimise the reward regret.

While similar algorithms are proposed for both stochastic and linear contextual bandit settings, only the contextual algorithm is highlighted here. The $fairness\ ratio$ here is given by $\frac{f_{i,t}}{\sum_{j \in K} f_{j,t}}$ if f denotes the calculated reward for arm i at round t.

The FairX bandit algorithm incurs a fairness regret of $O(d\sqrt{T})$ with high probability.

Algorithm 4.2: FairX-LinUCB

```
Input: $\beta_t > 0$
$V = I_d$
$b = 0_{d \times 1}$
for $t \in [T]$ do
    Observe context for all arms $i \in [K]$, $\mathcal{X}_t$
    $\hat{\theta} = V^{-1} b$
    Construct $CR_t = (\theta : \|\theta - \hat{\theta}\|_V \leq \sqrt{\beta_t})$
    Select $\theta_t \in \theta$ which has maximum $\{fairness\ ratio\} * \{reward\}$
    Construct selection policy, $\pi_t$, using $\theta_t$ which is equal to $fairness\ ratio$ for each arm

    Sample $i_t \sim \pi_t$
    Observe reward $r_{i_t,t}$
    $V = V + x_t(i_t) \cdot x_t(i_t)^\intercal$
    $b = b + r_{i_t,t} x_t(i_t)$
end for
```

4.2 Notion Of Differential Privacy in Bandits

The integration of differential privacy into the framework of multi-armed bandits (MAB) presents a unique set of challenges and opportunities. While the concepts of MAB and differential privacy have been introduced individually, their intersection opens up a new dimension in the field of privacy-preserving sequential decision making. This section delves into the analysis of differential privacy in the context of MAB and discusses various proposed algorithms that aim to balance the trade-off between privacy, and the exploration-exploitation dilemma inherent in MAB problems.

4.2.1 Privacy Preserving Bandits

Let's take the example of a recommendation system: A user provides a context, which includes elements like past preferences and browsing history. The system then suggests an item and receives a reward if the user interacts with it. If we ignore this context and model the problem as a standard Multi-Armed Bandit problem, where each possible item is an action, we overlook the diversity of users' preferences. On the other hand, learning each user's preferences separately doesn't allow for generalization across users.

Hence, it's common to model this task as a contextual linear bandit problem. However, it also highlights the need for privacy in the contextual bandit settings. Users' past interactions and browsing history are sensitive personal information, yet they are strong predictors of future interactions. Therefore, it's crucial to handle this data with care to respect users' privacy.

Similarly in multi-agent MAB setting, cooperative (often modeled as federated learning) learning is very beneficial but often requires privacy guarantees for the learning agent to be effective.

Incorporating differential privacy into MAB introduces a new layer of complexity to the exploration-exploitation trade-off. The need to preserve privacy can limit the ability of the algorithm by introducing extra noise, potentially impacting the overall utility. Therefore, a careful balance must be struck to ensure both robust utility and strong privacy guarantees.

The strength of the privacy guarantee, often quantified by the differential privacy parameter ϵ, can inversely affect the utility of the learning algorithm. Lower values of ϵ provide stronger privacy guarantees but can limit the algorithm's ability to learn from the data, thereby reducing utility.

4.2.2 Existing Algorithms

This section presents a discussion on some algorithms proposed for achieving differential privacy in MAB settings. Despite the relative novelty of this field, which has resulted in a limited body of work, this section will explore two notable contributions that have significantly impacted this area.

4.2.2.1 LinUCB with Noise

The authors of Shariff and Sheffet [22] study linear bandit problem for private learners. They adopt the notion of joint differential privacy, converting LinUCB to joint differentially private version.

The central idea is to maintain a noisy gram matrix and reward vector by adding perturbations to it. They propose use of a tree-based mechanism to ensure DP efficiently.

Algorithm 4.3: LinUCB with noise

```
$G = O_{dxd}$
$g = 0_{d\times 1}$
for $t \in [T]$ do
    Observe context for all arms $i \in [K]$, $\mathcal{X}_t$
    Compute perturbed gram matrix $V = G + H_t$ and reward vector $b = g + h_t$
    $\hat{\theta} = V^{-1}b$
    Compute confidence-set bound, $\beta_t$
    Select arm $i_t = \text{argmax}_{i\in[K]} < \hat{\theta}, x_t(i) > +\beta_t \|x_t(i)\|_{V^{-1}}$
    Observe reward $r_{i_t,t}$
    $G = G + x_t(i_t) \cdot x_t(i_t)^\intercal$
    $g = g + r_{i_t,t} x_t(i_t)$
    Sample perturbation values, $H_t, h_t$ using the tree-based mechanism
end for
```

Tree-based Mechanism (DP): The tree-based mechanism for preserving differential privacy operates by constructing a binary tree, where each leaf node is associated with an individual entry from the input sequence, totaling ‘n’ entries. This binary tree structure is designed in such a way that every node in the tree holds a ‘noisy’ sum of the input entries present in its corresponding subtree. This ‘noisy’ sum is a privacy-preserving measure, ensuring that the original data is protected. The cumulative sums of the inputs are then calculated by combining the ‘noisy’ sums from the nodes. Due to the binary tree structure, this process involves at most log(n) ‘noisy’ sums. This efficient approach allows for the preservation of privacy while still enabling the computation of cumulative sums.

Under given set of assumptions, [22] show that the regret under joint differential privacy, quantified by ϵ, is $\tilde{O}(\sqrt{T}d^{\frac{3}{4}}/\sqrt{\epsilon})$

4.2.2.2 FedUCB

FedUCB, presented in [23], extends the algorithm presented in [22] to a federated setting. They define federated differential privacy, which states that any singular change in any agent’s selection history shouldn’t change the outcome of other agents’ selection by more than certain probability.

To formally define differential privacy in federated setting, the authors define sequences $\mathbf{S}_i = ((x_t^i, y_t^i))_{t\in T}$ and $\mathbf{S}_i' = ((x_t^{i\prime}, y_t^{i\prime}))_{t\in T}$ as neighbors if for $t \neq t'$, $(x_t^i, y_t^i) = (x_t^{i\prime}, y_t^{i\prime})$. Here, x_t^i denotes the observed context for round t by agent i, whereas y_t^i denotes the observed reward.

Definition 4.1: Federated Differential Privacy

Any mechanism A, in a multiagent setting, is $(\epsilon, \delta, M)-$federated differentially private under continuous observation if for any $i \neq j$, any t and set of sequences $\mathcal{S} = (\mathbf{S}_p)_{p \in [M]}$ and $\mathcal{S}' = (\mathbf{S}_p)_{p \in [M], p \neq i} \bigcup \mathbf{S}_i'$ where $\mathbf{S}_i$ and $\mathbf{S}_i'$ are neighbors, it holds that for agent j,

$$\mathbf{P}(A(\mathbf{S}_i) = \mathbf{S}_j) \leq e^{\epsilon} \cdot \mathbf{P}(A(\mathbf{S}_i') = \mathbf{S}_j) + \delta$$

FedUCB, a multi-agent private algorithm for both centralized and decentralized (peer-to-peer) federated learning is proposed to ensure federated differential privacy while minimising regret.

Here, ρ_{max} and ρ_{min} are the upper bounds on the perturbation matrix (and it's inverse) used to calculate shared gram matrix U.

Under a set of mild assumptions, FedUCB obtains the following pseudoregret with high probability,

$$Reg(\mathcal{A}) = \tilde{O}\left(\sqrt{MTd}\left(\log(\frac{\rho_{max}}{\rho_{min}} + \frac{T}{d\rho_{min}}) + M\sqrt{\rho_{max}} + M\right)\right)$$

Algorithm 4.4: FedUCB

```
for $j \in [M]$ do
    $G^j = O_{dxd}$
    $g^j = 0_{d \times 1}$
end for
for $t \in [T]$ do
    for Each agent $j \in [M]$ do
        Observe context for all arms $i \in [K]$, $\mathcal{X}_t^j$
        Compute combined gram matrix $V = G^j + U$ and reward vector $b = g^j + u$
        $\hat{\theta} = V^{-1}b$
        Compute confidence-set bound, $\beta_t$
        Select arm $i_t = \text{argmax}_{i \in [K]} < \hat{\theta}, x_t^j(i) > +\beta_t \|x_t^j(i)\|_{V^{-1}}$
        Observe reward $r_{i_t,t}^j$
        $G^j = G^j + x_t^j(i_t) \cdot x_t^j(i_t)^\top$
        $g^j = g^j + r_{i_t,t}^j x_t^j(i_t)$
        if $log\ det(V + x_t^j(i_t + M(\rho_{max} - \rho_{min})I) - log\ det(G^j) \geq D/\Delta_t$ then
            [$\forall$ agents] Add $G^j$ and $g^j$ to privatizer tree
            [$\forall$ agents] Fetch shared gram matrix $U$ and reward vector $u$ from the privatizer tree
            $G^j = O_{dxd}$
            $g^j = 0_{d \times 1}$
            $\Delta_t = 0$
        else
            $\Delta_t = \Delta_t + 1$
        end if
    end for
end for
```

Chapter 5

Differentially Private Federated Combinatorial Bandits with Constraints

Unlike most FL settings, there are many situations where the agents are competitive. Each agent would like to learn from others, but the part of the information it shares for others to learn from could be sensitive; thus, it desires its *privacy*. This chapter investigates a group of agents working concurrently to solve similar combinatorial bandit problems while maintaining quality constraints. Can these agents collectively learn while keeping their sensitive information confidential by employing differential privacy? We observe that communicating can reduce the *regret*. However, differential privacy techniques for protecting sensitive information makes the data noisy and may deteriorate than help to improve regret. Hence, we note that it is essential to decide *when to communicate* and *what shared data to learn* to strike a functional balance between regret and privacy. For such a federated combinatorial MAB setting, we propose a Privacy-preserving Federated Combinatorial Bandit algorithm, `P-FCB`. We illustrate the efficacy of `P-FCB` through simulations. We further show that our algorithm provides an improvement in terms of regret while upholding quality threshold and meaningful privacy guarantees.

5.1 Introduction

A large portion of the manufacturing industry follows the Original Equipment Manufacturer (OEM) model. In this model, companies (or aggregators) that design the product usually procure components required from an available set of OEMs. For example, foundries like TSMC, UMC, and GlobalFoundries handle the production of components used in a wide range of smart electronic offerings [24]. We also observe a similar trend in the automotive industry [25].

However, aggregators are required to maintain minimum *quality* assurance for their products while maximizing their revenue. Hence, they must judicially procure the components with desirable quality and cost from the OEMs. For this, aggregators need to learn the quality of components provided by an OEM. OEM businesses often have numerous agents engaged in procuring the same or similar components. In such a setting, one can employ *online learning* where multiple aggregators, referred henceforth as *agents*, cooperate to learn the qualities [26, 27]. Further, decentralized (or federated) learning is gaining traction for large-scale applications [28, 29].

In general, an agent needs to procure and utilize the components from different OEMs (referred to as *producers*) to learn their quality. This learning is similar to the exploration and exploitation problem of*Multi-armed Bandit* (MAB) [30, 31, 32, 33]. It needs sequential interactions between sets of producers and the learning agent. Further, we associate qualities, costs, and capacities with the producers for each agent. This can be modeled as a combinatorial multi-armed bandit (CMAB) [15] problem with assured qualities [31]. Our model allows the agents to maximize their revenues by communicating their history of procurements with each other to have better estimations of the qualities. Since the agents can benefit from sharing their past quality realizations, we consider them engaged in a *federated* learning process. Federated MAB often improves performance in terms of regret incurred per agent [34, 35].

Such a federated exploration/exploitation paradigm is not just limited to selecting OEMs. It is useful in many other domains such as stocking warehouse/distribution centres, flow optimization, and product recommendations on e-commerce websites [36, 37]. However, agents are competitive; thus, engaging in federated learning is not straightforward. Agents may not be willing to share their private experiences since that could negatively impact them. For example, sharing the exact procurement quantities of components specific to certain products can reveal the market/sales projections. Thus, we desire (or at times, it is necessary) to maintain privacy when engaged in federated learning. The aim of this work was to design a privacy-preserving algorithm for federated CMAB with quality assurances.

While we do not emphasize on a notion of fairness in rest of this chapter, we view the social welfare optimisation for the group of agents as an important result of this work. Engaging in federated learning decreases the fairness regret for all agents, hence decreasing the maximum regret amongst all agents.

5.1.1 Approach Outline

Privacy concerns for sensitive information pose a significant barrier to adopting federated learning. To preserve the privacy of such information, we employ the strong notion of *differential privacy* (DP) [38]. Note that naive approaches (e.g., Laplace or Gaussian Noise Mechanisms [39]) to achieve DP for CMAB may come at a high privacy cost or outright perform worse than non-federated solutions. Consequently, the primary challenge is carefully designing methods to achieve DP that provide meaningful privacy guarantees while performing significantly better than its non-federated counterpart.

To this end, we introduce `P-FCB`, a Privacy-preserving Federated Combinatorial Bandit algorithm. `P-FCB` comprises a novel communication algorithm among agents, while each agent is learning the qualities of the producers to cooperate in the learning process. Crucially in `P-FCB`, the agent only communicates within a specific time frame – since it is not beneficial to communicate in (i) earlier rounds (estimates have high error probability) or (ii) later rounds (value added by communicating is minimal). While communicating in each round reduces per agent regret, it results in a high privacy loss. P-FCB strikes an effective balance between learning and privacy loss by limiting the number of rounds in which agents communicate. Moreover, to ensure the privacy of the shared information, the agents add calibrated noise to sanitize the information a priori. P-FCB also uses error bounds generated for UCB exploration [40] to determine if shared information is worth learning.

5.1.2 Related Works

Multi-armed bandits

While we have looked at MAB and their variants in Chapter 2, we re-highlight a few works. Our work deals with combinatorial bandits (CMAB) [15, 41, 42, 43], whereby the learning agent pulls a superarm. We remark that our single-agent (non-federated) MAB formulation is closely related to the MAB setting considered in [16], but the authors there do not consider federated learning.

Privacy-preserving MAB

The authors in [44, 45] consider a differentially private MAB setting for a single learning agent, while the works in [46, 47] consider differentially private federated MAB setting. However, these works focus only on the classical MAB setting, emphasising the communication bottlenecks. There also exists works that deal with private and federated setting for the contextual bandit problem [26, 48]. However, they do not consider pulling subsets of arms. Further, Hannun et al. [48] consider privacy over the context, while Dubey and Pentland [26] consider privacy over context and rewards. Contrarily, we consider privacy over the procurement strategy used.

5.2 Preliminaries

In this section, we formally describe the problem setting and its federated extension. We also define differential privacy in our context.

5.2.1 Federated Combinatorial Multi Armed Bandits

We consider a combinatorial MAB (CMAB) setting where there are K producers and m agents. Each producer $i \in [K]$ has a capacity d_{ij} and cost c_{ij} for every agent $j \in [m]$ interacting with it. At any round $t \in \{1, 2, \ldots, T\}$, agents procure some quantity of goods from a subset of producers under given constraint(s). We denote the procurement vector of an agent j by $\mathbf{s}_j = (l_{1j}, l_{2j}, \ldots, l_{mj})$ where $l_{ij} \in [0, d_{ij}]$ is the quantity procured from producer i.

Qualities. Each agent observes a quality realisation for each unit it procured from producers. Since the quality of a single unit of good may not be easily identifiable, we characterize it as a Bernoulli random variable. This simulates if a unit was defective or not in the OEMs scenario. The expected realisation of a unit procured from a producer i is referred to as its quality, q_i. In other words, q_i denotes the probability with which a procured unit of good from producer i will have a quality realisation of one. While the producer's cost and capacity vary across agents, the quality values are indifferent based on agents.

Regret. We use r_{ij} to denote expected utility gain or revenue for the agent j by procuring a single unit from producer i, where $r_{ij} = \rho q_i - c_{ij}$ (where $\rho > 0$, is a proportionality constant). Further, the expected revenue/reward for a procurement vector $\mathbf{s}_j$, is given by $r_{\mathbf{s}_j} = \sum_{i \in [K]} l_{ij} r_{ij}$.

The goal for the agent is to maximise its revenue, under given constraints. We consider a constraint of maintaining a minimum expected quality threshold α (quality constraint), for our setting. To measure the performance of an a given algorithm A, we use the notion of regret which signifies the deviation of the algorithm from the procurement set chosen by an Oracle when mean qualities are known. For any round $t \in \{1, 2, \ldots, T\}$, we use the following to denote the regret for agent j given an algorithm A,

$$Reg^t_{A_j} = \begin{cases} r_{\mathbf{s}^*_\mathbf{j}} - r_{\mathbf{s}^t_{Aj}}, & \textit{if } s^t_{Aj} \textit{ satisfies the quality constraint} \\ L, & \textit{otherwise} \end{cases}$$

where $\mathbf{s}^*_\mathbf{j}$ denotes the procurement set chosen by an Oracle, with the mean qualities known. $\mathbf{s}^t_A$ is the set chosen by the algorithm A in round t. $L = \max_{r_\mathbf{s}}(r_{\mathbf{s}^*_\mathbf{j}} - r_\mathbf{s})$ is a constant that represents the maximum regret one can acquire. The overall regret for algorithm A is given by $Reg_A = \sum_{j \in [m]} \sum_{t \in [T]} Reg^t_{Aj}$.

