A Simple and Provably Efficient Algorithm for Asynchronous Federated Contextual Linear Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study federated contextual linear bandits, where M agents cooperate with each other to solve a global contextual linear bandit problem with the help of a central server. We consider the asynchronous setting, where all agents work independently and the communication between one agent and the server will not trigger other agents' communication. We propose a simple algorithm named `FedLinUCB` based on the principle of optimism. We prove that the regret of `FedLinUCB` is bounded by $\widetilde{O}(d\sqrt{\sum_{m=1}^{M} T_m})$ and the communication complexity is $\widetilde{O}(dM^2)$, where d is the dimension of the contextual vector and T_m is the total number of interactions with the environment by m-th agent. To the best of our knowledge, this is the first provably efficient algorithm that allows fully asynchronous communication for federated contextual linear bandits, while achieving the same regret guarantee as in the single-agent setting.

1 Introduction

Contextual linear bandit is a canonical model in sequential decision making with partial information feedback that has found vast applications in real-world domains such as recommendation systems (Li et al., 2010a,b; Gentile et al., 2014; Li et al., 2020), clinical trials (Wang, 1991; Durand et al., 2018) and economics (Jagadeesan et al., 2021; Li et al., 2022). Most existing works on contextual linear bandits focus on either the single-agent setting (Auer, 2002; Abe et al., 2003; Dani et al., 2008; Li et al., 2010a; Rusmevichientong and Tsitsiklis, 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2013) or multi-agent settings where communications between agents are instant and unrestricted (Cesa-Bianchi et al., 2013; Li et al., 2016; Wu et al., 2016; Li et al., 2021). Due to the increasing amount of data being distributed across a large number of local agents (e.g., clients, users, edge devices), federated learning (McMahan et al., 2017; Karimireddy et al., 2020) has become an emerging paradigm for distributed machine learning, where agents can jointly learn a global model without sharing their own localized data. This motivates the development of distributed/federated linear bandits (Wang et al., 2019; Huang et al., 2021; Li and Wang, 2022a), which enables a collection of agents to cooperate with each other to solve a global linear bandit problem while enjoying performance guarantees comparable to those in the classical single-agent centralized setting.

However, most existing federated linear bandits algorithms are limited to the synchronous setting (Wang et al., 2019; Dubey and Pentland, 2020; Huang et al., 2021), where all the agents have to first upload their local data to the server upon the request of the server, and the agents will download the latest data from the server after all uploads are complete. This requires full participation of the agents and global synchronization mandated by the server, which is impractical in many real-world application scenarios. The only notable exception is Li and Wang (2022a), where an asynchronous federated linear bandit algorithm is proposed. Nevertheless, in their algorithm, the upload by one

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

agent may trigger the download from the server to all other agents. Therefore, the communications between different agents and the server are not totally independent. Moreover, they make a stringent regularity assumption on the contexts, which basically requires the contexts to be stochastic rather than adversarial as in standard contextual linear bandits. That said, their theoretical proof is actually flawed as they ignored some unique challenges caused by asynchronous communication (see Appendix A.1 for details). Therefore, how to design a truly asynchronous contextual linear bandit algorithm remains an open problem.

In this work, we resolve the above open problem by proposing a simple algorithm for asynchronous federated contextual linear bandits over a star-shaped communication network. Our algorithm is based on the principle of optimism (Abbasi-Yadkori et al., 2011) and enjoys the following advantages: (i) Each agent can decide whether or not to participate in each round. Full participation is not required, thus it allows temporarily offline agents. This is much more flexible than existing algorithms for federated linear bandits in Wang et al. (2019); Dubey and Pentland (2020); Huang et al. (2021) where all agents are required to participate in each round; and (ii) the communication between each agent and the server is asynchronous and totally independent of other agents. There is no need of global synchronization or mandatory coordination by the server, in contrast to Li and Wang (2022a) where each agent might be asked by the server to download data. In particular, the communication between the agent and the server is triggered by a matrix determinant-based criterion that can be computed independently by each agent. Our algorithm design not only allows the agents to independently operate and synchronize with the server, but also ensures low communication complexity (i.e., total number of rounds of communication between agents and the server) and low switching cost (i.e., total number of local model updates for all agents) (Abbasi-Yadkori et al., 2011).

While being simple, our algorithm design introduces a challenge in the regret analysis. Since the order of the interaction between the agent and the environment is not fixed, standard martingale-based concentration inequality cannot be directly applied. Specifically, this challenge arises due to the mismatch between the partial data information collected by the central server and the true order of the data generated from the interaction with the environment, as is explained in detail in Section 5 and illustrated by Figure 1. We address this challenge by a novel proof technique, which first establishes the local concentration of each agent's data and then relates it to the "virtual" global concentration of all data via the determinant-based criterion. Based on this proof technique, we are able to obtain tight enough confidence bounds that leads to a nearly optimal regret. Moreover, our theoretical analysis relies only on minimal assumptions that are standard for contextual linear bandits, relaxing the strong assumptions made in Li and Wang (2022a).

Main contributions. Our contributions are highlighted as follows:

- We devise a simple algorithm named `FedLinUCB` that achieves near-optimal regret, low communication complexity and low switching cost simultaneously for asynchronous federated contextual linear bandits. In detail, we prove that our algorithms achieves a near-optimal $\widetilde{O}(d\sqrt{T})$ regret with merely $\widetilde{O}(dM^2)$ total communication complexity and $\widetilde{O}(dM^2)$ total switching cost. Here M is the number of agents, d is the dimension of the context and $T = \sum_{m=1}^{M} T_m$ is the total number of rounds with T_m being the number of rounds that agent m participates in. When degenerated to single-agent bandits, the regret of our algorithm matches the optimal regret $\widetilde{O}(d\sqrt{T})$ (Abbasi-Yadkori et al., 2011).
- We also prove an $\Omega(M/\log(T/M))$ lower bound for the communication complexity. Together with the $O(dM^2)$ upper bound of our algorithm, it suggests that there is only an $\widetilde{O}(dM)$ gap between the upper and lower bounds of the communication complexity.
- We identify the issue of ill-defined filtration caused by the unfixed order of interactions between agents and the environment, which is absent in previous synchronous or single-agent settings. We tackle this unique challenge by connecting the local concentration of each local agent and the global concentration of the aggregated data from all agents. We believe this proof technique is of independent interest for the analysis of other asynchronous bandit problems.

