Federated Online Clustering of Bandits

Abstract

Contextual multi-armed bandit (MAB) is an important sequential decision making problem in recommendation systems. A line of works, called the clustering of bandits (CLUB), utilizes the collaborative effect over users and dramatically improves the recommendation quality. Owing to the increasing scale of application and public concerns of privacy, there is a growing demand to keep user data decentralized and push bandit learning to the local server side. Existing CLUB algorithms, however, are designed under the centralized setting where data are available at a central server. We focus on the study of the federated clustering of bandit (FCLUB) problem, whose task is to minimize the total regret meanwhile satisfying privacy and communication considerations. For this problem, we design a new phase-based scheme for cluster detection and a novel asynchronous communication protocol for cooperative bandit learning. To protect users' privacy, previous differential privacy (DP) definitions are not very suitable and we propose a new DP notion which acts on the user cluster level. We provide rigorous proofs to show that our algorithm simultaneously achieves (clustered) DP, sublinear communication complexity and sublinear regret. Finally, experimental evaluations show our superior performance compared with benchmark algorithms.

1 INTRODUCTION

Stochastic multi-armed bandit (MAB) [Auer et al., 2002] is a well-known sequential decision-making problem, where a learner sequentially selects actions so as to maximize the cumulative rewards (or minimize the cumulative regret). One fruitful application area of MAB is the online recommendation systems (RecSys) [Chu et al., 2011, Abbasi-Yadkori et al., 2011, Gentile et al., 2014, Li et al., 2019, Zhang et al., 2020, Li et al., 2021], where MAB algorithms provide a principled way to handle the challenge of exploration-exploitation trade-off [Lattimore and Szepesvári, 2020].

To advance the bandit algorithm for large-scale applications, contextual linear bandits adds the simple yet effective linear structure assumptions on actions and reward functions [Chu et al., 2011, Li et al., 2010, Abbasi-Yadkori et al., 2011]. One limitation, however, is that such a model mainly works in a content-dependent manner, ignoring the often used tool of collaborative filtering. To address this issue, the clustering of bandits (CLUB) are proposed [Gentile et al., 2014, Li et al., 2016, Li and Zhang, 2018, Li et al., 2019]. The CLUB algorithms adaptively cluster similar users and utilize the collaborative information given by the cluster structure, which dramatically improves the quality of recommendation.

While most existing bandit algorithms are designed under a centralized setting, in response to the increasing scale of application and public concerns of privacy, there is a growing demand to keep user data decentralized and push the learning of bandit models to the client or the local server side . This paradigm is now known as the federated learning [Kairouz et al., 2021]. Owing to its overall applicability, there has been a surge of interest in studying federated MAB [Dubey and Pentland, 2020, Zhu et al., 2021, Shi and Shen, 2021], which promises cooperative bandit learning with larger amounts of data (across multiple local servers) while keeping the data decentralized. This motives us to study the CLUB problem to its federated counterpart, i.e., the federated clustering of bandits (FCLUB). In FCLUB, each local server is allowed to conduct its own local clustering of bandit algorithms. To enable the collaborative effects of users over different servers, the local server could also collaborate with other local servers under the coordination of a global server, whose communication needs to satisfy certain privacy and communication requirements. The goal of this work is to design an federated online clustering of bandit framework, so as to minimize the T-round regret un-

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

der the privacy protection requirements and communication cost considerations.

The key challenge of FCLUB is designing collaborative bandit learning procedures and cluster detection strategies to identify the overall cluster structures across different local servers, where each local server only hold part of the users with unknown interests. Such a problem is more challenging due to the following privacy and communication cost requirements, which are two first-order requirements for any federated applications [Kairouz et al., 2021].

Privacy protection: To reduce the privacy leakage of each user, we expect local servers to only share user clusters' data instead of individuals' raw data. In addition, we still need a mechanism to protect the uploaded (cluster) information against possible adversaries outside the local server, for which we adopt the solution concept differential privacy (DP). Off-the-shelf DP notion, however, is defined on individual users, hence are not suitable for FCLUB. It is challenging and unclear what is a suitable notion of privacy over the clustering of users and how to devise algorithms to guarantee the corresponding privacy requirements.

Communication: Communication is critical for collaborative learning, but may also be expensive or time-consuming. For FCLUB, it is desired to minimize the total regret while keeping the communication costs (in terms of communication rounds) as low as possible. Another requirement is to design an asynchronous communication protocol incorporating the randomly arriving users and possibly lagging servers, which prevents commonly used synchronous protocols [Dubey and Pentland, 2020].

1.1 OUR CONTRIBUTIONS

To address the aforementioned challenges, this paper makes four contributions.

1. Problem formulation: We propose the setting of online clustering of bandits to its federated counterpart, which considers the privacy protection and communication requirements. We also propose a novel cluster differential privacy (CDP) notion tailored for the FCLUB setting.

2. Algorithm design: We propose a private and communication efficient FCLUB-CDP algorithm. For privacy protection, a tree-based privatizer is designed to guarantee our proposed CDP. For communication efficiency, we follow the phase-based principle for cluster detection and propose the asynchronous communication protocol for delayed information sharing. In particular, each local server maintains upload/download buffers and occasionally uploads/downloads the buffered information to/from the global server only if it finds the latest information deviates too far from the last update.

3. Theoretical analysis: We prove that FCLUB-CDP achieves the $O(dL\sqrt{mT\frac{\log(1/\delta)}{\varepsilon}}\log^{1.5}T)$ regret bounds, $O(dmL\log T)$ communication costs and $(\varepsilon, \delta, L, m)$-CDP privacy guarantee, respectively.