Federated Regret Ratio (FRR). We introduce FRR to help quantify the reduction in regret brought on by engaging in federated learning. FRR is the ratio of the regret incurred by an agent via a federated learning algorithm A over agent's learning individually via a non-federated algorithm NF, i.e., $FRR = \frac{Reg_A}{Reg_{NF}}$. We believe, FRR is a comprehensive indicator of the utility gained by engaging in federated learning, compared to direct regret, since it presents a normalised value and performance comparison over different data sets/algorithms is possible.

Definition 5.1: FRR

For a CMAB setting with $m \geq 2$, FRR for a federated learning algorithm, A, is given by,

$$FRR = \frac{Reg_A}{Reg_{NF}} \tag{5.1}$$

where NF denotes a non-federated (single agent) algorithm.

Observe that, $FRR \approx 1$ indicates that there is not much change in terms of regret by engaging in federated learning. If $FRR > 1$, it is detrimental to engage in federated learning, whereas if $FRR < 1$, it indicates a reduction in regret. When $FRR \approx 0$, there is almost complete reduction of regret in federated learning.

Figures/model.pdf

Figure 5.1: Overview of the communication model for P-FCB: Agents interact with producers as part of the exploration and exploitation process. Agents also communicate among themselves to learn the qualities of producers. However, they share noisy data to maintain the privacy of their sensitive information.

In our setting, we consider that agents communicate with each other to improve their regret. But in general, agents often engage in a competitive setting, and revealing true procurement values can negatively impact them. For instance, knowing that a company has been procuring less than their history can reveal their strategic plans, devalue their market capital, hinder negotiations etc. We give a formalisation of the notion of privacy used in our setting in the next subsection.

5.2.2 Differential Privacy (DP)

As opposed to typical federated models, we assume that the agents in our setting may be competing. Thus, agents will prefer the preservation of their sensitive information. While a broader definition for differential privacy has been discussed in Section 3.2.3 and federated differential privacy (*FDP*) definition in Section 4.2.2.2, we state the FDP definition for the given setting. Specifically, consider the history of procurement quantities $\mathbf{H}_{ij} = (l_{ij}^t)_{t \in [T]}$ for any producer $i \in [K]$ is private to agent j. To preserve the privacy of $\mathbf{H}_{ij}$ while having meaningful utilitarian gains, we use the concept of Differential Privacy (DP). We tweak the FDP definition in 4.2.2.2 for our setting. For this, let $\mathbf{S}_j = (\mathbf{s}_j^t)_{t \in [T]}$ be complete history of procurement vectors for agent j.

Our concept of DP in a federated CMAB formalizes the idea that the selection of procurement vectors by an agent is insusceptible to any single element l_{ij}^t from another agent's procurement history. Note that the agents are not insusceptible to their own histories here.

Definition 5.2: Differential Privacy

In a federated setting with $m \geq 2$ agents, a combinatorial MAB algorithm $A = (A_j)_{j=1}^m$ is said to be $(\epsilon, \delta, m)-$differentially private if for any $u, v \in [m], s.t., u \neq v$, any t_o, any set of adjacent histories $\mathbf{H}_{iu} = (l_{iu}^t)_{t \in [T]}, \mathbf{H}_{iu}^{'} = (l_{iu}^t)_{t \in [T] \setminus \{t_o\}} \ \cup \ \bar{l}_{iu}^{t_o}$ for producer i and any complete history of procurement vector $\mathbf{S}_v$,

$$\Pr(A_v(\mathbf{H}_{iu}) \in \mathbf{S}_v) \leq e^{\epsilon} \Pr(A_v(\mathbf{H}_{iu}^{'}) \in \mathbf{S}_v) + \delta$$

The *privacy loss* variable $\mathcal{L}$ is often useful for the analysis of DP. More formally, given a randomised mechanism $\mathcal{A}(\cdot)$ and for any output o, the privacy loss variable is defined as,

$$\mathcal{L}^{o}_{\mathcal{A}(\mathbf{H})||\mathcal{A}(\mathbf{H}')} = \ln\left(\frac{\Pr[\mathcal{A}(\mathbf{H}) = o]}{\Pr[\mathcal{A}(\mathbf{H}') = o]}\right). \tag{5.2}$$

Gaussian Noise Mechanism To ensure DP, standard techniques of adding noise to values to be communicated are leveraged. The Gaussian Noise mechanism is a popular mechanism for the same and is discussed in Section 3.4.3.

In summary, Figure 6.1 provides an overview of the model considered. Recall that we aim to design a differentially private algorithm for federated CMAB with assured qualities. Before this, we first highlight the improvement in regret using the federated learning paradigm. Next, we discuss our private algorithm, P-FCB, in Section 5.4.

5.3 Non-private Federated Combinatorial Multi-armed Bandits

We now demonstrate the advantage of federated learning in CMAB by highlighting the reduction in regret incurred compared to agents learning individually. We first categorize Federated CMAB into the following two settings: (i) *homogeneous*: where the capacities and costs for producers are the same across agents, and (ii) *heterogeneous*: where the producer's capacity and cost varies depending on the agent interacting with them.

Homogeneous Setting.

The core idea for single-agent learning in CMAB involves using standard *UCB* exploration [40]. We consider an Oracle that uses the *UCB* estimates to return an optimal selection subset. In this paper, we propose that to accelerate the learning process and for getting *tighter* error bound for quality estimations, the agents communicate their observations with each other in every round. In a homogeneous setting, this allows all agents to train a shared model locally without a central planner since the Oracle algorithm is considered deterministic. We present the formal algorithm in the supplementary material (Section A). It's important to note that in such a setting, each agent has the same procurement history and the same expected regret.

Algorithm 5.1: FCB

```
Inputs : Total rounds $T$, Quality threshold $\alpha$, $\epsilon$, $\delta$, Cost set $\{\mathbf{c}\} = \{(c_i)_{i\in[K]}\}$, Capacity set
$\{\mathbf{d}\} = \{(d_i)_{i\in[K]}\}$,
$\forall j \in [m]$ Initialise $W_i$ (Total units procured from producer $i$) and $\hat{q}_i$ (quality estimate for
producer $i$)
while $t \leq \frac{3ln(mt)}{2m\epsilon_2^2}$ (Explore Phase) do
    for each agent $j \in [m]$ do
        Pick a procurement vector $\mathbf{s}^t = (1)^k$
        Observe quality realisations $\mathbf{X}^t_{\mathbf{s}^t,j}$
        (Synchronise) Communicate $\mathbf{X}^t_{\mathbf{s}^t,j}$ to all other
        agents
        $[\forall i \in [K]]\ \hat{q}_i \longleftarrow \frac{\hat{q}_i W_i + \sum_{j\in[m]} x^t_{ij}}{W_i + n}$
        $[\forall i \in [K]]\ W_i \longleftarrow W_i + n$
    end for
    $t \longleftarrow t + 1$
end while
while $t \leq T$ (Explore-Exploit Phase) do
    for each agent $j \in [m]$ do
        $[\forall i \in [K]]\ (\hat{q}_i)^+ = \hat{q}_i + \sqrt{\frac{3ln(mt)}{2mW_i}}$
        Pick a procurement vector $\mathbf{s}^t = Oracle(\{(\hat{q}_i)^+\}_{i\in[K]},$
        $\mathbf{c},, \alpha + \gamma, R)$
        Observe quality realisations $\mathbf{X}^t_{\mathbf{s}^t,j}$
        (Synchronise) Communicate $\mathbf{X}^t_{\mathbf{s}^t,j}$ to all other
        agents
        $[\forall i \in [K]]\ \hat{q}_i \longleftarrow \frac{\hat{q}_i W_i + \sum_{j\in[m]} x^t_{ij}}{W_i + ml_i}$
        $[\forall i \in [K]]\ W_i \longleftarrow W_i + ml_i$
    end for
end while
```

Further, the quality constraint guarantees for the federated case follow trivially from the single agent case ([16, Theorem 2]). Additionally, in Theorem 5.1, we prove that the upper bound for regret incurred by each agent is $O(\frac{\ln(mT)}{m})$; a significant improvement over $O(\ln T)$ regret the agent will incur when playing individually. The formal proof is provided in the supplement.

Theorem 5.1: FCB Regret

For Federated CMAB in a homogeneous setting with m agents, if the qualities of producers satisfy γ-seperatedness, then the individual regret incurred by each of the agents is bounded by $O(\frac{\ln(mT)}{m})$.

Proof. We follow a proof sketch similar to Chen et al [15]. For the rest of the proof, we consider any arbitrary agent $j \in [m]$ and omit explicit denotation.

$$
\begin{aligned}
Reg^t_A &= \sum_{t=1}^{\tau-1} Reg^t_A + \sum_{t=\tau}^{T} Reg^t_A \\
&\leq L\tau + \sum_{t=\tau}^{T} Reg^t_A
\end{aligned}
$$

$$
E[Reg^t_A] \leq L\tau + (\sum_{t\geq\tau}[(1-\sigma)(r_{\mathbf{s}^*}) - r_{\mathbf{s}^t_A})) + \sigma L]
$$

Let,

$$
Reg^T_u = \sum_{t\leq\tau}(1-\sigma)(r_{\mathbf{s}^*}) - r_{\mathbf{s}^t_A})
$$

Some Additional Notations

Symbol	**Description**
V^T	Number of times a sub-optimal procurement vector is chosen
F^t	Event that Oracle failed to produce ω-approximation solution
W^t_i	Total units procured from i till round t
S_b	denotes the set of bad procurement vectors
$d = argmax_{i\in[K]}\ d_i$	represents the max capacity amongst all arms
Δ^i_{min}	$\omega r_{\mathbf{s}^*} - \max\{r_{\mathbf{s}}, \mathbf{s} \in S_b, l_i \neq 0\}$
Δ_{min}	$\min_{i\in[m]} \Delta^i_{min}$
Δ^i_{max}	$\omega r_{\mathbf{s}^*} - \min\{r_{\mathbf{s}}, \mathbf{s} \in S_b, l_i \neq 0\}$

Table 5.1: Additional Notations

We can see that,

$$E[Reg_u^T] \leq E[V_T]\Delta_{max}$$

Bounding number of round in which sub-optimal procurement vector are chosen

We can use a proof similar to the proof provided in [15] to tightly bound V^T. Let each arm i have a counter Z_i associated with it. Z_i^t represents the value of Z_i after t rounds.

Counters $\{Z_i\}_{i\in[K]}$ are updated as follows,

1. After initial k rounds, $\sum_i Z_i^k = k$.

2. For round $t > k$, let $\mathbf{s}^t$ be the selected procurement vector in round t. We say round t is bad if oracle selects a bad arm.

3. For a bad round, we increase one of the counters. Let $j = argmin_{i\in[K],l_i^t\neq 0} Z_i^{t-1}$, then $Z_j^t = Z_j^{t-1}+1$ (If multiple counters have min value, select i randomly from the set).

Total number of bad rounds in first p rounds is less than or equal to $\sum_i Z_i^p$.

Let $\gamma_t = \frac{6\log(nt)}{n(f^-(\Delta_{min}))^2}$,

$$\begin{aligned}
&\sum_{i=1}^{k} Z_i^p - k(\gamma_p + 1)\\
&= \sum_{t=k+1}^{p} \mathbb{I}\{\mathbf{s}^t \in S_b\} - k\gamma_p\\
&\leq \sum_{t=k+1}^{p}\sum_{i=1}^{k}\mathbb{I}\{\mathbf{s}^t \in S_b, Z_i^t > Z_i^{t-1}, Z_i^{t-1} > \gamma_p\}\\
&\leq \sum_{t=k+1}^{p}\sum_{i=1}^{k}\mathbb{I}\{\mathbf{s}^t \in S_b, Z_i^t > Z_i^{t-1}, Z_i^{t-1} > \gamma_t\}\\
&= \sum_{t=k+1}^{p} \mathbb{I}\{\mathbf{s}^t \in S_b, \forall i\ s.t.\ l_i^t \neq 0, Z_i^{t-1} > \gamma_t\} \qquad (5.3)\\
&\leq \sum_{t=k+1}^{p} \mathbb{I}\{F^t\} + \mathbb{I}\{\neg F^t, \mathbf{s}^t \in S_b, \forall i\ s.t.\ l_i^t \neq 0, Z_i^{t-1} > \gamma_t\}\\
&\leq \sum_{t=k+1}^{p} \mathbb{I}\{F^t\} + \mathbb{I}\{\neg F^t, \mathbf{s}^t \in S_b, \forall i\ s.t.\ l_i^t \neq 0, W_i^{t-1} > \gamma_t\}
\end{aligned}$$

Eq. (5.3) holds due to the rule of updating the counters.

Now we first claim that $Pr\{\neg F^t, \mathbf{s}^t \in S_b, \forall i \ s.t. \ l_i^t \neq 0, W_i^{t-1} > \gamma_t\} \leq 2dm^{-3}t^{-2}$.

For any $i \in [K]$,

$$\begin{aligned}
&Pr[|\hat{q}_{i,W_i^{t-1}} - q_i| \geq \sqrt{\frac{3\log(mt)}{2mW_i^{t-1}}}] \\
&= \sum_{b=1}^{d(t-1)} Pr[|\hat{q}_{i,W_i^{t-1}} - q_i| \geq \sqrt{\frac{3\log(mt)}{2mb}}, b = W_i^{t-1}] \\
&\leq \sum_{b=1}^{d(t-1)} Pr[|\hat{q}_{i,W_i^{t-1}} - q_i| \geq \sqrt{\frac{3\log(mt)}{2mb}}] \\
&\leq \sum_{b=1}^{d(t-1)} 2e^{-2(ms)(\frac{3\log(mt)}{2mb})} \qquad (5.4) \\
&= 2dm^{-3}t^{-2}
\end{aligned}$$

Eq. 5.4 holds due to Hoeffding inequality.

Let $\Lambda_i^t = \sqrt{\frac{3\log(mt)}{2mW_i^{t-1}}}$.

Let $E^t = \{\forall i \in [K], |\hat{q}_{i,W_i^{t-1}} - q_i| \leq \Lambda_i^t\}$ be an event. Then by union bound on Eq. 5.4, $Pr[\neg E^t] \leq 2dm^{-3}t^{-2}$. Also, since $|(q_i^t)^+ - \hat{q}_i^t| = \Lambda_{i,t}$, that means, $E^t \implies (q_i^t)^+ \geq q_i^t, \forall i \in [K]$.