Notation. For any positive integer n, we denote the set $\{1, 2, \ldots, n\}$ by $[n]$. We use $\mathbf{I}$ to denote the $d \times d$ identity matrix. We use $\mathcal{O}$ to hide universal constants and $\widetilde{\mathcal{O}}$ to further hide poly-logarithmic terms. For any vector $\mathbf{x} \in \mathbb{R}^d$ and positive semi-definite $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, we denote $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}}$.

Setting	Algorithm	Regret	Communication	Low-switching	No extra assump. on contexts	Allow free participation
Single-agent	OFUL (Abbasi-Yadkori et al., 2011)	$d\sqrt{T}\log T$	N/A	✓	✓	✘
Federated (Sync.)	DisLinUCB (Wang et al., 2019)	$d\sqrt{T}\log^2 T$	$dM^{3/2}$	✘	✓	✘
Federated (Async.)	Async-LinUCB (Li and Wang, 2022a)	$d\sqrt{T}\log T$	$dM^2\log T$	✘	✘	✘
Federated (Async.)	FedLinUCB (Our Algorithm 1)	$d\sqrt{T}\log T$	$dM^2\log T$	✓	✓	✓

Table 1: Comparison of our result with baseline approaches for contextual linear bandits. Our result achieves near-optimal regret under low communication complexity. Here d is the dimension of the context, M is the number of agents, and $T = \sum_{m=1}^{M} T_m$ with each T_m being the number of rounds that agent m participates in.

2 Related Work

We review related work on distributed/federated bandit algorithms based on the type of bandits: (1) multi-armed bandits, (2) stochastic linear bandits and (3) contextual linear bandits.

Distributed/federated multi-armed bandits. There is a vast literature on distributed/federated multi-armed bandits (MABs) (Liu and Zhao, 2010; Szorenyi et al., 2013; Landgren et al., 2016; Chakraborty et al., 2017; Landgren et al., 2018; Martínez-Rubio et al., 2019; Sankararaman et al., 2019; Wang et al., 2019, 2020; Zhu et al., 2021), to mention a few. However, none of these algorithms can be directly applied to linear bandits, needless to say contextual linear bandits with infinite decision sets.

Distributed/federated stochastic linear bandits. In distributed/federated stochastic linear bandits, the decision set is fixed across all the rounds $t \in [T]$ and all the agents $m \in [M]$. Wang et al. (2019) proposed the DELD algorithm for distributed stochastic linear bandits on both star-shaped network and P2P network. Huang et al. (2021) proposed an arm elimination-based algorithm called Fed-PE for federated stochastic linear bandits on the star-shaped network. Both algorithms are in the synchronous setting and require full participation of the agents upon the server's request.

Distributed/federated contextual linear bandits. The contextual linear bandit is more general and challenging than stochastic linear bandits, because the decision sets can vary for each t and m. In this setting, Korda et al. (2016) considered a P2P network and proposed the DCB algorithm based on the OFUL algorithm in Abbasi-Yadkori et al. (2011). Wang et al. (2019) considered both star-shaped and P2P communication networks and achieved the near-optimal $\widetilde{O}(d\sqrt{T})$ regret in the synchronous setting.[1] Dubey and Pentland (2020) further introduced the differential privacy guarantee into the setting of Wang et al. (2019). Li and Wang (2022b) extended distributed contextual linear bandits to generalized linear bandits (Filippi et al., 2010; Jun et al., 2017) in the synchronous setting. Li and Wang (2022a) proposed the first asynchronous algorithm for federated contextual linear bandits with the star-shaped graph and achieve near-optimal $\widetilde{O}(d\sqrt{T})$ regret. However, their setting is different from ours in two aspects: (1) the upload triggered by an agent will lead the server to trigger download possibly for all the agents in their setting. In contrast, the upload triggered by an agent will only lead to download to the same agent in our setting; (2) their regret guarantee relies on a stringent regularity assumption on the contexts, which basically requires the contexts to be stochastic. As a comparison, the contexts in our setting can be even adversarial, which is exactly the setting of contextual linear bandits (Abbasi-Yadkori et al., 2011; Li et al., 2019). This difference in the setting makes our algorithm a truly asynchronous contextual linear bandit algorithm but also makes our regret analysis more challenging.

For better comparison, we compare our work with the most related contextual linear bandit algorithms in Table 1.

[1] In the original paper of Wang et al. (2019), the regret bound is expressed as $\widetilde{O}(d\sqrt{MT})$. The T in their paper is equivalent to the T_m in ours, so their $d\sqrt{MT}$ should be understood as $d\sqrt{T}$ under our notation.

3 Preliminaries

Federated contextual linear bandits. We consider the federated contextual linear bandits as follows: At each round $t \in [T]$, an arbitrary agent $m_t \in [M]$ is active for participation. This agent receives a decision set $D_t \subset \mathbb{R}^d$, picks an action $\mathbf{x}_t \in D_t$, and receives a random reward r_t. We assume that the reward r_t satisfies $r_t = \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle + \eta_t$ for all $t \in [T]$, where η_t is conditionally independent of $\mathbf{x}_t$ given $\mathbf{x}_{1:t-1}, m_{1:t}, r_{1:t-1}$. More specifically, we make the following assumption on η_t, $\boldsymbol{\theta}^*$ and D_t, which is a standard assumption in the contextual linear bandit literature (Abbasi-Yadkori et al., 2011; Wang et al., 2019; Dubey and Pentland, 2020).

Assumption 3.1. The noise η_t is R-sub-Gaussian conditioning on $\mathbf{x}_{1:t}$, $m_{1:t}$ and $r_{1:t-1}$, i.e.,

$$\mathbb{E}\big[e^{\lambda \eta_t} \mid \mathbf{x}_{1:t}, m_{1:t}, r_{1:t-1}\big] \leq \exp(R^2\lambda^2/2), \quad \text{for any } \lambda \in \mathbb{R}.$$

We also assume that $\|\boldsymbol{\theta}^*\|_2 \leq S$ and $\|\mathbf{x}\|_2 \leq L$ for all action $\mathbf{x} \in \mathcal{D}_t$, for all $t \in [T]$.

Notably, we assume m_t can be arbitrary for all t, which basically says that each agent can decide whether and when to participate or not.[2] Our setting is more general than the synchronous setting in Wang et al. (2019); Dubey and Pentland (2020), which requires a round-robin participation of all agents.

Learning objective. The goal of the agents is to collaboratively minimize the cumulative regret defined as

$$\mathrm{Regret}(T) \coloneqq \sum_{t=1}^{T} \Big(\max_{\mathbf{x} \in D_t} \langle \mathbf{x}, \boldsymbol{\theta}^* \rangle - \langle \mathbf{x}_t, \boldsymbol{\theta}^* \rangle \Big) = \sum_{t=1}^{T} \langle \mathbf{x}_t^* - \mathbf{x}_t, \boldsymbol{\theta}^* \rangle. \tag{3.1}$$

To achieve such a goal, we allow the agents to collaborate via communication through the central server. Below we will explain the details of the communication model.

Communication model. We consider a star-shaped communication network (Wang et al., 2019; Dubey and Pentland, 2020) consisting of a central server and M agents, where each agent can communicate with the server by uploading and downloading data. However, any pair of agents cannot communicate with each other directly. We define the communication complexity as the total number of communication rounds between agents and the server (counting both the uploads and downloads) (Wang et al., 2019; Dubey and Pentland, 2020; Li and Wang, 2022a). For simplicity, we assume that there is no latency in the communication channel.

We consider the asynchronous setting, where the communication protocol satisfies: (1) each agent can decide whether or not to participate in each round. Full participation is not required, which allows temporarily offline agents; and (2) the communication between each agent and the server is asynchronous and independent of other agents without mandatory download by the server.

Switching cost. The notion of switching cost in online learning and bandits refers to the number of times the agent switches its policy (i.e., decision rule) (Kalai and Vempala, 2005; Abbasi-Yadkori et al., 2011; Dekel et al., 2014; Ruan et al., 2021). In the context of linear bandits, it corresponds to the number of times the agent updates its policy of selecting an action from the decision set (Abbasi-Yadkori et al., 2011). Algorithms with low switching cost are preferred in practice since each policy switching might cause additional computational overhead.

4 The Proposed Algorithm

We propose a simple algorithm based on the principle of optimism that enables collaboration among agents through asynchronous communications with the central server. The main algorithm is displayed in Algorithm 1. For clarity, we first summarize the related notations in Table 2.

Specifically, in each round $t \in [T]$, agent m_t participates and interacts with the environment (Line 3). The environment specifies the decision set D_t (Line 4), and the agent m_t selects the action based on its current optimisitic estimate of the reward (Line 5). Here the bonus term $\beta \|\mathbf{x}\|_{\boldsymbol{\Sigma}_{m_t,t}^{-1}}$ reflects the uncertainty of the estimated reward $\langle \widehat{\boldsymbol{\theta}}_{m_t,t}, \mathbf{x} \rangle$ and encourages exploration. After receiving the true reward r_t from the environment, agent m_t then updates its local data (Line 7).

[2] Without loss of generality, we can assume that it cannot happen that more than one agent participate at the same time. Therefore, there is always a valid order of participation indexed by $t \in [T]$.

Algorithm 1 Federated linear UCB (`FedlinUCB`)

1: Initialize $\mathbf{\Sigma}_{m,1} = \mathbf{\Sigma}_1^{\text{ser}} = \lambda \mathbf{I}$, $\widehat{\boldsymbol{\theta}}_{m,1} = 0$, $\mathbf{b}_{m,0}^{\text{loc}} = 0$ and $\mathbf{\Sigma}_{m,0}^{\text{loc}} = 0$ for all $m \in [M]$
2: **for** round $t = 1, 2, \ldots, T$ **do**
3: Agent m_t is active
4: Receive D_t from the environment
5: Select $\mathbf{x}_t \leftarrow \operatorname{argmax}_{\mathbf{x} \in D_t} \langle \widehat{\boldsymbol{\theta}}_{m_t,t}, \mathbf{x} \rangle + \beta \|\mathbf{x}\|_{\mathbf{\Sigma}_{m_t,t}^{-1}}$ `/* Optimistic decision */`
6: Receive r_t from environment
7: $\mathbf{\Sigma}_{m_t,t}^{\text{loc}} \leftarrow \mathbf{\Sigma}_{m_t,t-1}^{\text{loc}} + \mathbf{x}_t \mathbf{x}_t^\top, \quad \mathbf{b}_{m_t,t}^{\text{loc}} \leftarrow \mathbf{b}_{m_t,t-1}^{\text{loc}} + r_t \mathbf{x}_t$ `/* Local update */`
8: **if** $\det(\mathbf{\Sigma}_{m_t,t} + \mathbf{\Sigma}_{m_t,t}^{\text{loc}}) > (1+\alpha) \det(\mathbf{\Sigma}_{m_t,t})$ **then**
9: Agent m_t sends $\mathbf{\Sigma}_{m_t,t}^{\text{loc}}$ and $\mathbf{b}_{m_t,t}^{\text{loc}}$ to server `/* Upload */`
10: $\mathbf{\Sigma}_t^{\text{ser}} \leftarrow \mathbf{\Sigma}_t^{\text{ser}} + \mathbf{\Sigma}_{m_t,t}^{\text{loc}}, \quad \mathbf{b}_t^{\text{ser}} \leftarrow \mathbf{b}_t^{\text{ser}} + \mathbf{b}_{m_t,t}^{\text{loc}}$ `/* Global update */`
11: $\mathbf{\Sigma}_{m_t,t}^{\text{loc}} \leftarrow 0, \quad \mathbf{b}_{m_t,t}^{\text{loc}} \leftarrow 0$
12: Server sends $\mathbf{\Sigma}_t^{\text{ser}}$ and $\mathbf{b}_t^{\text{ser}}$ back to agent m_t `/* Download */`
13: $\mathbf{\Sigma}_{m_t,t+1} \leftarrow \mathbf{\Sigma}_t^{\text{ser}}, \quad \mathbf{b}_{m_t,t+1} \leftarrow \mathbf{b}_t^{\text{ser}}$
14: $\widehat{\boldsymbol{\theta}}_{m_t,t+1} \leftarrow \mathbf{\Sigma}_{m_t,t+1}^{-1} \mathbf{b}_{m_t,t+1}$ `/* Compute estimate */`
15: **else**
16: $\mathbf{\Sigma}_{m_t,t+1} \leftarrow \mathbf{\Sigma}_{m,t}, \quad \mathbf{b}_{m_t,t+1} \leftarrow \mathbf{b}_{m,t}, \quad \widehat{\boldsymbol{\theta}}_{m_t,t+1} \leftarrow \widehat{\boldsymbol{\theta}}_{m_t,t}$
17: **end if**
18: **for** other inactive agent $m \in [M] \setminus \{m_t\}$ **do**
19: $\mathbf{\Sigma}_{m,t+1} \leftarrow \mathbf{\Sigma}_{m,t}, \quad \mathbf{b}_{m,t+1} \leftarrow \mathbf{b}_{m,t}, \quad \widehat{\boldsymbol{\theta}}_{m,t+1} \leftarrow \widehat{\boldsymbol{\theta}}_{m,t}$
20: **end for**
21: **end for**