Let $\Lambda = \sqrt{\frac{3\log(mt)}{2m\gamma_t}}$ and $\Lambda^t = \max_{i\in[K]} \Lambda_i^t$.

$$E^t \implies |(q_i^t)^+ - q_i^t| \leq 2\Lambda^t \qquad (5.5)$$

$$\{\mathbf{s}^t \in S_b, \forall i \ s.t. \ l_i^t \neq 0, W_i^{t-1} > \gamma_t\} \implies \Lambda > \Lambda^t \qquad (5.6)$$

If $\{E^t, \neg F^t, \mathbf{s}^t \in S_b, \forall i \ s.t. \ l_i^t \neq 0, W_i^{t-1} > \gamma_t\}$ holds true, then using Eq. 5.5, Eq. 5.6, monotonicity of rewards and bounded smoothness property,

$$r_{\mathbf{s}^t} + f(2\Lambda) > \omega r_{\mathbf{s}^*} \qquad (5.7)$$

Since $\gamma_t = \frac{6\log(mt)}{m(f^{-}(\Delta_{min}))^2}$, $f(2\Lambda) = \Delta_{min}$. This is contradictory to definition of Δ_{min}.

$$\begin{aligned}
&Pr\{E^t, \neg F^t, \mathbf{s}^t \in S_b, \forall i \; s.t. \; l_i^t \neq 0, W_i^{t-1} > \gamma_t\} = 0 \\
&\implies \{\neg F^t, \mathbf{s}^t \in S_b, \forall i \; s.t. \; l_i^t \neq 0, W_i^{t-1} > \gamma_t\} \\
&\leq Pr[\neg E^t] \leq 2dm^{-3}t^{-2}
\end{aligned}$$

Thus,

$$\begin{aligned}
E[\sum_i^k Z_i^p] &\leq k(\gamma_p + 1) + (1-\beta)(p-k) + \sum_{t=1}^{p} 2kdm^{-3}t^{-2} \\
&\leq \frac{6k\log(mt)}{m(f^{-1}(\Delta_{min}))^2} + (\frac{\pi^2}{3} + 1)kdm^{-3} + (1-\beta)(p-k) \qquad (5.8)
\end{aligned}$$

Bounding Regret Using Eq. 5.8 and using the fact that $\beta = 0$

$$\begin{aligned}
E[Reg_u^T] &= E[V^T]\Delta_{max} \\
&\leq \left(\frac{\pi^2}{3}dm^{-3} + \frac{6\log(mt)}{m(\frac{\Delta_{min}}{R})^2}\right)k\Delta_{max}
\end{aligned}$$

This completes our proof for regret bound in a homogeneous federated setting. □

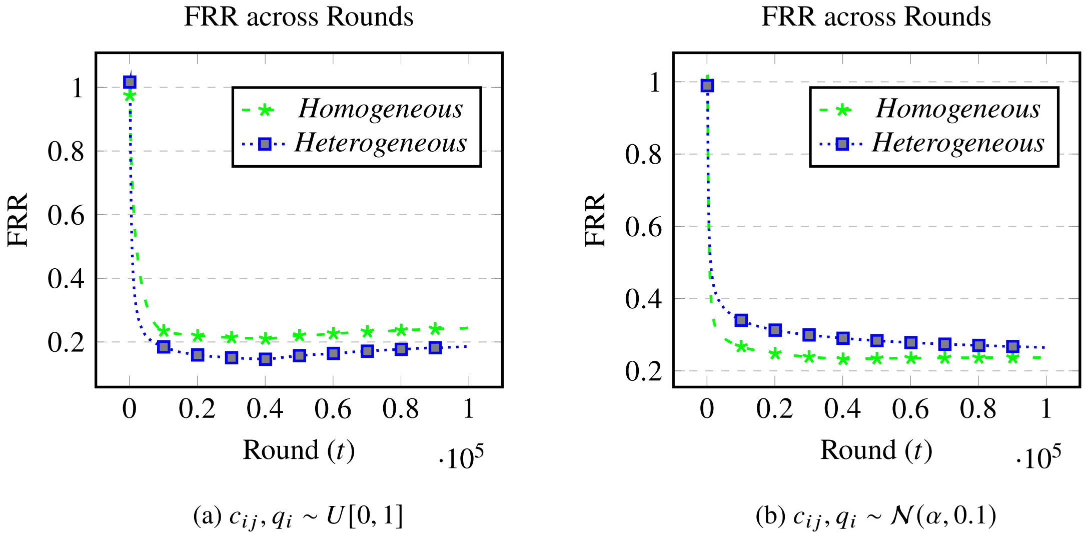
Heterogeneous Setting.

In real-world, the agents may not always have the same capacities. For such a heterogeneous setting, the regret analysis is analytically challenging. For instance, we can no longer directly use Hoeffding's inequality, needed for proving Theorem 5.1, since the procurement histories will differ across agents. Still, the intuition for regret reduction from cooperative learning carries over.

Even in a heterogeneous setting, communicating the observations allows the agent to converge their quality estimations to the mean faster and provide tighter error bounds. Even with shared quality estimates, Oracle may return different procurement vectors for different agents based on different capacities. Thus, a weighted update in estimation is essential, and the procurement vector would also need to be communicated.

We empirically demonstrate that using federated learning in heterogeneous setting shows similar FRR (ratio of regret incurred in federated setting compared to non federated setting) trend compared to homogeneous setting, over 100000 rounds for two scenarios: (i) Costs and qualities are sampled from uniform distributions, i.e. $c_{ij} \sim U[0,1]$, $q_i \sim U[0,1]$, (ii) Costs and qualities are sampled from normal distributions around the quality threshold, i.e., $c_{ij} \sim \mathcal{N}(\alpha, 0.1)$, $q_i \sim \mathcal{N}(\alpha, 0.1)$.

Fig. 5.2 depicts the results. From Fig. 5.2 we observe that the trend for both homogeneous and heterogeneous settings are quite similar. This shows that, similar to the homogeneous setting, employing federated learning reduces regret even in the heterogeneous setting.



(a) $c_{ij}, q_i \sim U[0,1]$

(b) $c_{ij}, q_i \sim \mathcal{N}(\alpha, 0.1)$

Figure 5.2: Comparing FRR values for Homogeneous and Heterogeneous Federated CMAB ($m = 10$, $k = 30$)

5.4 P-FCB: Privacy-preserving Federated Combinatorial Bandit

From Section 5.2.2, recall that we identify the procurement history of an agent-producer pair as the agent's sensitive information. We believe that the notion of DP w.r.t. the agent-producer procurement history is reasonable. A differentially private solution ensures that the probability with which other agents can distinguish between an agent's adjacent procurement histories is upper bounded by the privacy budget ϵ.

Section Outline: In this section, we first argue that naive approaches for DP are not suitable due to their lack of meaningful privacy guarantees. Second, we show that all attributes dependent on the sensitive attribute must be sanitised before sharing to preserve privacy. Third, we define a privacy budget algorithm scheme. Fourth, we formally introduce P-FCB including a selective learning procedure. Last, we provide the (ϵ, δ)-DP guarantees for P-FCB.

5.4.1 Privacy Budget and Regret Trade-off

Additive noise mechanism (e.g., Gaussian Noise mechanism [39]) is a popular technique for ensuring (ϵ, δ)-DP. To protect the privacy of an agent's procurement history within the DP framework, we can build a naive algorithm for heterogeneous federated CMAB setting by adding noise to the elements of the procurement vectors being communicated in each round.

However, such a naive approach does not suitably satisfy our privacy needs. Using the Basic Composition theorem [39], which adds the ϵs and δs across queries, it is intuitive to see that communicating in every round results in a high overall ϵ value which may not render much privacy protection in practice [49]. Consider the agents interacting with the producers for 10^6 rounds. Let $\epsilon = 10^{-2}$ for each round they communicate the perturbed values. Using Basic Composition, we can see that the overall privacy budget will be bounded by $\epsilon = 10^4$, which is practically not acceptable. The privacy loss in terms of overall ϵ grows at worst linearly with the number of rounds.

It is also infeasible to solve this problem merely by adding more noise (reducing ϵ per round) since if the communicated values are too noisy, they can negatively affect the estimates. This will result in the overall regret increasing to a degree that it may be better to not cooperatively learn. To overcome this challenge, we propose to decrease the number of rounds in which agents communicate information.

Secondly, if the sample size for the local estimates is too small, noise addition can negatively effect the regret incurred. On the other hand, if the sample size of local estimate is too large, the local estimate will have tight error bounds and deviating from the local estimate too much may result in the same.

When to Learn. Based on the above observations, we propose the following techniques to strike an effective trade-off between the privacy budget and regret.

1. To limit the growth of ϵ over rounds, we propose that communication happens only when the current round number is equal to a certain threshold (denoted by τ) which doubles in each communication round. Thus, there are only $\log(T)$ communications rounds, where density of communication rounds decrease over rounds.

2. We propose to communicate only for a specific interval of rounds, i.e., for each round $t \in [\underline{t}, \bar{t}]$. *No* communication occurs outside these rounds. This ensures that agent communication only happens in rounds when it is useful and not detrimental.

5.4.2 Additional Information Leak with Actual Quality Estimates and Noisy Weights

It is also important to carefully evaluate the way data is communicated every round since it may lead to privacy leaks. For example, consider that all agents communicate their local estimates of the producer qualities and perturbation of the total number of units procured from each producer to arrive at the estimation. We now formally analyse the additional information leak in this case. W.l.o.g. our analysis is for any arbitrarily picked producer $i \in [K]$ and agent $j \in [m]$. As such, we omit the subscripts "i" for producer and "j" for the agent. We first set up the required notations as follows.

Notations: Consider $\hat{q}^t, W^t$ as *true* values for the empirical estimate of quality and total quantity procured till the round t (not including t). Next, let $\tilde{W}^t$ denote *noisy* value of W^t (with the noise added using any additive noise mechanism for DP [39]). We have w^t as the quantity procured in round t. Last, let $\hat{q}^{obsv_t}$ denote the quality estimate based on just round t. Through these notations, we can compute $\hat{q}^{t+1}$ for the successive round $t+1$ as follows: $\hat{q}^{t+1} = \frac{W^t \times \hat{q}^t + w^t \times \hat{q}^{obsv_t}}{W^t + w^t}$.

Claim 5.1

Given $\hat{q}^t$, W^t, $\tilde{W}^t$, w^t and $\hat{q}^{obsv_t}$, the privacy loss variable $\mathcal{L}$ is not defined if $\hat{q}^t$ is also not perturbed.

Proof. If $w^t = 0$, then it follows that $\hat{q}^{t+1} = \hat{q}^t$ irrespective of $\tilde{W}^t, \tilde{W}^{t+1}$. So, if it values $\hat{q}^{t+1} \neq \hat{q}^t$ are communicated, other agents can conclude that w^t cannot be zero. This implies that the privacy loss variable $\mathcal{L}$ (Eq. 5.2) is not defined as an adversary can distinguish between two procurement histories. □

With Claim 5.1, we show that ϵ may not be bounded even after sanitising the sensitive data due to its dependence on other non-private communicated data. This is due to the fact that the local mean estimates are a function of the procurement vectors and the observation vectors. Thus, it becomes insufficient to just perturb the quality estimates.

We propose that whenever communication happens, only procurement and observation values based on rounds since last communication are shared. Additionally, to communicate weighted quality estimates, we use the Gaussian Noise mechanism to add noise to *both* the procurement values and realisation values. The sensitivity (Δ) for noise sampling is equal to the capacity of the producer-agent pair.

5.4.3 Privacy Budget Allocation

Since the estimates are more sensitive to noise addition when the sample size is smaller, we propose using monotonically decreasing privacy budget for noise generation. Formally, let total privacy budget be denoted by ϵ with $(\epsilon^1, \epsilon^2, \ldots)$ corresponding to privacy budgets for communication rounds $(1, 2, \ldots)$. Then, we have $\epsilon^1 > \epsilon^2 > \ldots$. Specifically, we denote ϵ^z as the privacy budget in the z^{th} communication round, where $\epsilon^z \longleftarrow \frac{\epsilon}{2\times\log(T)} + \frac{\epsilon}{2^{z+1}}$.

Algorithm 5.2: CheckandUpdate($W, \tilde{w}, Y, \tilde{y}, \omega_1, \omega_2, n, t$)

```
$\hat{q} \longleftarrow \frac{Y}{W}$
if $\frac{\tilde{y}}{\tilde{w}} \in \left[\hat{q} - \omega_1\sqrt{\frac{3ln(mt)}{2W}}, \hat{q} + \omega_1\sqrt{\frac{3ln(mt)}{2W}}\right]$ then
    $W \longleftarrow W + \omega_2\tilde{w}$
    $Y \longleftarrow Y + \omega_2\tilde{y}$
end if
return $W, Y$
```

5.4.4 P-FCB: Algorithm

Based on the feedback from the analysis made in previous subsections, we now present a private federated CMAB algorithm for the heterogeneous setting, namely P-FCB. Algorithm 5.3 formally presents P-FCB. Details follow.

Algorithm 5.3 Outline. The rounds are split into two phases. During the initial pure exploration phase (Lines 6-22), the agents explore all the producers by procuring evenly from all of them. The length of the pure exploration phase is carried over from the non-private algorithm. In this second phase (Lines 23-38), explore-exploit, the agents calculate the UCB for their quality estimates. Then the Oracle is used to provide a procurement vector based on the cost, capacity, UCB values as well as the quality constraint (α). Additionally, the agents communicate their estimates as outlined in Sections 5.4.1 and 5.4.2. The agents update their quality estimates at the end of each round using procurement and observation values (both local and communicated), Lines 19 and 36.

$$\begin{aligned} w_{i,j}^{t+1} \longleftarrow w_{i,j}^{t} + l_{i,j}^{t} \ ; \ W_{i,j}^{t+1} \longleftarrow W_{i,j}^{t} + l_{i,j}^{t} \\ y_{i,j}^{t+1} \longleftarrow y_{i,j}^{t} + x_{i,j}^{t} \ ; \ Y_{i,j}^{t+1} \longleftarrow Y_{i,j}^{t} + x_{i,j}^{t} \\ q_{i,j}^{t+1} \longleftarrow \frac{Y_{i,j}^{t+1}}{W_{i,j}^{t+1}} \end{aligned} \tag{5.9}$$

Algorithm 5.3: P-FCB

```
Inputs : Total rounds $T$, Quality threshold $\alpha$, $\epsilon$, $\delta$, Cost set $\{\mathbf{c}_j\} = \{(c_{i,j})_{i\in[K]}\}$, Capacity set $\{\mathbf{d}_j\} = \{(d_{i,j})_{i\in[K]}\}$, Start round $\underline{t}$, Stop round $\bar{t}$
Initialization Step:
$t \longleftarrow 0$, $\tau \longleftarrow 1$
$[\forall i \in [K], \forall j \in [m]]$ Initialise total and uncommunicated procurement $(W_{i,j}, w_{i,j})$ and realisations $(Y_{i,j}, y_{i,j})$
while $t \leq \frac{3ln(yT)}{2m\zeta^2}$ (Pure Explore Phase) do
  for all the agents $j \in [m]$ do
    Pick procurement vector $\mathbf{s}^t_j = (1)^k$ and observe quality realisations $\mathbf{X}^t_{\mathbf{s}^t_j, j}$.
    $[\forall i \in [K]]$ Update $W^{t+1}_{i,j}, w^{t+1}_{i,j}, Y^{t+1}_{i,j}, y^{t+1}_{i,j}$ using Eq. 5.9
    if $t \in [\underline{t}, \bar{t}]$ and $t \geq \tau$ then
      Communication round
      $[\forall i \in [K]]$ Calculate $\tilde{w}_{i,j}, y_{\tilde{i,j}}$ according to Eq. 5.10,5.11
      for each agent $z \in [m]/j$ do
        Send $\{\tilde{w}_{i,j}, \tilde{y}_{i,j}\}$ to agent $z$
        $[\forall i \in [K]]$ $W^{t+1}_{i,z}, Y^{t+1}_{i,z} \longleftarrow$ CheckandUpdate($W^{t+1}_{i,z}, \tilde{w}_{i,j}, Y^{t+1}_{i,z}, \tilde{y}_{i,j}, .$)
      end for
      $[\forall i \in [K]]$ $w^{t+1}_{i,j} \longleftarrow 0$, $y^{t+1}_{i,j} \longleftarrow 0$
      $\tau \longleftarrow 2 \times \tau$
    end if
    Update quality estimate
    $t \longleftarrow t + 1$
  end for
end while
while $t \leq T, \forall j \in [m]$ (Explore-Exploit Phase) do
    $[\forall i \in [K]]$ Calculate the upper confidence bound of quality estimate, $(\hat{q}^t_{i,j})^+$
    Pick procurement vector using $\mathbf{s}^t_j = \textbf{Oracle}((\hat{q}^t_{i,j})^+, \mathbf{c}_j, \mathbf{d}_j, .)$ and observe its realisations $\mathbf{X}^t_{\mathbf{s}^t_j, j}$.
    $[\forall i \in [K]]$ Update $W^{t+1}_{i,j}, w^{t+1}_{i,j}, Y^{t+1}_{i,j}, y^{t+1}_{i,j}$ using Eq. 5.9
    if $t \in [\underline{t}, \bar{t}]$ and $t \geq \tau$ then
        Communication round
        $[\forall i \in [K]]$ Calculate $\tilde{w}_{i,j}, \tilde{y_{i,j}}$ according to Eq. 5.10,5.11
        for each agent $z \in [m]/j$ do
            Send $\{\tilde{w}_{i,j}, \tilde{y}_{i,j}\}$ to agent $z$
            $[\forall i \in [K]]$ $W^{t+1}_{i,z}, Y^{t+1}_{i,z} \longleftarrow \text{CheckandUpdate}(W^{t+1}_{i,z}, \tilde{w}_{i,j}, Y^{t+1}_{i,z}, \tilde{y}_{i,j}, .)$
        end for
        $[\forall i \in [K]]$ $w^{t+1}_{i,j} \longleftarrow 0, y^{t+1}_{i,j} \longleftarrow 0$
        $\tau \longleftarrow 2 \times \tau$
    end if
    Update quality estimate
    $t \longleftarrow t + 1$
end while
```

Noise Addition. From Section 5.4.2, we perturb both uncommunicated procurement and realization values for each agent-producer pair using the Gaussian Noise mechanism. Formally, let $w^t_{i,j}, y^t_{i,j}$ be the uncommunicated procurement and realization values. Then $\tilde{w}_{i,j}, \tilde{y}_{i,j}$ are communicated, which are calculated using the following privatizer,

$$\tilde{w}_{i,j} = w^t_{i,j} + \mathcal{N}(0, \frac{2k^2_{i,j}\log(1.25/\delta)}{(\epsilon^z)^2}) \tag{5.10}$$

$$\tilde{y}_{i,j} = y^t_{i,j} + \mathcal{N}(0, \frac{2k^2_{i,j}\log(1.25/\delta)}{(\epsilon^z)^2}) \tag{5.11}$$

where ϵ^z is the privacy budget corresponding to the z^{th} communication round.

What to Learn. To minimise the regret incurred, we propose that the agents selectively choose what communications to learn from. Weighted confidence bounds around local estimates are used to determine if a communication round should be learned from. Let $\xi^t_{i,j} = \sqrt{\frac{3ln(t)}{2\sum_{z\in\{1,2,\ldots,t\}} l^z_{i,j}}}$ denote the confidence interval agent j has w.r.t. local quality estimate of producer i. Then, the agents only selects to learn from a communication if $\hat{q}^t_{i,j} - \omega_1\xi^t_{i,j} < q_{(communicated)i,j} < \hat{q}^t_{i,j} + \omega_1\xi^t_{i,j}$ where ω_1 is a weight factor and $q_{(communicated)i,j} = \frac{\tilde{y}_{i,j}}{\tilde{w}_{i,j}}$.

The local observations are weighed more compared to communicated observations for calculating overall estimates. Specifically, $\omega_2 \in [0, 1]$ is taken as the weighing factor for communicated observations.

5.4.5 P-FCB: (ϵ, δ)-DP Guarantees

In each round, we perturb the values being communicated by adding Gaussian noises satisfying (ϵ', δ')-DP to them. It is a standard practice for providing DP guarantees for group sum queries. Let $\mathcal{A}$ be a randomised mechanism which outputs the sum of values for a database input using Gaussian noise addition. Since Oracle is deterministic, each communication round can be considered a post-processing 3.3.2 of $\mathcal{A}$ whereby subset of procurement history is the the database input. Thus making individual communication rounds satisfy (ϵ', δ')-DP.

The distinct subset of procurement histories used in each communication round can be considered as independent DP mechanisms. Using the Basic Composition theorem, we can compute the overall (ϵ, δ)-DP guarantee. In P-FCB, we use a target privacy budget, ϵ, to determine the noise parameter σ in each round based on Basic composition. Thus, this can be leveraged as a tuning parameter for privacy/regret optimisation.

5.5 Experimental Analysis

In this section, we compare P-FCB with non-federated and non-private approaches for the combinatorial bandit (CMAB) setting with constraints. We first explain the experimental setup, then note our observations and analyze the results obtained.

5.5.1 Setup

For our setting, we generate costs and qualities for the producers from: (a) uniform distributions, i.e., $q_i, c_{ij} \sim U[0,1]$ (b) normal distributions, i.e., $q_i, c_{ij} \sim \mathcal{N}(\alpha, 0)$. For both cases, the capacities are sampled from a uniform distribution, $d_{ij} \sim U[1, 50]$. We use the following tuning parameters in our experiments: $\alpha = 0.4$, $\delta = 0.01$ (i.e., $\delta < 1/n$), $\underline{t} = 200$, $\bar{t} = 40000$, $\omega_1 = 0.1$, $\omega_2 = 10$. For our Oracle, we deploy the *Greedy SSA* algorithm presented in Deva et al. [16]. Further, to compare P-FCB's performance, we construct the following two *non-private* baselines:

1. Non-Federated. We use the single agent algorithm for subset selection under constraints proposed in Deva et al. [16]. It follows *UCB* exploration similar to P-FCB but omits any communication done with other agents.

2. FCB. This is the non-private variant of P-FCB. That is, instead of communicating $\tilde{w}_{ij}$ and $\tilde{y}_{ij}$, the true values w_{ij}^t and y_{ij}^t are communicated.

We perform the following experiments to measure P-FCB's performance:

- EXP1: For fixed $m = 10$, $k = 30$, we observe the regret growth over rounds (t) and compare it to non-federated and non-private federated settings.

- EXP2: For fixed $m = 10$, $k = 30$, we observe FRR (ratio of regret incurred in federated setting compared to non federated setting) at $t = 100000$ while varying ϵ to see the regret variance w.r.t. privacy budget.

- EXP3: For fixed $\epsilon = 1$, $k = 30$, we observe average regret at $t = 100000$ for varying m to study the effect of number of communicating agents.

For EXP1 and EXP2, we generate 5 instances by sampling costs and quality from both Uniform and Normal distributions. Each instance is simulated 20 times and we report the corresponding average values across all instances. Likewise for EXP3, instances with same producer quality values are considered with costs and capacities defined for different numbers of learners. For each instance, we average across 20 simulations. We provide the complete code-base along with the supplementary material.

5.5.2 Results

- *EXP1*. P-FCB shows significant improvement in terms of regret (Fig. 5.3) at the cost of relatively low privacy budget. Compared to FCB, P-FCB ($\epsilon = 1$) and Non-federated incurs 136%,233% more regret respectively for uniform sampling and 235%, 394% more regret respectively for normal sampling. This validates efficacy of P-FCB.

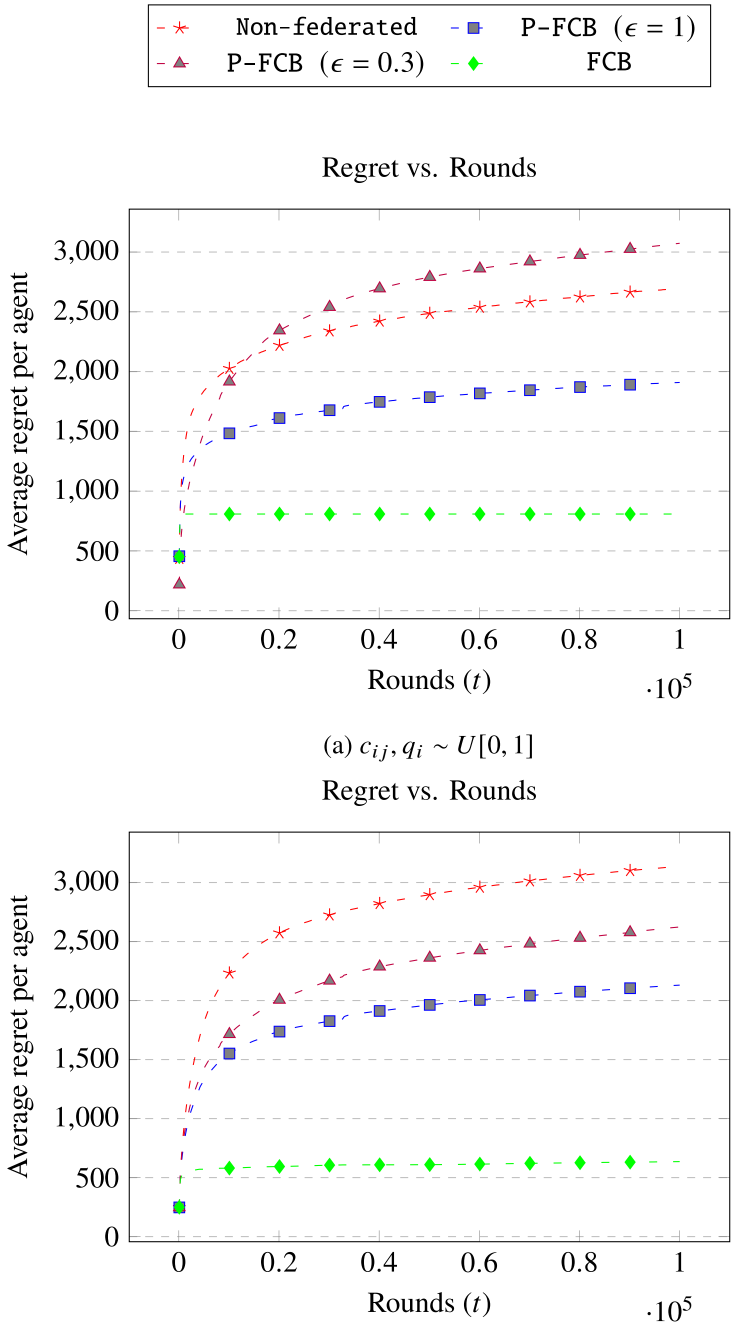
- *EXP2*. We study the performance of the algorithm with respect to privacy budget (Fig. 5.4). We observe that according to our expectations, the regret decreases as privacy budget is increased. This decrease in regret is sub-linear in terms of increasing ϵ values. This is because as privacy budget increases, the amount of noise in communicated data decreases.

- *EXP3*. We see (Fig. 5.5) an approximately linear decrease in per agent regret as the number of learning agents increases. This reinforces the notion of reduction of regret, suggested in Section 5.3, by engaging in federated learning is valid in a heterogeneous private setting.

Discussion: Our experiments demonstrate that P-FCB, through selective learning in a federated setting, is able to achieve a fair regret and privacy trade-off. P-FCB achieves reduction in regret (compared to non-federated setting) for low privacy budgets.

With regards to hyperparamters, note that lower ω_2 suggests tighter bounds while selecting what to learn, implying a higher confidence in usefulness of the communicated data. Thus, larger values for ω_1 can be used if ω_2 is decreased. In general, our results indicate that it is optimal to maintain the value $\omega_1 \cdot \omega_2$ used in our experiments. Also, the communication start time, should be such that the sampled noise is at-least a magnitude smaller than the accumulated uncommunicated data (e.g., $\underline{t} \approx 200$). This is done to ensure that the noisy data is not detrimental to the learning process.

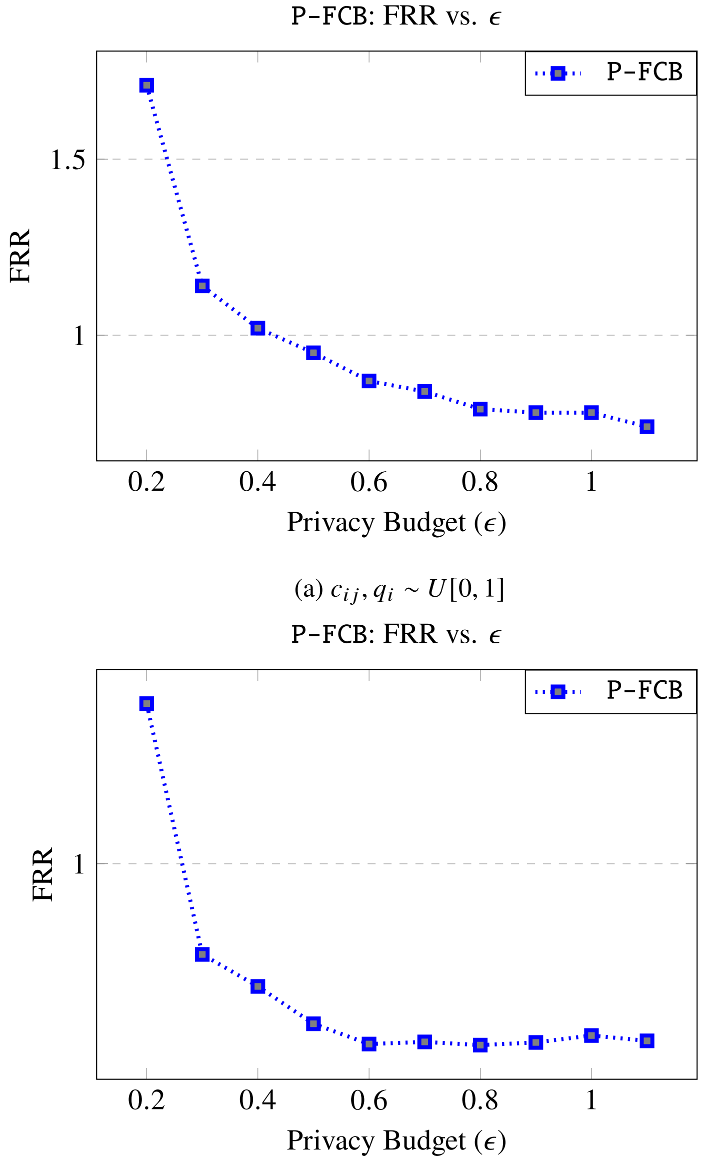
The DP-ML literature suggests a privacy budget $\epsilon < 1$ [49]. From Fig. **??**, we note that P-FCB performs well within this privacy budget. While our results achieve a fair regret and privacy trade-off, in future, one can further fine tune these hyperparameters through additional experimentation and/or theoretical analysis.



(a) $c_{ij}, q_i \sim U[0, 1]$

(b) $c_{ij}, q_i \sim \mathcal{N}(\alpha = 0.4, 0.2)$

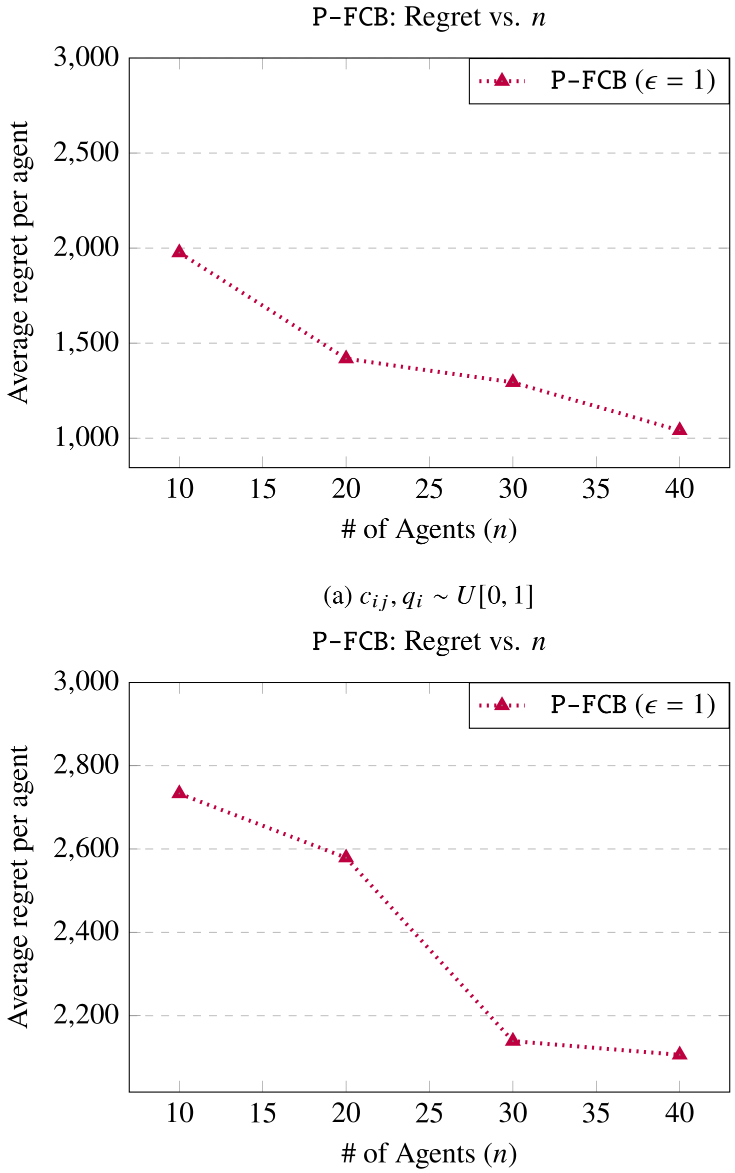
Figure 5.3: `EXP1`: Regret Comparison across rounds ($m = 10$, $k = 30$)



(a) $c_{ij}, q_i \sim U[0, 1]$

(b) $c_{ij}, q_i \sim \mathcal{N}(\alpha = 0.4, 0.2)$

Figure 5.4: EXP2: FRR for P-FCB while varying privacy budget ϵ (with $n = 10$, $m = 30$, $t = 100000$)



(a) $c_{ij}, q_i \sim U[0, 1]$

(b) $c_{ij}, q_i \sim \mathcal{N}(\alpha = 0.4, 0.2)$

Figure 5.5: EXP3: Average regret per agent with P-FCB with respect to varying number of learners n (with $\epsilon = 1$, $t = 100000$)

5.6 Conclusion

This work focuses on learning agents which interact with the same set of producers (“arms") and engage in federated learning while maintaining privacy regarding their procurement strategies. We first looked at a non-private setting where different producers’ costs and capacities were the same across all agents and provided theoretical guarantees over optimisation due to federated learning. We then show that extending this to a heterogeneous private setting is non-trivial, and there could be potential information leaks. We propose P-FCB which uses *UCB* based exploration while communicating estimates perturbed using Gaussian method to ensure differential privacy. We defined a communication protocol and a selection learning process using error bounds. This provided a meaningful balance between regret and privacy budget. We empirically showed notable improvement in regret compared to individual learning, even for considerably small privacy budgets.

5.7 Notational Summary

For ease of reference, Table 5.2 lists important notations used in this chapter.