Notation	Meaning
$\widehat{\boldsymbol{\theta}}_{m,t}$	estimate of $\boldsymbol{\theta}^*$
$\mathbf{\Sigma}_{m,t}, \mathbf{b}_{m,t}$	data used to compute $\widehat{\boldsymbol{\theta}}_{m,t}$
$\mathbf{\Sigma}_{m,t}^{\text{loc}}, \mathbf{b}_{m,t}^{\text{loc}}$	local data for agent m
$\mathbf{\Sigma}_{m,t}^{\text{ser}}, \mathbf{b}_{m,t}^{\text{ser}}$	data stored at the server

Table 2: Notations used in Algorithm 1.

The key component of the algorithm is the matrix determinant-based criterion (Line 8), which evaluates the information accumulated in current local data. If the criterion is satisfied, it suggests that the local data would help significantly reduce the uncertainty of estimating the model $\boldsymbol{\theta}^*$. Therefore, agent m_t will share its progress by uploading the local data to the server (Line 9) so that it can benefit other agents. Then the server updates the global data accordingly (Line 10). Afterwards, agent m_t downloads the latest global data from the server (Line 12) and updates its local data and model (Line 13-14). If the criterion in Line 8 is not met, then the communication between the agent and the server will not be triggered, and the local data remains unchanged for agent m_t (Line 16). Finally, all the other inactive agents remain unchanged (Line 19).

Note that in Algorithm 1, the communication between the agent and the server (Line 9 and 12) involves only the active agent in that round, which is completely independent of other agents. This is in sharp contrast to existing algorithms. For example, in the main algorithm in Li and Wang (2022a), upload by any agent may trigger other agents to download the latest data, while our algorithm does not mandate this. On the other hand, many existing algorithms for multi-agent settings (e.g., Wang et al. (2019)) require all agents to interact with the environment in each round, which essentially require full participation of all the agents.

The determinant-based criterion in Line 8 has been a long-standing design trick in contextual linear bandits that can help address the issue of unknown time horizon and reduce the switching cost (Abbasi-Yadkori et al., 2011; Ruan et al., 2021). For multi-agent bandits, such a criterion has also been used to control the communications complexity (Wang et al., 2019; Dubey and Pentland, 2020; Li and Wang, 2022a). This is because the need for policy switching or communication essentially reflects the same fact: enough information has been collected and it is time to update the (local) model. Therefore, achieving low communication complexity and low switching cost are unified in our `FedlinUCB` algorithm in the sense that the communication complexity is exactly twice the switching cost. Furthermore, using lazy update makes our algorithm amenable for analysis, which will be clear later in Section 6. In addition, we leave $\alpha > 0$ as a tuning parameter as it controls the trade-off between the regret and the communication complexity.

5 Theoretical Results

We now present our main result on the theoretical guarantee of Algorithm 1.

Theorem 5.1. Under Assumption 3.1 for Algorithm 1, if we set the confidence radius $\beta = \sqrt{\lambda}S + (\sqrt{1+M\alpha} + M\sqrt{2\alpha})\Big(R\sqrt{d\log\Big((1+TL^2/(\min(\alpha,1)\lambda))/\delta\Big)} + \sqrt{\lambda}S\Big)$, then with probability at least $1-\delta$, the regret in the first T rounds can be upper bounded by

$$\mathrm{Regret}(T) \leq 2dSLM\log(1+TL^2/\lambda) + 2\sqrt{2(1+M\alpha)}\beta\sqrt{2dT\log(1+TL^2/\lambda)}.$$

Moreover, the communication complexity and switching cost are both bounded by $2\log 2 \cdot d(M + 1/\alpha)\log(1+TL^2/(\lambda d))$.

Remark 5.2. Theorem 5.1 suggests that if we set the parameters $\alpha = 1/M^2$ and $\lambda = 1/S^2$ in Algorithm 1, then its regret is bounded by $\widetilde{O}(Rd\sqrt{T})$ and the corresponding communication complexity and switching cost are both bounded by $\widetilde{O}(dM^2)$. This choice of parameters yields the regret bound and the communication complexity presented in Table 1.

As a complement, we also provide a lower bound for the communication complexity as stated in the following theorem. See Appendix C for the proof.

Theorem 5.3. For any algorithm **Alg** with expected communication complexity less than $O(M/\log(T/M))$, there exist a linear bandit instance with $R = L = S = 1$ such that for $T \geq Md$, the expected regret for algorithm **Alg** is at least $\Omega(d\sqrt{MT})$.

Remark 5.4. Suppose each agent runs the `OFUL` algorithm (Abbasi-Yadkori et al., 2011) separately, then each agent $m \in [M]$ admits an $\widetilde{O}(d\sqrt{T_m})$ regret, where T_m is the number of rounds that agent m participates in. Thus the total regret of M agents is upper bounded by $\sum_{m=1}^{M}\widetilde{O}(\sqrt{T_m}) = \widetilde{O}(d\sqrt{MT})$. Theorem 5.3 implies that for any algorithm **Alg**, if its communication complexity is less than $O(M/\log(T/M))$, then its regret cannot be better than naively running M independent `OFUL` algorithms. In other words, Theorem 5.3 suggests that in order to perform better than the single-agent algorithm through collaboration, an $\Omega(M)$ communication complexity is necessary.

6 Overview of the Proof

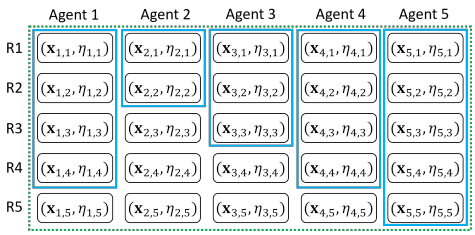

Figure 1: Illustration of ill-defined filtration.

When analyzing the performance of `FedLinUCB`, we face a unique challenge caused by the asynchronous communication, as illustrated in Figure 1. Here $(\mathbf{x}_{m,t}, \eta_{m,t})$ denotes the decision and the noise for agent m in its own t-th round. Specifically, in the synchronous setting, the filtration is generated by all the data collected by all agents, i.e., $\mathcal{F}_5 = \sigma\{\mathbf{x}_{m,t}, \eta_{m,t}\}_{t=1,m=1}^{5,5}$, as marked by the green dashed rectangle. This kind of filtration is well-defined since all agents share their data with each other at the end of each round. In sharp contrast, in our asynchronous setting, the data at the server can be generated by an irregular set of data from the agents, as marked by the blue rectangles. Such data pattern can be arbitrary and depends on the data collected in all previous rounds, which prevents us from defining a fixed filtration as we can do in the synchronous setting. Since the application of standard martingale concentration inequalities relies on the well-defined filtration, they cannot be directly applied to our asynchronous setting.

To circumvent the above issue, we need to analyze the concentration property of the local data for each agent and then relate it to the concentration of the global data, so that we can control the sum of the bonuses and hence the regret. This requires a careful quantitative comparison of the local and global data covariance matrices, which is enabled by our design of determinant-based criterion. The details will be further explained in Section 6.2. Next, we present the key ingredients in the proof of Theorem 5.1.

Remark 6.1. Recall the notations in Table 2, where the values of those matrices and vectors might change within each round. To eliminate the possible confusion, from now on we follow the convention that all matrices and vectors in the analysis correspond to their values at the end of each round in Algorithm 1.

6.1 Analysis for communication complexity and switching cost

We first analyze the communication complexity and switching cost of Algorithm 1. For each $i \geq 0$, we define

$$\tau_i = \min\{t \in [T] \mid \det(\mathbf{\Sigma}_t^{\text{ser}}) \geq 2^i \lambda^d\}. \tag{6.1}$$

We divide the set of all rounds into epochs $\{\tau_i, \tau_i + 1, \ldots, \tau_{i+1} - 1\}$ for each $i \geq 0$. Then the communication complexity *within each epoch* can be bounded using the following lemma.

Lemma 6.2. Under the setting of Theorem 5.1, for each epoch from round τ_i to round $\tau_{i+1} - 1$, the number of communications in this epoch is upper bounded by $2(M + 1/\alpha)$.

Proof of Theorem 5.1: communication complexity and switching cost. It suffices to bound the number of epochs. By Assumption 3.1, we have $\|\mathbf{x}_t\|_2 \leq L$ for all $t \in [T]$. Since $\mathbf{\Sigma}_T^{\text{ser}}$ is positive definite, by the inequality of arithmetic and geometric means, we have

$$\begin{aligned}\det(\mathbf{\Sigma}_T^{\text{ser}}) &\leq \left(\frac{\operatorname{tr}(\mathbf{\Sigma}_T^{\text{ser}})}{d}\right)^d \leq \left(\frac{1}{d}\operatorname{tr}\left(\lambda \mathbf{I} + \sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t^\top\right)\right)^d \\ &= \left(\lambda + \frac{1}{d}\sum_{t=1}^T \|\mathbf{x}_t\|_2^2\right)^d \leq \lambda^d \left(1 + \frac{TL^2}{\lambda d}\right)^d.\end{aligned}$$

Then recalling the definition of epochs based on (6.1), we have

$$\max\{i \geq 0 \mid \tau_i \neq \emptyset\} = \log_2 \frac{\det(\mathbf{\Sigma}_T^{\text{ser}})}{\lambda^d} \leq \log 2 \cdot d \log\left(1 + \frac{TL^2}{\lambda d}\right).$$

Therefore, the total number of epochs is bounded by $\log 2 \cdot d \log(1 + TL^2/(\lambda d))$. Now applying Lemma 6.2, the total communication complexity is bounded by $2 \log 2 \cdot d(M + 1/\alpha) \log(1 + TL^2/(\lambda d))$. Note that in Algorithm 1, each agent only switch its policy after communicating with the server, so the switching cost is exactly equal to half of the communication complexity. This finishes the proof for the claim on communication complexity and switching cost in Theorem 5.1. □

6.2 Analysis for regret upper bound

The regret analysis for Theorem 5.1 is much more involved, and it relies on a series of intermediate lemmas establishing the concentration.