Symbol	**Description**
K	Number of arms
$\mathcal{K} = [K] = \{1, 2, \ldots, K\}$	Set of arms
T	Number of rounds
m	Number of learning agents
i_t	Arm chosen at round t
d_{ij}	Capacity threshold between agent j and producer i
c_{ij}	Cost of procurement from i for agent j
s_j	Procurement vector of agent j
l_{ij}	Quantity procured from i by agent j
q_i	Quality estimate for producer i
W_i	Total units procured from producer i
r_{ij}	Expected reward/revenue
S_b	Set of bad procurement vectors
Reg_{A_j}	Regret for agent j given an algorithm A

Table 5.2: Differentially Private Federated Combinatorial Bandits with Constraints: Important Notations

Chapter 6

Exposure of Fairness in Multi-agent Contextual Bandits with Privacy Guarantees

This chapter considers the contextual multi-armed bandit (CMAB) problem with fairness and privacy guarantees in a federated environment. We consider merit-based exposure as the desired *fair* outcome, which provides exposure to each action in proportion to the reward associated. We model the algorithm's effectiveness using fairness regret. The primary challenge in extending the existing privacy framework is designing the communication protocol for communicating required information across agents. A naive protocol can either lead to weaker privacy guarantees or higher regret. We design a novel communication protocol that allows for (i) Sub-linear theoretical bounds on fairness regret for Fed-FairX-LinUCB and comparable bounds for the private counterpart, Priv-FairX-LinUCB (relative to single-agent learning), (ii) Effective use of privacy budget in Priv-FairX-LinUCB. We demonstrate the efficacy of our proposed algorithm with extensive simulations-based experiments. We show that both Fed-FairX-LinUCB and Priv-FairX-LinUCB achieve near-optimal fairness regret.

6.1 Introduction

Linear contextual bandits [50], as explored in Section 2.5 associate dynamic contexts with each action by assuming that the reward for each action is modeled as a fixed but unknown linear combination of the context and thus aims to learn these linear weights for maximizing the reward of a single learning agent. Multiple agents can collaborate in many real-world applications such as crowdsourcing, service procurement, and recommender systems for better effective learning [51, 52, 53]. For example, in crowdsourcing, requesters (agents) of similar tasks intend to learn the qualities of a pool of workers (actions), which are context-dependent. In such examples, agents can help each other by collaborating to learn the correlation between worker attributes (action context) and task completion proficiency (rewards) faster.

Such collaborative learning should be allowed without sharing sensitive data (such as specific worker selection in any given round) among the agents while allowing for effective learning, i.e., it should protect the privacy of individual agents' sensitive information. Existing literature model collaborative learning with privacy requirements via the paradigm of federated learning for practical collaboration [54, 55, 56]. Recent works [53, 23] have explored differential privacy guarantees in federated bandits which extend bandit problem in federated settings.

In many practical applications, actions often involve interactions with humans, e.g., workers in crowdsourcing. Here, it becomes crucial to ensure that each action receives sufficient exposure. Traditional bandit approaches exhibit a *"winner takes all"* behaviour [21], which consistently favors the optimal action and deprives other actions of opportunity, leading to starvation among actions. We address this issue by considering *fairness of exposure* [21, 57, 58] in multi-agent contextual bandit problems. Other fairness notions, as explored in Chapter 4, in the context of bandit problems, such as guaranteeing minimum exposure to each action [20], group fairness, and fair treatment [59] depend solely on the rewards or prioritize fairness for the learning agents rather than the individual actions.

On the other hand, fairness of exposure ensures proportionality [60, 61] for the actions, meaning that every action would be selected proportional to its merit/reward. This is an essential indicator of individual fairness in ML algorithms and proportionality in game theoretical frameworks. The algorithm in [21], discussed in Section 4.1.2.2, provides fairness of exposure guarantees but works only for a single-agent setting. There are a few works [62, 63] that provide fairness guarantees in a federated setting; however, these works consider fairness for agents instead of actions.

Fairness of exposure in bandits focuses on minimizing *fairness regret*, which measures the deviation of action selection policy from the optimal policy satisfying fairness. This work provides fairness regret guarantees in the federated setting while ensuring privacy. One naive way to ensure fairness in federated learning is to adapt existing single-agent bandit techniques (e.g., from [21]) by integrating suitable *communication protocol*. Following a naive communication protocol, all agents will communicate with each other in every round by sharing all their information about the actions. It can be easily proved the naive communication protocol, as expected, offers better fairness guarantees than the single-agent setting. However, it leads to maximum privacy leakage. Another extreme is not to allow any communication among agents. It leads to maximum privacy, but in the absence of collaborative learning, the regret blooms in terms of the number of agents. If we adopt existing communication protocols designed for privacy preservation in federated bandits, we observe that they fail to offer good regret guarantees. In summary, developing an intelligent communication protocol that provides a regret bound that is sub-linear in the number of agents and extends to the private setting is essential.

This work designs a novel communication protocol for federated bandits while learning generalizes the techniques from FairX-LinUCB [21], an algorithm designed for a single-agent setting, to a federated setting. We call our algorithm as *Fed-FairX-LinUCB*. Our communication protocol is scalable to differentially private methods since the number of communication rounds is bounded while ensuring fairness given the bounded communication gaps. We denote the privacy-ensuring version of the proposed algorithm by *Priv-FairX-LinUCB*.

6.1.1 Related Work

This work leverages federated learning, which does large dataset querying. Analysing bandit problems in a federated setting [51, 52] has been an important exploration of cooperative learning in recent times. We use differential privacy, introduced by [64], to provide privacy for context/reward information. [65] and [66] introduced the notion of differential privacy under continual observation using a **tree-based algorithm**, which we leverage. This method has seen utilisation across several online learning problems [67, 68, 54, 69]. [22] (Section 4.2.1) studies differential privacy for the traditional contextual bandit setting but is limited to a single learning agent. Differential private federated bandits have been studied in [70] and [53] (Chapter 5). However, our work is closely related to the important work of [23] 4.2.2.2, extending it for non-traditional bandit optimisation.

While significant progress has been made in traditional bandits, but as discussed in Chapter 4 bandits with fairness objectives have only recently gained popularity. [59], propose bandit fairness which is achieved by ensuring that a better arm is always chosen with at least the same likelihood as a worse arm. Several other works, including [71, 20], aim to guarantee a minimum exposure for arms in the stochastic bandit problem. Whereas, [62] and [63] define fairness for a multi-agent setting, but fairness with respect to the agents rather than actions is considered.

The notion of fairness, for actions, in the aforementioned works is modelled as a constraint rather than a desired outcome, with reward maximisation being the primary objective. In our work, we use the concept of fairness of exposure, introduced by [21] (Section 4.1.2.2) for the single-agent setting, which is an objective-oriented notion of fairness that addresses the problem of starvation among actions. Additionally, it is important to highlight that no work had previously studied proportionality-based fairness in a federated bandit setting with respect to the actions.

6.2 Preliminaries

In this section we outline the necessary preliminaries. While most of the notations from the previous chapters is still followed, the notations are re-iterated to avoid any ambiguity as well as to provide a continuity between previously introduced notations and new notations.

6.2.1 Setting and Notations

We abstract the problem as a federated contextual bandit setting where each of $M = [m]$ agents are learning the mapping from context $x_t(i) \in \mathbb{R}^d$ to reward. The set of context observed in each round is given by $\mathcal{D}$, and here the context instance of traditional arm is considered (and denoted by i). The bandit algorithm runs for T rounds, where, at each round t, an agent $j \in M$ observes a context vector $\mathcal{X}_t^j = (x_t^j(i))_{i \in \mathcal{D}}$ ($\| x_t^j(i) \|_2 \leq 1; \forall i$) with $x_t^j(i) \in \mathbb{R}^d$ and selects an action i_t^j. Each agent observes a different context vector and selects an action independently at each round t. The agent obtains a reward for a selected action i_t^j at time t which we represent as $y_t^j(i_t^j) = \theta^* \cdot x_t^j(i_t^j) + \eta_t(i_t^j)$. Here, $\theta^* \in \mathbb{R}^d$ is an unknown but fixed parameter. As standard in the literature, $\eta_t(i_t^j)$ is a noise parameter, which is i.i.d. sub-Gaussian with mean 0. Thus, the expected reward for an action i at time t, for an agent j, is given by $\mathbb{E}[y_t^j(i)] = \theta^* \cdot x_t^j(i)$. We denote this reward by the quantity $\mu_a \mid \mathcal{X}_t^j$ representing the expected reward for an action i, when j^{th} agent is observing the context vector $\mathcal{X}_t^j$. Note that θ^* (the true parameter) is the same for all the agents and is learned by the agents till time T in a collaborative fashion while preserving the privacy of their contexts/reward observations and satisfying the fairness guarantees.

We denote the set of available contexts to all the agents at time t as $\mathcal{X}_t = (\mathcal{X}_t^j)_{j \in M}$, $\mathcal{X}^j = (\mathcal{X}_t^j)_{t=1}^{t=T}$ and $\mathcal{X} = \{\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^m\}$. The goal of each agent i is to implement a policy $\pi_t^j(\mathcal{X}_t^j)$ which denotes the vector of probabilities of action selection by j^{th} agent at time t. The probability of selecting action i is denoted by $\pi_t^j(i, \mathcal{X}_t^j)$. Instead of maximizing the reward, each agent needs to ensure fairness amongst the actions so that all actions get a fair fraction of chances to avoid otherwise observed "winner takes it all" [72] problem. Specifically, this setting aims to learn a policy that selects actions with probabilities proportional to their merit. Note that the objective here is to learn the fair policy rather than the optimal-reward policy.

Agents assign a merit score function f^j over the actions based on their expected rewards for the given context. $f^j : \mathbb{R}^+ \to \mathbb{R}^+$ where $f^j(\mu_i \mid \mathcal{X}_t^j)$ denotes the score assigned by agent j for the action i when observed context is $\mathcal{X}_t^j$. Each agent then needs to implement the policy such that the following fairness constraint, which is denoted as fairness of exposure, is satisfied:

$$\frac{\pi_t^j(i, \mathcal{X}_t^j)}{f^j(\mu_i \mid \mathcal{X}_t^j)} = \frac{\pi_t^j(i', \mathcal{X}_t^j)}{f^j(\mu_{i'} \mid \mathcal{X}_t^j)} \; \forall i, i' \in \mathcal{D} \tag{6.1}$$

f^j quantifies the utility of rewards derived from an arm for the agent. We assume Minimum merit and Lipschitz continuity properties on merit function [21]. The minimum merit property provides a lower bound on the merit function, i.e. $\min_\mu f^j(\mu) \geq \gamma$, $\forall j \in M$ for some $\gamma > 0$. Lipschitz continuity property assumes that the merit function is Lipschitz continuous, i.e., $\forall \mu_1, \mu_2, j \in M$, $|f^j(\mu_1) - f^j(\mu_2)| \leq L|\mu_1 - \mu_2|$ for some $L > 0$.

We denote the optimal policy by $\pi_*^j(\mathcal{X}_t^j)$ when θ^* is known, i.e., at round t, it satisfies fairness condition (Eq. 6.1). Note that given a context vector the optimal policy, $\pi_*^j(.)$, does not depend on round t, $\pi_*^j(i, \mathcal{X}_t^j) = \frac{f^j(\theta^* \cdot x_t^j(i))}{\sum_{i' \in \mathcal{D}} f^j(\theta^* \cdot x_t^j(i'))}$. Typically, θ^* being unknown, each agent is learning θ^* and in turn the optimal policy through algorithm $\mathcal{A}$ over the rounds, taking actions using policy $\pi_t^j(\cdot)$. $\hat{\theta}_t^j$ is used to denote the learnt θ^* for agent j at time t. Unlike the optimal policy, $\pi_t^j(\cdot)$ is round dependent. For agent j at round t the *instantaneous fairness regret* is defined as: $FR_t^j(\mathcal{A}, \mathcal{X}_t^j) = \sum_{i \in \mathcal{D}} |\pi_*^j(i, \mathcal{X}_t^j) - \pi_t^j(i, \mathcal{X}_t^j)|$. As these agents learn about the same actions, they can communicate with each other about their estimates of θ^* and learn it faster, reducing the per-agent fairness regret. We assume that all the agents deploy the same learning algorithm. Thus, we define the *fairness regret* defined as:

Figures/Slide2.jpg

Figure 6.1: Fairness of exposure in Bandits

Definition 6.1: Fairness Regret

For a learning algorithm $\mathcal{A}$, we define fairness regret as $FR(\mathcal{A}, T, \mathcal{X}) = \frac{1}{m}\sum_{j\in M} FR^j(\mathcal{A}, T, \mathcal{X}^j)$ where $FR^j(\mathcal{A}, T, \mathcal{X}^j) = \sum_{t=1}^{T} FR_t^j(\mathcal{A}, \mathcal{X}_t^j)$

Henceforth, we will avoid using $\mathcal{X}_t^j$ from fairness regret to avoid notation clutter. Additionally, since we are bounding it only for the algorithms in the paper, we refer to the above quantities as FR_t^j, FR_j, FR. We also use $FR^j([T_1, T_2])$ to denote $\sum_{t=T_1}^{t=T_2} FR_t^j$ and similarly $FR([T_1, T_2])$.

6.2.2 Why Fairness of Exposure?

We motivate with a single agent setting who is interested in assigning tasks to 3 workers with unknown completion times. Let the optimal task assignment (according to Eq. 1)distribution be $[0.14, 0.28, 0.56]$, where faster worker is assigned more tasks, if the goal is to minimize total project completion time while ensuring exposure guarantees to the workers.

Traditional regret optimization finds the best worker which does not lead to balanced/fairer task allocation. While some approaches try to incorporate fairness into bandit algorithms, they often fall short in the task assignment scenario:

- Delta-fairness [73, 74], which prioritizes arms (workers) with higher rewards will essentially lead to giving maximum tasks to optimal (faster) worker, in this case the worker 3, however it does not provide any exposure guarantee.

- Minimum share fairness [20, 71]ensures each worker receives a minimum fraction of tasks. Utility optimisation in this case relies on knowing expected completion times, which are unknown in our problem. This makes its effectiveness uncertain.

In contrast, proportionality-based fairness offers a more promising approach by directly aligning fairness with utility optimization. Furthermore, when workers are involved in multiple projects simultaneously, (i.e., multiple agents are learning about the workers) federated learning with differential privacy can further optimize task assignment by sharing limited information privately, leading to faster learning and improved project completion times.

6.2.3 Privacy Requirements

In this work, similar to Chapter 5, privacy over the agent-action interaction is considered, i.e., for any agent j, we consider that the context vectors ($\mathcal{X}^j$) and the observed feedback ($(y_t^j(i_t^j))_{t\in[T]}$) should be kept private. Considering that agent only needs to store $x_t^j(i_t^j)$ for feedback estimation, we use the differential privacy definition with respect $(x_t^j(i_t^j), y_t^j(i_t^j))_{t\in[T]}$. Here, the differential privacy notion matches the one defined in 4. Let us consider two sets $\mathbf{S}_j = (x_t^j(i_t^j), y_t^j(i_t^j))_{t\in[T]}$ and $\mathbf{S}_j{}' = (x_t^j(i_t^j)', y_t^j(i_t^j)')_{t\in[T]}$. They are considered to be $t' - neighbors$ if at all time steps $t \neq t'$, $(x_t^j(i_t^j), y_t^j(i_t^j)) = (x_t^j(i_t^j)', y_t^j(i_t^j)')$.

Definition 6.2: Federated Differential Privacy

A randomized multi-agent contextual bandit algorithm $\mathcal{A} = (\mathcal{A}^j)_{j=1}^m$, in a federated learning setting with $m \geq 2$ agents, is $(\epsilon, \delta, m)-$ federated differentially private under continual multi-agent observation if for any $i, j \in M$ such that $i \neq j$, any t and set of sequences $\mathbb{S}_j = (\mathbf{S}_k)_{k=1}^m$ and $\mathbb{S}_j{}' = (\mathbf{S}_k)_{k=1,k\neq j}^m \bigcup \mathcal{S}_j{}'$ such that $\mathbf{S}_j{}'$ and $\mathbf{S}_j$ are $t'-$neighbors, and any subset of actions $(i_t^j)_{t\in[T]} \subset \mathcal{D} \times \ldots \times \mathcal{D}$ of actions, it holds that:

$$\mathbb{P}(\mathcal{A}^j(\mathbb{S}_j) \in (i_t^j)_{t\in[T]}) \leq e^{\epsilon}.\mathbb{P}(\mathcal{A}^j(\mathbb{S}_j{}') \in (i_t^j)_{t\in[T]}) + \delta$$

Differential privacy ensures that the presence/absence of one data point does not lead to significant learning changes – for federated bandit settings, it implies small changes in the data sharing do not lead to any major action changes.

Goal: Each agent's goal is to learn θ^* while minimizing fairness regret (Definition 6.1); albeit ensuring differential privacy guarantees (Definition 6.2).

6.3 Multi-Agent Fair and Private Contextual Bandit Algorithm

The communication protocol currently used in federated bandits literature is not suitable for achieving bounded fairness regret. It is important to limit the number of communication rounds and maintain a constrained gap between communication instances in order to ensure both bounded fairness regret, and scalability with private methods. The total privacy loss, which is the composition of privacy losses incurred overall communication rounds, is proportional to the number of communication rounds. Thus, it follows that for a budgeted (fixed) total privacy loss, the maximum possible per-round privacy loss is inversely proportional to the number of communication rounds. As a result, the number of communication rounds should be bounded to control the accumulation of noise and maintain privacy within acceptable limits. At the same time, bounding the gaps between communication rounds is necessary to make fairness regret claims.