Total information. We define the following auxiliary matrices and vectors that contain all the information up to round t:

$$\mathbf{\Sigma}_t^{\text{all}} = \lambda \mathbf{I} + \sum_{i=1}^t \mathbf{x}_i \mathbf{x}_i^\top, \qquad \mathbf{b}_t^{\text{all}} = \sum_{i=1}^t r_i \mathbf{x}_i, \qquad \mathbf{u}_t^{\text{all}} = \sum_{i=1}^t \eta_i \mathbf{x}_i, \tag{6.2}$$

where $\eta_i := r_i - \langle \mathbf{x}_i, \boldsymbol{\theta}^* \rangle$ is a R-sub-Gaussian noise by Assumption 3.1. In our setting, $\mathbf{\Sigma}_t^{\text{all}}, \mathbf{b}_t^{\text{all}}, \mathbf{u}_t^{\text{all}}$ are not accessible by the agents due to asynchronous communication, and we only use them to facilitate the analysis. With this notation, we can further define the following omnipotent estimate:

$$\widehat{\boldsymbol{\theta}}_t^{\text{all}} = (\mathbf{\Sigma}_t^{\text{all}})^{-1} \mathbf{b}_t^{\text{all}}. \tag{6.3}$$

As a direct application of the self-normalized martingale concentration inequality (Abbasi-Yadkori et al., 2011), we have the following global confidence bound due to the concentration of $\mathbf{\Sigma}_t^{\text{all}}$ and $\mathbf{b}_t^{\text{all}}$.

Lemma 6.3 (Global confidence bound; Theorem 2, Abbasi-Yadkori et al., 2011)**.** With probability at least $1 - \delta$, for each round $t \in [T]$, the estimate $\widehat{\boldsymbol{\theta}}_t^{\text{all}}$ in (6.3) satisfies

$$\|\widehat{\boldsymbol{\theta}}_t^{\text{all}} - \boldsymbol{\theta}^*\|_{\mathbf{\Sigma}_t^{\text{all}}} \leq R\sqrt{d \log\big((1 + TL^2/\lambda)/\delta\big)} + \sqrt{\lambda} S.$$

Per-agent information. Next, for each agent $m \in [M]$, we denote the rounds when agent m communicate with the server (i.e., upload and download data) as $\{t_{m,1}, t_{m,2}, \ldots\}$. For simplicity, *at the end of round* t, we denote by $N_m(t)$ the last round when agent m communicated with the server (so if agent m communicated with the server in round t, then $N_m(t) = t$). With this notation, for each round t and agent $m \in [M]$, the data that has been uploaded by agent m is then[3]

$$\mathbf{\Sigma}_{m,t}^{\text{up}} = \sum_{j=1, m_j=m}^{N_m(t)} \mathbf{x}_j \mathbf{x}_j^\top, \qquad \mathbf{u}_{m,t}^{\text{up}} = \sum_{j=1, m_j=m}^{N_m(t)} \mathbf{x}_j \eta_j.$$

Correspondingly, the local data that has not been uploaded to the server is

$$\mathbf{\Sigma}_{m,t}^{\text{loc}} = \sum_{j=N_m(t)+1, m_j=m}^{t} \mathbf{x}_j \mathbf{x}_j^\top, \qquad \mathbf{u}_{m,t}^{\text{loc}} = \sum_{j=N_m(t)+1, m_j=m}^{t} \mathbf{x}_j \eta_j.$$

Again, applying the self-normalized martingale concentration (Abbasi-Yadkori et al., 2011) together with a union bound, we can get the per-agent local concentration.

Lemma 6.4 (Local concentration)**.** Under the setting of Theorem 5.1, with probability at least $1 - \delta$, for each round $t \in [T]$ and each agent $m \in [M]$, it holds that

$$\left\| \left(\alpha\lambda \mathbf{I} + \mathbf{\Sigma}_{m,t+1}^{\text{loc}}\right)^{-1} \mathbf{u}_{m,t}^{\text{loc}} \right\|_{\alpha\lambda\mathbf{I} + \mathbf{\Sigma}_{m,t}^{\text{loc}}} \leq R\sqrt{d \log\left(\left(1 + TL^2/(\alpha\lambda)\right)/\delta\right)} + \sqrt{\lambda} S.$$

Moreover, based on our determinant-based communication criterion, we have the following lemma that describes the quantitative relationship among the local data, uploaded data and global data.

Lemma 6.5 (Covariance comparison)**.** Under the setting of Theorem 5.1, it holds that

$$\lambda \mathbf{I} + \sum_{m'=1}^{M} \mathbf{\Sigma}_{m',t}^{\text{up}} \succeq \frac{1}{\alpha} \mathbf{\Sigma}_{m,t}^{\text{loc}} \tag{6.4}$$

for each agent $m \in [M]$. Moreover, for any $1 \leq t_1 < t_2 \leq T$, if agent m is the only active agent from round t_1 to $t_2 - 1$ and agent m only communicates with the server at round t_1, then for all $t_1 + 1 \leq t \leq t_2$, it further holds that

$$\mathbf{\Sigma}_{m,t} \succeq \frac{1}{1 + M\alpha} \mathbf{\Sigma}_t^{\text{all}}. \tag{6.5}$$

Combining the above results, we obtain the local confidence bound which then leads to the per-round regret in each round, as summarized in the following lemma.

Lemma 6.6 (Local confidence bound and per-round regret)**.** Under the setting of Theorem 5.1, with probability at least $1-\delta$, for each $t \in [T]$, the estimate $\widehat{\boldsymbol{\theta}}_{m,t+1}$ satisfies that $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{m,t+1}\|_{\mathbf{\Sigma}_{m,t+1}} \leq \beta$ for all $m \in [M]$. Consequently, for each round $t \in [T]$, the regret in round t satisfies

$$\Delta_t = \max_{\mathbf{x} \in \mathcal{D}_t} \langle \boldsymbol{\theta}^*, \mathbf{x} \rangle - \langle \boldsymbol{\theta}^*, \mathbf{x}_t \rangle \leq 2\beta \sqrt{\mathbf{x}_t^\top \mathbf{\Sigma}_{m_t,t}^{-1} \mathbf{x}_t}.$$

Now, we are ready to prove the regret bound in Theorem 5.1.

Proof of Theorem 5.1: regret. Firstly, according to Lemma 6.6, the regret in first T round can be decomposed and upper bounded by

$$\text{Regret}(T) = \sum_{t=1}^{T} \langle \boldsymbol{\theta}^*, \mathbf{x}_t^* - \mathbf{x}_t \rangle \leq \sum_{t=1}^{T} 2\beta \|\mathbf{x}_t\|_{\mathbf{\Sigma}_{m_t,t}^{-1}}.$$

Now, we only need to control the summation of the upper confidence bonus term $2\beta\|\mathbf{x}_t\|_{\mathbf{\Sigma}_{m_t,t}^{-1}}$ and we focus on the agent-action sequence $\{(m_t, \mathbf{x}_t)\}_{t=1}^{T}$. Notice that if an agent m communicate with

[3] Strictly speaking, the uploaded data only consists of $\mathbf{\Sigma}_{m,t}^{\text{up}}$ and $\mathbf{b}_{m,t}^{\text{up}}$, and here we introduce $\mathbf{u}_{m,t}^{\text{up}}$ and $\mathbf{u}_{m,t}^{\text{loc}}$ solely for the purpose of analysis.

the server at time t_1 and t_2, then the order of actions between round t_1 and t_2 will not effect the covariance matrix for agent m. In addition, since agent m does not upload new data between round t_1 and t_2, the order of actions from agent m will also not affect other agents' covariance matrix. Thus, without effect the covariance matrix and the corresponding upper confidence bonus, we can always reorder the sequence of active agents such that each agent communicates with the server and stays active until the next agent kicks in to communicate with the server. Such reordering is valid according to the communication protocol as each agent has only local updates between communications with the server.

More specifically, we assume that the sequence of round that the active agent communicates with server is $0 = t_0 < t_1 < t_2 < \cdots < t_N = T + 1$[4], and from round $t_i + 1$ to $t_{i+1} - 1$ there is only one agent active, that is, $m_{t_i} = m_{t_i+1} = \cdots = m_{t_{i+1}-1}$. Then we apply Lemma 6.5 and get that the bonus term for each round $t_i < t < t_{i+1}$ can be controlled by

$$2\beta\|\mathbf{x}_t\|_{(\mathbf{\Sigma}_{m_t,t})^{-1}} \leq 2\beta\sqrt{1+M\alpha}\|\mathbf{x}_t\|_{(\mathbf{\Sigma}_t^{\text{all}})^{-1}}. \tag{6.6}$$

In addition, to control the bonus term for rounds $\{t_i\}_{i=1}^N$, we define $T_i = \min\{t \in [T] \mid \det(\mathbf{\Sigma}_t^{\text{all}}) \geq 2^i\lambda^d\}$. For each time interval from T_i to T_{i+1}, if an agent m communicate with the server more than once, e.g., agent m communicates with the server at round $T_{i,1}$ and $T_{i,2}$ such that $T_i \leq T_{i,1} < T_{i,2} < T_{i+1}$, then for the latter round $T_{i,2}$, the bonus term can be controlled by

$$\begin{aligned}
2\beta\|\mathbf{x}_{T_{i,2}}\|_{(\mathbf{\Sigma}_{m,T_{i,2}})^{-1}} &\leq 2\beta\sqrt{1+M\alpha}\|\mathbf{x}_{T_{i,2}}\|_{(\mathbf{\Sigma}_{T_{i,1}}^{\text{all}})^{-1}} \\
&\leq 2\beta\sqrt{2(1+M\alpha)}\|\mathbf{x}_{T_{i,2}}\|_{(\mathbf{\Sigma}_{T_{i+1}-1}^{\text{all}})^{-1}} \\
&\leq 2\beta\sqrt{2(1+M\alpha)}\|\mathbf{x}_{T_{i,2}}\|_{(\mathbf{\Sigma}_{T_{i,2}}^{\text{all}})^{-1}},
\end{aligned} \tag{6.7}$$

where the first inequality holds due Lemma 6.5 and agent m communicate with the server at the previous round $T_{i,1}$, the second inequality holds due to Lemma D.4 with the fact that $\det \mathbf{\Sigma}_{T_{i+1}-1}^{\text{all}} / \det(\mathbf{\Sigma}_{T_{i,1}}^{\text{all}}) \leq 2^{i+1}\lambda^d/(2^i\lambda^d) = 2$, and the last inequality holds due to the fact that $\mathbf{\Sigma}_{T_{i+1}-1}^{\text{all}} \succeq \mathbf{\Sigma}_{T_{i,2}}^{\text{all}}$. On the other hand, for each time interval from T_i to T_{i+1}, the bonus term for the first communication can always be trivially bounded by 1 for all agent m. Therefore, the summation of the regret over first communication can be upper bounded by 1. In addition, since the norm of each action $\|\mathbf{x}\|_2 \leq L$ and it implies that we have

$$\det(\mathbf{\Sigma}_T^{\text{all}}) \leq (\lambda + TL^2)^d, \tag{6.8}$$

which implies that the number of different intervals is at most $d\log(1 + TL^2/\lambda)$. Combining the upper bound of regret in (6.6), (6.7) and (6.8), we have

$$\begin{aligned}
\text{Regret}(T) &\leq dM\log(1+TL^2/\lambda) + \sum_{t=1}^{T} 2\sqrt{2(1+M\alpha)}\beta\|\mathbf{x}_{T_{i,2}}\|_{(\mathbf{\Sigma}_{T_{i,2}}^{\text{all}})^{-1}} \\
&\leq d\log(1+TL^2/\lambda) + 2\sqrt{2(1+M\alpha)}\beta\sqrt{2dT\log(1+TL^2/\lambda)},
\end{aligned}$$

where the last inequality follows from a standard elliptic potential argument (Abbasi-Yadkori et al., 2011). □

7 Conclusion and Future Work

In this work, we study federated contextual linear bandit problem with fully asynchronous communication. We propose a simple and provably efficient algorithm named `FedLinUCB`. We prove that `FedLinUCB` obtains a near-optimal regret of order $\widetilde{O}(d\sqrt{T})$ with $\widetilde{O}(dM^2)$ communication complexity. We also prove a lower bound on the communication complexity, which suggests that an $\Omega(M)$ communication complexity is necessary for achieving a near-optimal regret. There still exists an $O(dM)$ gap between the upper and lower bounds for the communication complexity and we leave it as a future work to close this gap. Another important direction for future work is to study federated linear bandits with a decentralized communication network without a central server (i.e., P2P networks). In addition, there are potential privacy concerns when the agents upload and download data from the server, and it remains an open problem to devise provably efficient algorithm for asynchronous federated linear bandits with privacy guarantees.