In this section, we firstly build an algorithm, Fed-FairX-LinUCB, that learns θ^* collectively amongst m agents using a novel communication protocol. We then design a privacy-preserving version, Priv-FairX-LinUCB.

We consider a group of m agents actively participating in the contextual bandit problem and maintaining synchronization through periodic communication. Algorithm 6.1 without the privatizer routine represents Fed-FairX-LinUCB. Essentially, the exact information of the agents is sent to other agents when communication is required. For any agent j, at round t, let the last synchronization round take place at instant t'. Then, there exist two sets of parameters. The first set of parameters is the set of all observations made by all m agents till round t'. We store this in terms of a shared gram matrix, $U_t = \sum_{j \in M}(\lambda I + \sum_{\tau=1}^{t'}(x_\tau^j(i_\tau^j))(x_\tau^j(i_\tau^j))^\intercal)$, and a shared vector, $u_t = \sum_{j \in M} \sum_{\tau=1}^{t'}(x_\tau^j(i_\tau^j))y_\tau^j(i_\tau^j)$. Secondly, each agent has access to its own observations since the last communication round. We note those using the gram matrix $S_t^j = \sum_{\tau=t'}^{t}(x_\tau^j(i_\tau^j))(x_\tau^j(i_\tau^j))^\intercal$ and the reward vector $s_t^j = \sum_{\tau=t'}^{t}(x_\tau^j(i_\tau^j))y_\tau^j(i_\tau^j)$, where t' was the last communication round. The agents use combined parameters for estimating the linear regression estimate, $\hat{\theta}_t^j$. For an agent j, $V_t^j = U_t + S_t^j$, $b_t^j = u_t + s_t^j, \hat{\theta}_t^j = (V_t^j)^{-1}b_t^j$.j The agents then constructs a confidence region, CR_t^j around $\hat{\theta}_t^j$. Suitable sequence $[\sqrt{\beta_t^j}]_{j \in M, t \in [T]}$ needs to be used, ensuring that with high probability $\forall j, t,\ \theta^* \in CR_t^j$. An optimistic estimate, θ_t^i is selected from CR_t^j (line 6 of Algorithm. 6.1). The agent selects the action using a policy construction, π_t^j. This ensures fairness by assigning a probability distribution for action selection based on estimated merit. We now explain our communication protocol that achieve sub-linear fairness regret.

6.3.1 Fed-FairX-LinUCB

Communication Protocol. If the agents were to communicate in every round without any optimization, they could enhance their fairness regret by order of $O(1/\sqrt{m})$. However, communicating at every round results in inefficiencies and potential privacy breaches. To address these concerns, our algorithm suggests a communication strategy allowing agents to communicate only $\lceil 2md^2 \log^2(1 + T/d) \rceil$ times while achieving comparable fairness regret performance. In our proposed approach, we suggest that the agents communicate with increasing intervals between two consecutive communication rounds during the first $\lceil \frac{T}{md^2 \log^2(1+T/d)} \rceil$ rounds (line 12-13 of Algorithm 6.1). Subsequently, they communicate only after every $\lceil \frac{T}{md^2 \log^2(1+T/d)} \rceil$ rounds. Rapid communication in the initial rounds proves beneficial in practice, considering the trend in regret is sub-linear in T. Concurrently, the number of communication rounds and the gap between the communication rounds remain bounded. The number of communication rounds is upper bounded by $O\left(\log(\lceil \frac{T}{md^2 \log^2(1+T/d))} \rceil) + md^2 \log^2(1+T/d)\right)$ while the gap between communication rounds is upper bounded by $\lceil \frac{T}{md^2 \log^2(1+T/d)} \rceil$, both of which can be trivially calculated. This distinguishes it from the communication protocols employed by [23, 53], where the gaps between communication rounds can be of the order $O(T)$, which makes it difficult to bound fairness regret. In summary, on observing the context set, each agent utilizes their estimate of θ^* to formulate a selection policy, which yields a probability distribution for choosing an action. Once an action is selected and the corresponding reward is observed, the agents update their local estimates and periodically exchange these updates with each other to enhance the accuracy of the shared estimates.

6.3.2 Priv-FairX-LinUCB

The key difference between Priv-FairX-LinUCB and Fed-FairX-LinUCB lies in the communication perturbation. In a non-private setting, we communicate exact observations about context and reward to all other agents. However, we must carefully add perturbation for the private setting to satisfy the differential privacy constraints mentioned in section **??**. In the private setting, let $\hat{U}_t^j = \sum_{\tau=1}^{t-1}(x_\tau^j(i_\tau^j))(x_\tau^j(i_\tau^j))^\intercal + H_t^j$, $\hat{u}_t^j = \sum_{\tau=1}^{t-1}(x_\tau^j(i_\tau^j))y_\tau^j(i_\tau^j) + h_t^j$ denote the perturbed contexts and rewards. Here H_t^j and h_t^j are noise additions used for perturbation. Here, $V_t^j = \sum_{j\in M}\hat{U}_t^j + S_t^j$ and $b_t^j = \sum_{j\in M}\hat{u}_t^j + s_t^j$, where S_t^j and s_t^j remains same as stated in Section 6.3.1. We note that V_t^j can also be represented as: $V_t^j = G_t^j + H_t^j$ with G_t^j denoting the gram matrix in absence of noise perturbations.

To achieve privacy, we introduce a privatized version of the synchronization process amongst the agents. We do so by using the privatizer routine, which uses a tree-based mechanism to communicate while limiting the noise addition. The tree-based mechanism for differential privacy maintains a binary tree of logarithmic depth in terms of communication rounds.

The sequential data released at communication rounds are stored at the leaf nodes, while every parent node stores the sum of the child nodes' data. In addition, noise is sampled at each node to maintain privacy. This allows for returning partial sums by adding at max p nodes if p was the depth of the tree. While our algorithm vastly differs from the FedUCB algorithm [23] in terms of objective constraint, arm selection protocol, and communication round selection, it resembles our algorithm in terms of linear regressor estimation in a federated setting.

Based on this, we can use the privatizer routine with marginal changes to ensure privacy guarantees. This is formalised in Claim 6.1 The privatizer routine is formally outlined for completeness.

6.4 Theoretical Analysis

Algorithm 6.1: Priv-FairX-LinUCB

```
**Input:** $\beta_t$, $[f^j]_{\forall j \in [m]}$, $\lambda$, $m$
**Initialization:** $\forall j \in [m]$, $V_1^j = S_1^j = U_1 = \lambda \mathbf{I}_d$, $b_1^j = s_1^j = u_1 = \mathbf{0}_d$, $\tau = 1$.
**for** $t = 1$ to $T$ **do**
  **for** $j = 1$ to $m$ **do**
    Observe contexts $\mathcal{X}_t^j$; $\hat{\theta}_t^j = (V_t^j)^{-1} b_t^j$; $\mathbf{CR}_t^j = (\theta : \|\theta - (\hat{\theta}_t^j)\|_{V_t^j} \leq \sqrt{\beta_t^j})$
    $\theta_t^j = \text{argmax}_{\theta \in CR_t^j} \sum_{i \in \mathcal{D}} \frac{f(\theta . x_t^j(i))}{\sum_{i' \in \mathcal{D}} f(\theta . x_t^j(i'))} \theta . x_t^j(i)$
    Construct Policy $\pi_t^j(i) = \frac{f(\theta_t^j . x_t^j(i))}{\sum_{i'} f(\theta_t^j . x_t^j(i'))}$
    Sample arm $i_t^j \sim \pi_t^j$ and observe reward $y_t^j(i_t^j)$
    $S_{t+1}^j = S_t^j + (x_t^j(i_t^j))(x_t^j(i_t^j))^\intercal$; $s_{t+1}^j = s_t^j + (x_t^j(i_t^j)) y_t^j(i_t^j)$
    **if** $t == \tau$ **then**
      $Sync \longleftarrow true$
      **if** $t < \lceil \frac{T}{md^2 \log^2(1+T/d)} \rceil$ **then**
        $\tau = 2\tau$
      **else**
        $\tau = \tau + \lceil \frac{T}{md^2 \log^2(1+T/d)} \rceil$
      **end if**
    **end if**
    **if** $Sync$ **then**
      $[\forall p \in M]$ Send $S_t^p, s_t^p \longrightarrow PRIVATIZER$
      $[\forall p \in M]$ Receive $\hat{U}_t^p, \hat{u}_t^p \longleftarrow PRIVATIZER$
      $[\forall p \in M]$ Communicate $\hat{U}_t^p, \hat{u}_t^p$ to others
      $[\forall p \in M]$ $U_{t+1} = \sum_{k=1}^M \hat{U}_t^k$; $u_{t+1} = \sum_{k=1}^M \hat{u}_t^k$; $S_t^p = \mathbf{0}_{d \times d}$; $s_t^p = \mathbf{0}_d$; $\Delta_t^p = 0$
      $Sync \longleftarrow false$
    **else**
      $U_{t+1} = U_t^j$; $u_{t+1} = u_t^j$; $\Delta_{t+1}^j = \Delta_t^j + 1$
    **end if**
    $V_{t+1}^j = U_{t+1} + S_{t+1}^j$; $b_{t+1}^j = u_{t+1} + s_{t+1}^j$
  **end for**
**end for**
```

Algorithm 6.2: PRIVATIZER

```
**Input:** $\epsilon, \delta, d, \tau$ (number of communication rounds), $L$ (upper bound on norm of context vector)
**Initialization:**
$n = 1 + \lceil \log \tau \rceil$
$\mathcal{T} \leftarrow$ a binary tree of depth $n$
**for** each node $i$ in $\mathcal{T}$ **do**
    Create a noise matrix: $\hat{N} \in \mathbb{R}^{d\times(d+1)}$, where $\hat{N}_{kl} \sim \mathcal{N}(0, 16n(L^2+1)^2 \log(2/\delta)^2/\epsilon^2)$
    $N = (\hat{N} + \hat{N}^\top)/\sqrt{2}$
**end for**
**Runtime:**
**for** each communication round $t$ **do**
    Receive $S_t^i, s_t^i$ from agent, and insert it into $\mathcal{T}$ as a $d \times (d+1)$ matrix (Alg. 5, [69])
    Receive $M_t^i$ using the least nodes of $\mathcal{T}$ (Alg. 5, [69])
    $\hat{U}_t^i = U_t^i + H_t^i$, top-left $d \times d$ submatrix of $M_t^i$
    $\hat{u}_t^i = u_t^i + h_t^i$, last column of $M_t^i$
    Return $\hat{U}_t^i, \hat{u}_t^i$
**end for**
```

On a high level, the fairness regret proof considers a single hypothetical agent who plays mT rounds instead of considering m agents playing T rounds, each with sparse communication. The bounded deviation from this scenario to our intended setting is used to show the fairness regret analysis. Lemma 6.1 captures the fairness regret in terms of the determinant of the gram matrices, which is important to capture the deviation between the hypothetical agent and our intended set of agents, while lemma 6.2 is useful for fairness regret bounds for a single-agent. Lemma 6.3 formalizes the instantaneous fairness regret, a prerequisite for proving Theorem 6.1.

6.4.1 Regret Analysis

The following lemma ([22, Lemma 22]) is useful in proving the fairness regret of Fed-FairX-LinUCB.

Lemma 6.1

Elliptical Potential Let $x_1, \ldots, x_n \in R^d$ be vectors with each $\|x_t\| \leq L$. Given a positive definite matrix $U_1 \in R^{d\times d}$, define $U_{t+1} := U_t + x_t x_t^\top$ for all t. Then $\sum_{t=1}^n \min\left\{1, \|x_t\|^2_{U_t^{-1}}\right\} \leq 2\log\frac{\det U_{n+1}}{\det U_1} \leq 2d\log\frac{\operatorname{tr} U_1 + nL^2}{d\det^{1/d} U_1}$

Also, we extend Lemma A.6.4 from [21] to multi-agent setting as follows.

Lemma 6.2

When || $x_t^j(i)$ ||$_2 \le 1\ \forall a, t, j$, for the Fed-FairX-LinUCB algorithm, $\forall j \in [m]$, with probability $1 - \delta/2$,

$$\left|\sum_{t=1}^{T} w_t^j(i_t^i) - \sum_{t=1}^{T} \mathbb{E}_{i\sim\pi_t^j} w_t^j(i)\right| \le \sqrt{2T\ln(4/\delta)}$$

Here, $w_t^j(i) = \sqrt{x_t^j(i)(V_t^j)^{-1}(x_t^j(i))^\top}$ is the normalized width. With the help of the above lemmas, we now provide bounds on instantaneous regret FR_t^j.

Lemma 6.3

For the Fed-FairX-LinUCB, with high probability, the instantaneous regret for any agent j is bounded by,

$$FR_t^j = \sum_{i\in\mathcal{D}} |\pi_t^j - \pi_*^j| \le \frac{4L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i\sim\pi_t^j} \left\|x_t^j(i)\right\|_{(V_t^j)^{-1}}$$

Here, $w_t^j(i) = \sqrt{x_t^j(i)(V_t^j)^{-1}(x_t^j(i))^\top}$ is the normalized width.

Proof.

$$FR_t^j = \sum_{i\in\mathcal{D}} \left| \frac{f^j(\theta^* x_t^j(i))}{\sum_{i'\in\mathcal{D}} f^j(\theta^* x_t^j(i'))} - \frac{f^j(\theta_t^j x_t^j(i))}{\sum_{i'\in\mathcal{D}} f^j(\theta_t^j x_t^j(i'))} \right|$$

$$= \sum_i \left| \frac{f^j(\theta^* x_t^j(i)) \sum_{i'} f^j(\theta_t^j x_t^j(i')) - f^j(\theta_t^j x_t^j(i)) \sum_{i'} f^j(\theta^* x_t^j(i'))}{\sum_{i'} f^j(\theta_t^j x_t^j(i')) \sum_{i'} f^j(\theta^* x_t^j(i'))} \right|$$

$$= \sum_i \frac{\left| f^j(\theta^* x_t^j(i)) \sum_{i'} \left(f^j(\theta_t^j x_t^j(i')) - f^j(\theta^* x_t^j(i')) \right) + \left(f^j(\theta^* x_t^j(i)) - f^j(\theta_t^j x_t^j(i)) \right) \sum_{i'} f^j(\theta^* x_t^j(i')) \right|}{\sum_{i'} f^j(\theta_t^j x_t^j(i')) \sum_{i'} f^j(\theta^* x_t^j(i'))}$$

$$\leq \sum_i \frac{\begin{array}{r}\left| f^j(\theta^* x_t^j(i)) \sum_{i'} \left(f^j(\theta_t^j x_t^j(i')) - f^j(\theta^* x_t^j(i')) \right) \right| \\ + \left| \left(f^j(\theta^* x_t^j(i)) - f^j(\theta_t^j x_t^j(i)) \right) \sum_{i'} f^j(\theta^* x_t^j(i')) \right|\end{array}}{\sum_{i'} f^j(\theta_t^j x_t^j(i')) \sum_{i'} f^j(\theta^* x_t^j(i'))}$$

$$\leq \frac{2 \sum_i \left| f^j(\theta^* x_t^j(i)) - f^j(\theta_t^j x_t^j(i)) \right|}{\sum_{i'} f^j(\theta_t^j x_t^j(i'))}$$

$$= 2 \sum_i \frac{\pi_t^j}{f^j(\theta_t^j x_t^j(i))} \left| \begin{array}{r} f^j(\theta^* x_t^j(i)) - f^j(\hat{\theta}_t^j x_t^j(i)) \\ + f^j(\hat{\theta}_t^j x_t^j(i)) - f^j(\theta_t^j x_t^j(i)) \end{array} \right|$$

$$\begin{aligned} &\leq \frac{2L}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left[\begin{array}{r} \left\| \theta^* - \hat{\theta}_t^j \right\|_{V_t^j} \left\| x_t^j(i) \right\|_{(V_t^j)^{-1}} \\ + \left\| \hat{\theta}_t^j - \theta_t^j \right\|_{V_t^j} \left\| x_t^j(i) \right\|_{(V_t^j)^{-1}} \end{array} \right] \\ &\leq \frac{4L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V_t^j)^{-1}} \end{aligned}$$

□

The probability with which Lemma 6.3 holds true is dependent on β_t^j, where $\beta_t = max_{j \in M} \beta_t^j$. With the help of the above lemmas, we now provide bounds on instantaneous regret FR_t^j.

Theorem 6.1: Fairness Regret (Fed-FairX-LinUCB)

With high probability, Fed-FairX-LinUCB achieves the following fairness regret when $|| \ x_t^j(i) \ ||_2 \leq 1 \ \forall i, t, j$.

$$O\left(\frac{4\nu L\sqrt{\beta_t}}{\gamma} \sqrt{mTd \log{(1 + \frac{T}{d})} + m^2 d^3 \log^3{(1 + \frac{T}{d})}} \right)$$

Proof. Consider a hypothetical agent denoted by index 0 who plays in the following mT rounds - $(1,1), (1,2), \ldots (1,m), \ldots, (T,m)$ sequentially. Let the gram matrix for agent 0 till round (τ, n) be given by $V^0_{(\tau,n)} = mI + \sum_{j=1}^{j=m} \sum_{t=1}^{\tau-1} (x_t^j(i_t^j))(x_t^j(i_t^j))^T + \sum_{j=1}^{j=n} (x_\tau^j(i_\tau^j))(x_\tau^j(i_\tau^j))^T$. Substituting $U_1 = mI$

and $L = 1$ in Lemma 6.1 we get,

$$\sum_{j=1}^{j=m} \sum_{t=1}^{T} \left\| x_t^j(i_t^j) \right\|^2_{(V^0_{(t,j)})^{-1}} \leq 2d \log\left(1 + \frac{T}{d}\right)$$

Let the communication in the original algorithm occur at rounds $T_1, T_2, \ldots, T_{p-1}$. Let $\Psi_k = mI + \sum_{j=1}^{j=m} \sum_{t=1}^{T_k} (x_t^j(is_t^j))(x_t^j(is_t^j))^T$ be the synchronised gram matrix after communication round k. Then $\det \Psi_0 = (m)^d$ and $\det \Psi_p \leq \left(\frac{\mathrm{tr}(\Psi_p)}{d}\right)^d \leq (m + mT/d)^d$. Thus, for any $\nu > 1$, $\log_\nu \left(\frac{\det(\Psi_p)}{\det(\Psi_0)}\right) \leq d \log_\nu(1 + \frac{T}{d})$. Let event E represent the set of rounds when $1 \leq \frac{\det(\Psi_k)}{\det(\Psi_{k-1})} \leq \nu$ is true. Then, in all but $\lceil d \log_\nu(1 + \frac{T}{d}) \rceil$ rounds E is true.

For any $T_{k-1} \leq t \leq T_k$, when E is true,

$$\begin{aligned} fr_t^j &\leq \frac{4L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V_t^j)^{-1}} \\ &\leq \frac{4L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V^0_{(t,j)})^{-1}} \sqrt{\frac{\det V^0_{(t,j)}}{\det V_t^j}} \\ &\leq \frac{4L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V^0_{(t,j)})^{-1}} \sqrt{\frac{\det \Psi_k}{\det \Psi_{k-1}}} \\ &\leq \frac{4\nu L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V^0_{(t,j)})^{-1}} \end{aligned}$$

Here, second last equation follows because $V_t^j \geq \Psi_{k-1}$ and $\Psi_k \geq V^0_{(t,j)}$. Now, using Lemma 1 (main text),

$$\begin{aligned} \sum_{j=1}^{m} \sum_{t \in E} fr_t^j &\leq \sum_{j=1}^{m} \sum_{t \in E} \frac{4\nu L\sqrt{\beta_t}}{\gamma} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V^0_{(t,j)})^{-1}} \\ &\leq \frac{4\nu L\sqrt{\beta_T}}{\gamma} \left(\sum_{j=1}^{m} \sum_{t=1}^{T} \left\| x_t^j(i_t^j) \right\|_{(V^0_{(t,j)})^{-1}} + \sqrt{2mT \log(4/\delta)} \right) \\ &\leq \frac{4\nu L\sqrt{mT\beta_T}}{\gamma} \left(\sqrt{d \log(1 + \frac{T}{d})} + \sqrt{2 \log(4/\delta)} \right) \end{aligned}$$

Now, let us consider any period $t \in [T_{k-1}, T_k]$, where E does not hold and $t_k = T_k - T_{k-1}$ represent the length of the interval. Fairness regret during this period is given by,

$$FR([T_{k-1}, T_k]) \leq \frac{4L\sqrt{\beta_T}}{\gamma} \sum_{j=1}^{m} \sum_{t=T_{K-1}}^{T_k} \mathbb{E}_{i \sim \pi_t^j} \left\| x_t^j(i) \right\|_{(V_t^j)^{-1}}$$

$$\leq \frac{4L\sqrt{\beta_T}}{\gamma}\left(\sum_{j=1}^{m}\sum_{t=T_{K-1}}^{T_k}\left\|x_t^j(i_t^j)\right\|_{(V_t^j)^{-1}} + m\sqrt{2t_k\log(4/\delta)}\right) \qquad \text{(Using Lemma 6.1)}$$

$$\leq \frac{4L\sqrt{\beta_T}}{\gamma}\left(\sum_{j=1}^{m}\sqrt{t_k\log_\nu\frac{\det V^j_{T_{k-1}+t_k}}{\det V^j_{T_{k-1}}}} + m\sqrt{2t_k\log(4/\delta)}\right)$$

We know that ∀ agents, $t_k \leq \frac{T}{md^2\log^2(1+T/d)} + 1$ (otherwise there be a communication round), thus $FR([T_{k-1}, T_k]) \leq \frac{4L\sqrt{\beta_T}}{\gamma}$
$\left(\sqrt{\frac{m(T+md^2\log^2(1+\frac{T}{d}))}{d\log(1+\frac{T}{d})}} + \sqrt{\frac{2m(T+md^2\log^2(1+\frac{T}{d}))}{d^2\log^2(1+\frac{T}{d})}\log(\frac{4}{\delta})}\right)$.
Using the fact that E does not hold true in at most in $\lceil d\log_\nu(1+\frac{T}{d})\rceil$ rounds, we get

$$FR(T) \leq O\left(\frac{4\nu L\sqrt{\beta_T}}{\gamma}\sqrt{mTd\log(1+T/d) + m^2d^3\log^3(1+T/d)}\right)$$

□

The values in sequence of β_t dictates the probability with which Lemma 6.3, and in turn Theorem 6.1 holds. The problem of selection of values in sequence of β_t is well studied in the literature. For instance, using Theorem 2 from [75], it can be said that θ^* lies in the confidence region with probability $1-\alpha$ for $\beta_t = O\left(d\log(\frac{1+mt}{\alpha})\right)$ resulting in a regret bounds of $\tilde{O}\left(d\sqrt{mT\log^2(1+mT/d)}\right)$ for Fed-FairX-LinUCB (typically $m << T$ and hence the $\sqrt{mT}$ term dominates $\sqrt{m^2\log^2(1+T/d)}$).

The key difference between private and non-private regret analysis lies in the gram matrix regularization and confidence interval construction (use of appropriate β_t).

We note the following claim (Similar to [23, Proposition 2]) is useful for completing Priv-FairX-LinUCB's regret analysis. It provides values for the sequence of β_t for which the confidence interval contains θ^* with high probability.

Lemma 6.4

For an instance of problem where synchronisation occurs exactly n times in a span of T trials, and $\underline{\rho}, \bar{\rho}$ and z are $(\alpha/2nm)$-accurate [23, Definition 3]. Then for Priv-FairX-LinUCB with bounded target parameter ($||\ \theta^*\ ||_2 \leq c$), the sequence of $\sqrt{\beta_t^j}$ is (α, M, T)-accurate if, $\sqrt{\beta_t^j} = \sigma\sqrt{2\log(\frac{2}{\alpha}) + d\log(\frac{\bar{\rho}}{\underline{\rho}} + \frac{t}{d\underline{\rho}})} + mc\sqrt{\bar{\rho}} + mz$

Theorem 6.2: Fairness Regret (Priv-FairX-LinUCB)

With high probability, when $\| \ x_t^j(i) \ \|_2 \leq \ 1 \ \forall i, t, j$ and Lemma **??** holds, Priv-FairX-LinUCB achieves a fairness regret of
$O\left(\frac{4\nu L\sqrt{\beta_T}}{\gamma}\sqrt{mTd\log\left(\frac{\bar{\rho}}{\underline{\rho}}+\frac{T}{d\underline{\rho}}\right)+m^2d^3\log^3\left(\frac{\bar{\rho}}{\underline{\rho}}+\frac{T}{d\underline{\rho}}\right)}\right)$.

Proof. We note that the proof follows from the proof of Theorem 6.1 with minor changes. The regularisation of Ψ_k is done using $m\underline{\rho}I$ instead of mI. This allows for a tight bound on $\log_\nu\left(\frac{\det(\Psi_p)}{\det(\Psi_0)}\right)$ with appropriate values of $\bar{\rho}$ and $\underline{\rho}$. In addition, the property $V_t^j \geq G_t^j + M\underline{\rho}I$, is important for stating that $fr_t^j \leq \frac{4\nu L\sqrt{\beta_t}}{\gamma}\mathbb{E}_{a\sim\pi_t^j}\left\|x_t^j(i)\right\|_{(V^0_{(t,j)})^{-1}}$ when E holds true. The rest of the proof follows similar to the proof of Theorem 6.1. □

6.4.2 Privacy Guarantees

As mentioned in Section **??**, we can leverage the privatizer routines to provide differential privacy guarantees for Priv-FairX-LinUCB. At each synchronization, new observations, S_t^i and s_t^i, are added to a leaf node, while all other nodes store the sum of the child nodes. Thus, $1 + \lceil \log(n) \rceil$ nodes of the tree, where n is the total number of communication rounds, are sufficient to represent any partial sum till the last synchronization round. Since the privatizer routine follows the routine introduced by earlier works, it trivially follows that if each node guarantees $(\epsilon/\sqrt{8m\ln(2/\delta)}, \delta/2m)-$privacy, the outgoing communication is guaranteed to be $(\epsilon, \delta, m)-$federated differential private for each synchronization with similar values for $\bar{\rho}, \underline{\rho}, z$.

Claim 6.1

(Follows from [23, Remark 3]) The privatizer routine in Priv-FairX-LinUCB guarantees that each of the outgoing messages for an agent i is $(\epsilon, \delta)-$differentially private.

6.5 Experimental Analysis

6.5.1 Experimental Set-up

Dataset Synthetic datasets were generated for all experiments by randomly fixing the model parameter θ^*. Context size was set to five ($d = 5$), and feature vectors $\mathcal{X}_t^j$ were sampled from a uniform distribution, $x_t^j(i) \in [0, 1]^d$. Noise $\eta_t^j(i)$, sampled from a normal distribution centered at 0, was added to produce reward observations.

Merit Function and Optimization A steep merit function,$f(\cdot) = e^{10\mu}$, was employed, similar to [21]. Projected gradient descent was used in each round to solve the resulting non-convex optimization problem.

6.5.2 Evaluation Set-up

Evaluation Metric Fairness regret was used as the primary evaluation metric to assess the algorithms' ability to balance performance and fairness. Exp 1 and 2 shows fairness regret trends with respect to rounds while Exp 3 and 4 uses the fairness regret at $t = 100,000$. The objective is to minimise fairness regret, and thus it is being used as the evaluation metric in the experiments. (Though our focus is on fairness, for completeness, we also evaluate the proposed algorithms for reward regret [21] in Appendix.)

Experiment Repetition All reported results were averaged over 5 runs to ensure statistical significance.

6.5.3 Baselines

Since a novel setting is proposed, there are no algorithms for direct comparisons. 2 different kinds of baselines are used to demonstrate the efficacy of our proposed algorithm.

Single-Agent Baseline ($B0$) FairX-LinUCB algorithm was employed as a single-agent baseline to facilitate comparison with federated learning approaches. We note it as $B0$ in our experiments. Each agent essentially learns on their own do not communicate with other agents under this baseline.

Communication Protocol Baseline ($B1$, $B2$) Two existing communication protocols from [23] and [53] were compared against the proposed protocol to evaluate its efficacy. These have been termed $B1$ and $B2$ respectively. Note that the algorithms proposed in [23] and [53] optimize for traditional regret, hence Priv-FairX-LinUCB has been modified to just use their proposed communication protocols to form $B1$ and $B2$.

6.5.4 Experiments

Exp 1: Single-Agent vs Federated Learning

Compares the fairness regret of baseline $B0$ to the proposed non-private algorithm, Fed-FairX-LinUCB and its differentially private counterpart, Priv-FairX-LinUCB, for 10 agents (m). [$\epsilon = 2$, $\delta = 0.1$, $t \in [1, 100000]$]

Exp 2: Communication Protocol

Assesses the performance Priv-FairX-LinUCB against $B1$ and $B2$ with 10 agents. [$\epsilon = 2$, $\delta = 0.1$, $t \in [1, 100000]$]

Exp 3: Dependence on m

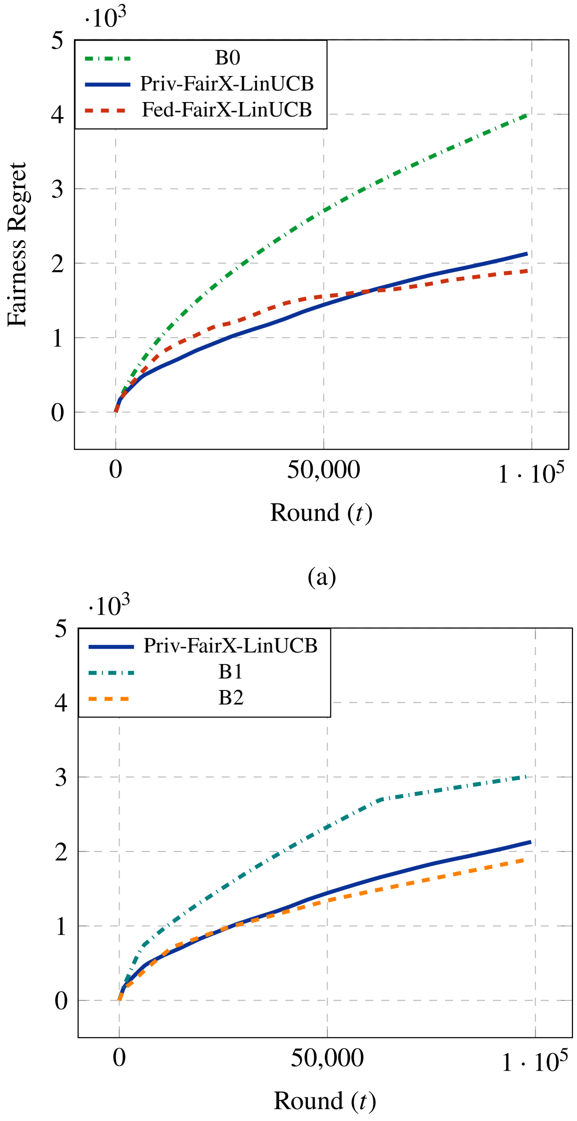
Compares the impact of the number of agents (m) on the fairness regret of both proposed algorithms. [$\epsilon = 2$, $\delta = 0.1$, $t = 100000$]]

Exp 4: Privacy Budget

Examines the effect of the privacy budget (ϵ) on the fairness regret of the private algorithm. [$m = 10$, $\delta = 0.1$, $t = 100000$]]

6.5.5 Inferences

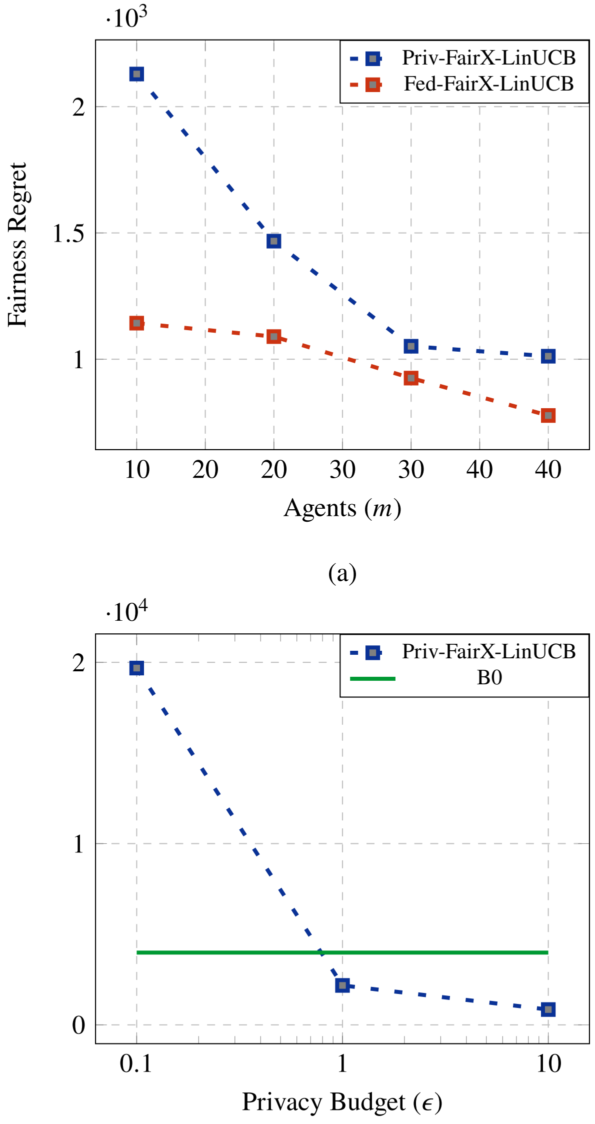
- Both federated learning algorithms outperformed the single-agent baseline in terms of fairness regret.
- Priv-FairX-LinUCB outperforms B1 while producing comparable performance for B2. But unlike B2, Priv-FairX-LinUCB has bounded communication gaps, which is necessary for the theoretical guarantees provided. In B2, communication gaps are as high as $O(T)$ in the later stages, and hence, in theory, fairness regrets could be as bad as $O(T)$ for B2.
- The fairness regret scales as expected with respect to the number of agents, validating theoretical results.
- The private algorithm achieved reasonable performance for ϵ values of 1 or greater, highlighting the trade-off between privacy and regret.