[4]There is no comminucation happening at t_0 or t_N, but we include them for notational convenience.

References

ABBASI-YADKORI, Y., PÁL, D. and SZEPESVÁRI, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems* **24** 2312–2320.

ABE, N., BIERMANN, A. W. and LONG, P. M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica* **37** 263–293.

AGRAWAL, S. and GOYAL, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR.

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

CESA-BIANCHI, N., GENTILE, C. and ZAPPELLA, G. (2013). A gang of bandits. *Advances in neural information processing systems* **26**.

CHAKRABORTY, M., CHUA, K. Y. P., DAS, S. and JUBA, B. (2017). Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *IJCAI*.

CHU, W., LI, L., REYZIN, L. and SCHAPIRE, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings.

DANI, V., 9, ., HAYES, T. and KAKADE, S. M. (2008). Stochastic linear optimization under bandit feedback. *21st Annual Conference on Learning Theory* 355–366.

DEKEL, O., DING, J., KOREN, T. and PERES, Y. (2014). Bandits with switching costs: T 2/3 regret. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*.

DUBEY, A. and PENTLAND, A. (2020). Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems* **33** 6003–6014.

DURAND, A., ACHILLEOS, C., IACOVIDES, D., STRATI, K., MITSIS, G. D. and PINEAU, J. (2018). Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine learning for healthcare conference*. PMLR.

FILIPPI, S., CAPPE, O., GARIVIER, A. and SZEPESVÁRI, C. (2010). Parametric bandits: The generalized linear case. *Advances in Neural Information Processing Systems* **23**.

GENTILE, C., LI, S. and ZAPPELLA, G. (2014). Online clustering of bandits. In *International Conference on Machine Learning*. PMLR.

HUANG, R., WU, W., YANG, J. and SHEN, C. (2021). Federated linear contextual bandits. *Advances in Neural Information Processing Systems* **34**.

JAGADEESAN, M., WEI, A., WANG, Y., JORDAN, M. and STEINHARDT, J. (2021). Learning equilibria in matching markets from bandit feedback. *Advances in Neural Information Processing Systems* **34**.

JUN, K.-S., BHARGAVA, A., NOWAK, R. and WILLETT, R. (2017). Scalable generalized linear bandits: Online computation and hashing. *Advances in Neural Information Processing Systems* **30**.

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

KARIMIREDDY, S. P., JAGGI, M., KALE, S., MOHRI, M., REDDI, S. J., STICH, S. U. and SURESH, A. T. (2020). Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606* .

KORDA, N., SZORENYI, B. and LI, S. (2016). Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*. PMLR.

LANDGREN, P., SRIVASTAVA, V. and LEONARD, N. E. (2016). On distributed cooperative decision-making in multiarmed bandits. In *2016 European Control Conference (ECC)*. IEEE.

LANDGREN, P., SRIVASTAVA, V. and LEONARD, N. E. (2018). Social imitation in cooperative multiarmed bandits: Partition-based algorithms with strictly local information. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE.

LATTIMORE, T. and SZEPESVÁRI, C. (2020). *Bandit algorithms*. Cambridge University Press.

LI, C. and WANG, H. (2022a). Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

LI, C. and WANG, H. (2022b). Communication efficient federated learning for generalized linear bandits. *arXiv preprint arXiv:2202.01087* .

LI, C., WU, Q. and WANG, H. (2021). Unifying clustered and non-stationary bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

LI, L., CHU, W., LANGFORD, J. and SCHAPIRE, R. E. (2010a). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*.

LI, S., KARATZOGLOU, A. and GENTILE, C. (2016). Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*.

LI, T., SONG, L. and FRAGOULI, C. (2020). Federated recommendation system via differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE.

LI, W., WANG, X., ZHANG, R., CUI, Y., MAO, J. and JIN, R. (2010b). Exploitation and exploration in a performance based contextual advertising system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*.

LI, Y., WANG, C.-H., CHENG, G. and SUN, W. W. (2022). Rate-optimal contextual online matching bandit. *arXiv preprint arXiv:2205.03699* .

LI, Y., WANG, Y. and ZHOU, Y. (2019). Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*. PMLR.

LIU, K. and ZHAO, Q. (2010). Distributed learning in multi-armed bandit with multiple players. *IEEE transactions on signal processing* **58** 5667–5681.

MARTÍNEZ-RUBIO, D., KANADE, V. and REBESCHINI, P. (2019). Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems* **32**.

MCMAHAN, B., MOORE, E., RAMAGE, D., HAMPSON, S. and Y ARCAS, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR.

RUAN, Y., YANG, J. and ZHOU, Y. (2021). Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*.

RUSMEVICHIENTONG, P. and TSITSIKLIS, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research* **35** 395–411.

SANKARARAMAN, A., GANESH, A. and SHAKKOTTAI, S. (2019). Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* **3** 1–35.

SZORENYI, B., BUSA-FEKETE, R., HEGEDUS, I., ORMÁNDI, R., JELASITY, M. and KÉGL, B. (2013). Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*. PMLR.

TIE, L., CAI, K.-Y. and LIN, Y. (2011). Rearrangement inequalities for hermitian matrices. *Linear algebra and its applications* **434** 443–456.

WANG, P.-A., PROUTIERE, A., ARIU, K., JEDRA, Y. and RUSSO, A. (2020). Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

WANG, Y., HU, J., CHEN, X. and WANG, L. (2019). Distributed bandit learning: Near-optimal regret with efficient communication. *arXiv preprint arXiv:1904.06309* .

WANG, Y.-G. (1991). Sequential allocation in clinical trials. *Communications in Statistics-Theory and Methods* **20** 791–805.

WU, Q., WANG, H., GU, Q. and WANG, H. (2016). Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*.

ZHU, Z., ZHU, J., LIU, J. and LIU, Y. (2021). Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work is seeking to develop a mathematical understanding of federated linear bandits.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We use existing theorems and cite them properly.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]