(a)

(b)

Figure 6.2: **(a)** Exp 1 : Fairness Regret vs. Rounds for single-agent baseline and proposed federated learning algorithms ($m = 10$) **(b)** Exp 2 : Fairness Regret vs. Rounds for different communication protocol baselines and proposed algorithms ($m = 10$)



(a)

(b)

Figure 6.3: **(a)** Exp 3 : Fairness Regret trend w.r.t. number of agents ($t = 100,000$) **(b)** Exp 4 : Fairness Regret trend w.r.t. privacy budget ($t = 100,000$)

6.6 Conclusion

This work looked at the federated contextual bandit problem with fairness optimization objective while ensuring privacy of the agents. In order to extend the FairX-LinUCB algorithm to a federated setting, we proposed a novel communication protocol in Algorithm 6.1. Through rigorous theoretical analysis, we proved that Fed-FairX-LinUCB, the non-private algorithm achieves sub-linear fairness regret (compared to linear regret in a non-federated setting) with respect to the number of agents, i.e., for the m agent setting, the total fairness regret is of the order $O(\sqrt{m})$ (Theorem 6.1). Additionally, we presented Priv-FairX-LinUCB, which ensured differential privacy guarantees for the agents. We show that Priv-FairX-LinUCB has bounded fairness regret (Theorem 6.2). We empirically validated our results, highlighting that Priv-FairX-LinUCB significantly improves over non-collaborative learning while maintaining privacy.

6.7 Notational Summary

For ease of reference, Table 5.2 lists important notations used in this chapter.

Symbol	**Description**
d	Context Length
$\mathcal{D}$	Set of arms
T	Number of rounds
m	Number of learning agents
$\mathcal{X}_t^j = (x_t^j(i))_{i\in\mathcal{D}}$	Set of context vectors observed in round t by agent j
i_t	Arm chosen at round t
θ	Learnable parameter
$\theta^\star$	True value learnable parameter
$y_t^j(i)$	Reward observed on choosing context i
$\pi_t^j(\mathcal{X})$	Policy denoting the vector of selection probabilities
f^j	Merit score function
FR	Fairness Regret
V_t^j	Gram matrix
b_t^j	Reward sequence vector
H_t^j	Gram matrix perturbation
h_t^j	Reward sequence vector perturbation

Table 6.1: Exposure of Fairness in Multi-agent Contextual Bandits with Privacy Guarantees: Important Notations

Chapter 7

Conclusion and Future Work

This thesis has explored the delicate balance between exploration and exploitation in decision-making processes, with a particular focus on optimization under privacy constraints in federated learning environments. Our investigation delved into both theoretical and practical aspects, addressing key challenges and proposing innovative solutions to improve decision-making while ensuring privacy and fairness. The primary contributions of this thesis are encapsulated in the novel algorithms and theoretical frameworks introduced in Chapters 5 and 6.

In Chapter 5, we tackled the challenge of optimizing federated combinatorial multi-armed bandits (CMAB) while maintaining differential privacy. The contributions in this chapter are multifaceted:

- *Theoretical Analysis of Regret Improvement:* We provided a rigorous theoretical analysis demonstrating that federated learning can significantly improve regret compared to individual learning in a non-private homogeneous setting. Theorem 5.1 shows the potential gains in terms of regret reduction, highlighting the benefits of collaborative learning.

- *Challenges with Naive Privacy Techniques:* We identified and demonstrated that employing privacy techniques in a naive manner is not only ineffective but also poses significant risks of information leakage. Claim 5.1 in Section 5.4.2 underscores the pitfalls of simplistic privacy approaches, which fail to protect sensitive information adequately.

- *Introduction of P-FCB:* To address these challenges, we introduced P-FCB, a sophisticated algorithm designed to implement privacy techniques practically. Algorithm 5.3 incorporates selective perturbation of information and well-defined communication rounds to ensure strong privacy guarantees. By learning selectively and using error bounds around current estimates, P-FCB minimizes regret while protecting privacy.

- *Empirical Validation:* The practical efficacy of P-FCB was empirically validated through extensive simulations. Our results, detailed in Section 5.5, demonstrate that P-FCB significantly improves per-agent regret in private settings compared to individual learning. This empirical evidence supports the theoretical claims and showcases the potential of P-FCB in real-world applications.

In Chapter 6, we shifted our focus to the integration of fairness into federated contextual bandit problems while maintaining privacy guarantees. The contributions in this chapter are as follows:

- *Introduction of Fairness Notion:* We introduced a novel notion of fairness for actions in federated contextual bandits. This concept ensures that each action is given equitable consideration across different learning agents, addressing potential biases in the decision-making process and combating the problem of action starvation.
- *Novel Communication Protocol:* We proposed a novel communication protocol that achieves sublinear fairness regret concerning the number of learning agents. Theorem 6.1 demonstrates that our protocol is optimal in terms of the number of rounds, up to a logarithmic factor. This result implies that fairness regret scales linearly in a non-collaborative setting, highlighting the benefits of our collaborative approach.
- *Extensible Communication Protocol:* Our communication protocol is extensible to the privatizer routine from previous work, allowing the development of Priv-FairX-LinUCB. This algorithm ensures differential privacy guarantees for the agents, integrating privacy and fairness seamlessly.
- *Theoretical Guarantees for Priv-FairX-LinUCB:* We provided theoretical evidence showing that Priv-FairX-LinUCBachieves differential privacy guarantees while maintaining bounded fairness regret, as demonstrated in Theorem 6.2.
- *Empirical Performance:* Through extensive empirical evaluations, we showed that both Fed-FairX-LinUCBand Priv-FairX-LinUCBoutperform non-collaborative learners. These results underscore the practical viability and effectiveness of our proposed algorithms in achieving fair and private learning outcomes.

While this thesis has made significant strides in addressing optimization under privacy and fairness constraints in federated learning, several avenues for future research remain open.

- *Extension to Heterogeneous Settings:* One promising direction is to extend the theoretical and empirical analysis to heterogeneous federated learning environments. This would involve considering scenarios where learning agents have diverse data distributions and capabilities, which is more representative of real-world applications.
- *Enhanced Fairness Metrics:* Developing more comprehensive and nuanced metrics for fairness in federated learning is another crucial area. These metrics should capture various dimensions of fairness, including long-term impacts and intersectional fairness, to ensure equitable outcomes for all participants.
- *Scalability and Efficiency:* Further research is needed to improve the scalability and computational efficiency of the proposed algorithms. As federated learning systems grow in size and complexity, optimizing resource usage and ensuring efficient communication become critical challenges.

- *Real-world Deployments and Case Studies:* Conducting real-world deployments and case studies to validate the theoretical and empirical findings of this thesis would provide valuable insights and feedback. Such studies could help identify practical challenges and refine the proposed algorithms for broader adoption.

In conclusion, this thesis has laid a robust foundation for optimizing decision-making under privacy and fairness constraints in federated learning environments. The contributions made in Chapters 5 and 6 provide valuable insights and tools for advancing the state of the art. Future work in the aforementioned areas will further enhance our understanding and capabilities, driving the development of more effective, fair, and privacy-preserving learning systems.

Bibliography

[1] Arthur J. Moss, Charles W. Francis, and Daniel Ryan. Collaborative clinical trials. *New England Journal of Medicine*, 2011.

[2] A collaborative learning environment based on intelligent agents. *Expert Systems with Applications*, 14, 1998. Artificial Intelligence in Mexico.

[3] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 1985.

[4] Akash Das Sharma, Sujit Gujar, and Y Narahari. Truthful multi-armed bandit mechanisms for multi-slot sponsored search auctions. *Current Science*, 2012.

[5] 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*, 2017.

[6] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[7] Sanjay Chandlekar, Easwar Subramanian, and Sujit Gujar. Multi-armed bandit based tariff generation strategy for multi-agent smart grid systems. In *Engineering Multi-Agent Systems*, 2023.

[8] Sanjay Chandlekar, Arthik Boroju, Shweta Jain, and Sujit Gujar. A novel demand response model and method for peak reduction in smart grids–powertac. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[9] Akansha Singh, M. Vamshidhar Reddy, Zoltán Nagy, Sujit Gujar, and Shweta Jain. Designing bounded min-knapsack bandits algorithm for sustainable demand response. In *Pacific Rim International Conference on Artificial Intelligence*, 2021.

[10] Kumar Abhishek, Shweta Jain, and Sujit Gujar. Designing truthful contextual multi-armed bandits based sponsored search auctions. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020.

[11] Shweta Jain and Sujit Gujar. A multiarmed bandit based incentive mechanism for a subset selection of customers for demand response in smart grids. In *AAAI Conference on Artificial Intelligence*, 2020.

[12] Ganesh Ghalme, Swapnil Dhamal, Shweta Jain, Sujit Gujar, and Y Narahari. Ballooning multi-armed bandits. *Artificial Intelligence*, 296, 2021.

[13] Debojit Das, Shweta Jain, and Sujit Gujar. Budgeted combinatorial multi-armed bandits. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 2022.

[14] Padala Manisha and Sujit Gujar. Thompson sampling based multi-armed-bandit mechanism using neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

[15] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, 2013.

[16] Ayush Deva, Kumar Abhishek, and Sujit Gujar. A multi-arm bandit approach to subset selection under constraints. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[17] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *international conference on World wide web*, 2010.

[18] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

[19] Sankarshan Damle, Aleksei Triastcyn, Boi Faltings, and Sujit Gujar. Differentially private multi-agent constraint optimization. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2022.

[20] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Y Narahari. Achieving fairness in the stochastic multi-armed bandit problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[21] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. Fairness of exposure in stochastic bandits. In *International Conference on Machine Learning*, 2021.

[22] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. *Advances in Neural Information Processing Systems*, 2018.

[23] Abhimanyu Dubey and AlexSandy' Pentland. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 2020.

[24] Foundry model — Wikipedia, the free encyclopedia, 2022. URL `https://en.wikipedia.org/w/index.php?title=Foundry_model&oldid=1080269386`.

[25] Original equipment manufacturer — Wikipedia, the free encyclopedia, 2022. URL `https://en.wikipedia.org/w/index.php?title=Original_equipment_manufacturer&oldid=1080228401`.

[26] Abhimanyu Dubey and AlexSandy' Pentland. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[27] Chengshuai Shi and Cong Shen. Federated multi-armed bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35, May 2021.

[28] 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*, 2017.

[29] Zhaohua Zheng, Yize Zhou, Yilong Sun, Zhang Wang, Boyi Liu, and Keqiu Li. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. *Connection Science*, 2021.

[30] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning*, 2013.

[31] Shweta Jain, Sujit Gujar, Satyanath Bhat, Onno Zoeter, and Yadati Narahari. A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing. *Artificial Intelligence*, 254, 2018.

[32] Satyanath Bhat, Shweta Jain, Sujit Gujar, and Yadati Narahari. An optimal bidimensional multi-armed bandit auction for multi-unit procurement. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015.

[33] Shweta Jain, Sujit Gujar, Satyanath Bhat, and Onno Zoeter. A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing. *Artificial Intelligence*, 254, 2018.

[34] Taehyeon Kim, Sangmin Bae, Jin-Woo Lee, and Seyoung Yun. Accurate and fast federated learning via combinatorial multi-armed bandits. *CoRR*, 2020.

[35] Chengshuai Shi, Cong Shen, and Jing Yang. Federated multi-armed bandits with personalization. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

[36] Dhanvin Mehta and Devesh Yamparala. Policy gradient reinforcement learning for solving supply-chain management problems. In *Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014*, 2014.

[37] Nicollas Silva, Heitor Werneck, Thiago Silva, Adriano CM Pereira, and Leonardo Rocha. Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. 197, 2022.

[38] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, 2006.

[39] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 2014.

[40] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47, 2004.

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

[42] Jain Shweta and Gujar Sujit. A multiarmed bandit based incentive mechanism for a subset selection of customers for demand response in smart grids. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, 2020.

[43] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[44] Mohammad Malekzadeh, Dimitrios Athanasakis, Hamed Haddadi, and Ben Livshits. Privacy-preserving bandits. In *Proceedings of Machine Learning and Systems*, volume 2, 2020.

[45] Hui Zhao, Mingjun Xiao, Jie Wu, Yun Xu, He Huang, and Sheng Zhang. Differentially private unknown worker recruitment for mobile crowdsensing using multi-armed bandits. *IEEE Transactions on Mobile Computing*, 2021.

[46] Shuzhen Chen, Youming Tao, Dongxiao Yu, Feng Li, Bei Gong, and Xiuzhen Cheng. Privacy-preserving collaborative learning for multiarmed bandits in iot. *IEEE Internet of Things Journal*, 8, 2021.

[47] Tan Li and Linqi Song. Privacy-preserving communication-efficient federated multi-armed bandits. *IEEE Journal on Selected Areas in Communications*, 40, 2022.

[48] Awni Y. Hannun, Brian Knott, Shubho Sengupta, and Laurens van der Maaten. Privacy-preserving contextual bandits. *CoRR*, 2019.

[49] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, 2019.

[50] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 2010.

[51] Jiafan He, Tianhao Wang, Yifei Min, and Quanquan Gu. A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. In *Advances in Neural Information Processing Systems*, 2022.

[52] Clémence Réda, Sattar Vakili, and Emilie Kaufmann. Near-optimal collaborative learning in bandits. *arXiv preprint arXiv:2206.00121*, 2022.

[53] Sambhav Solanki, Samhita Kanaparthy, Sankarshan Damle, and Sujit Gujar. Differentially private federated combinatorial bandits with constraints. *arXiv preprint arXiv:2206.13192*, 2022.

[54] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 2021.

[55] Kritika Prakash, Fiza Husain, Praveen Paruchuri, and Sujit Gujar. How private is your rl policy? an inverse rl based analysis framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022.

[56] Manisha Padala, Sankarshan Damle, and Sujit Gujar. Federated learning meets fairness and differential privacy. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part VI 28*, 2021.

[57] Subham Pokhriyal, Shweta Jain, Ganesh Ghalme, Swapnil Dhamal, and Sujit Gujar. Simultaneously achieving group exposure fairness and within-group meritocracy in stochastic bandits. *arXiv preprint arXiv:2402.05575*, 2024.

[58] Archit Sood, Shweta Jain, and Sujit Gujar. Fairness of exposure in online restless multi-armed bandits. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, Richland, SC, 2024.

[59] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, 2016.

[60] Robert J Aumann and Michael Maschler. Game theoretic analysis of a bankruptcy problem from the talmud. *Journal of Economic Theory*, 1985.

[61] Warut Suksompong. Asymptotic existence of proportionally fair allocations. *Mathematical Social Sciences*, 2016.

[62] Safwan Hossain, Evi Micha, and Nisarg Shah. Fair algorithms for multi-agent multi-armed bandits. *Advances in Neural Information Processing Systems*, 2021.

[63] Arpita Biswas, Jackson A Killian, Paula Rodriguez Diaz, Susobhan Ghosh, and Milind Tambe. Fairness for workers who pull the arms: An index based policy for allocation of restless bandit tasks. *arXiv preprint arXiv:2303.00799*, 2023.

[64] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*, 2006.

[65] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 2011.

[66] 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*, 2010.

[67] Aristide CY Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[68] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, 2013.

[69] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, 2012.

[70] Xutong Liu, Haoru Zhao, Tong Yu, Shuai Li, and John CS Lui. Federated online clustering of bandits. In *Uncertainty in Artificial Intelligence*, 2022.

[71] Yifang Chen, Alex Cuellar, Haipeng Luo, Jignesh Modi, Heramb Nemlekar, and Stefanos Nikolaidis. Fair contextual multi-armed bandits: Theory and experiments. In *Conference on Uncertainty in Artificial Intelligence*, 2020.

[72] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[73] Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Fair algorithms for infinite and contextual bandits. *arXiv preprint arXiv:1610.09559*, 2016.

[74] Shaarad A. R and Ambedkar Dukkipati. A regret bound for non-stationary multi-armed bandits with fairness constraints. *CoRR*, 2020.

[75] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.