4. Experiments: We conduct extensive experiments over synthetic and real-world dataset to validate our theoretical analysis. Empirical results show the superior performance of our algorithm over existing algorithms.

1.2 RELATED WORK

Online clustering of bandits. The online clustering bandits is first proposed by Gentile et al. [2014] and shows its effeteness by accelerating the learning process of contextual bandits. The key idea is to use a graph representing the user similarity and adaptively refine the user clusters for information sharing. This work has been extended by a series of works considering the collaborative effects on both users and items [Li et al., 2016], the context-aware settings [Gentile et al., 2017], the cascading bandit setting [Li and Zhang, 2018] and the users with different user frequency [Li et al., 2019]. However, none of these works consider the privacy constraints and communication cost requirements imposed by the FL paradigm like the current work, and therefore cannot give guarantees on these two critical criterion. Korda et al. [2016] considers the peer-to-peer but non-private clustering of bandit, our work studies the private bandit setting under the orchestration of a global server, which requires different algorithms and analysis.

Federated and distributed bandits. There has been growing interest in bandit learning with multiple players. One line of research investigates the competitive agents with collisions [Anandkumar et al., 2011, Rosenski et al., 2016, Bistritz and Leshem, 2018, Boursier and Perchet, 2019], in which reward for an arm is zero if it is chosen by more than one agents. The goal of these works is to minimize regret without communication, which is different from ours. The cooperative distributed bandits are most related to our work, in which multiple agents collaborates to solve a bandit problem over certain communication networks, e.g., peer-to-peer networks [Korda et al., 2016] or client-server networks [Dubey and Pentland, 2020, Li and Wang, 2021]. Our work belongs to the client-server setting but we differ from both Dubey and Pentland [2020] and Li and Wang [2021] since neither of them consider the clustering effects of users.

Differential privacy. Our work leverages on *differential privacy*, a rigorous mathematical framework of privacy first proposed by Dwork et al. [2006]. We utilize several useful techniques from the standard differential privacy to maintain our cluster differential privacy condition. Most notably, we use a tree-based algorithm which is introduced in Chan et al. [2011] to realize differential privacy for the continual release of statistics. In the single-agent bandit setting, Shariff and Sheffet [2018] also utilizes this tree-based algorithm to achieve Joint DP, which is then extended by Dubey and Pentland [2020] to the federated setting. The closest work to ours is Dubey and Pentland [2020] and they study the

simpler case where each local server only holds one user and all users are identical (with the same unknown preference vector), hence gives the different user-level DP definition with different privatizer and analysis.

To the best of knowledge, this paper is the first to generalize the CLUB to its federated setting, which simultaneously achieves privacy protection and communication requirements.

2 PROBLEM SETTINGS

In this section, we formulate the setting of "Federated Clustering of Bandits" (FCLUB). At the global level, there are n users, denoted by the set $\mathcal{U} = \{u_1, ..., u_n\}$. Each user $i \in \mathcal{U}$ has an *unknown* preference vector $\theta_i \in \mathbb{R}^d$ and for simplicity we assume $\|\theta_i\|$. Since users may have the same/similar preference vector, we assume there exists m (unknown) different preference vectors, i.e., $|\{\theta_1, ..., \theta_n\}| = m$. Users with the same preference vector form an underlying cluster and we denote these m (unrevealed) clusters by $\mathcal{C} = \{C_1, ..., C_m\}$. Different from CLUB, users in FCLUB are distributed in L local servers denoted by $\{1, ..., L\}$. At the local level, the local server ℓ contains n^ℓ users $\mathcal{U}^\ell = \{u_1^\ell, ..., u_{n^\ell}^\ell\}$ (with $\bigcup_{\ell \in [L]} \mathcal{U}^\ell = \mathcal{U}$) and similarly, these n^ℓ users form local cluster $\mathcal{C}^\ell = \{C_1^\ell, ..., C_{m_\ell}^\ell\}$ where $m_\ell \leq m$.

The learning agent interacts with the bandit game as the follows. At each time t, a user $i_t \in [n]$ randomly comes with probability $1/n$. Then K items are generated to form a item set D_t, where the feature of each item $x_t \in D_t$ is drawn independently from a fixed but unknown distribution ρ over $\{x \in \mathbb{R}^d : \|x\| \leq 1\}$. The learning agent identifies the local server ℓ_t that i_t belongs to and the current user cluster j_t (detected by our algorithm) which i_t lies in. The local server then recommends an item $x_t \in D_t$ to the user based on the aggregated information from cluster j_t. After i_t receives the recommendation, the learning agent receives a random reward $y_t \in [0, 1]$. Let $\mathcal{H}_t = \{i_1, x_1, y_1, ..., i_{t-1}, x_{t-1}, y_{t-1}, i_t\}$ be the historical information before time t. We assume the expectation of reward y is linear in the feature vector x and the unknown preference vector θ_{i_t}, i.e., $\mathbb{E}_t[y|x] = \theta_{i_t}^\top x$. Now we give some assumptions on preference vectors and item feature vectors. Note that all the assumptions follow the previous works Gentile et al. [2014], Li and Zhang [2018], Li et al. [2019].

Assumption 1 (Gap between preference vectors)**.** *For any two different preference vectors $\theta_{i_1} \neq \theta_{i_2}$, there is a fixed but unknown gap $\gamma > 0$ so that $\|\theta_{i_1} - \theta_{i_2}\| \geq \gamma$.*

Assumption 2 (Item regularity)**.** *For item distribution ρ, there exists a known $\lambda_x > 0$ so that $\mathbb{E}_{x\sim\rho}[xx^\top]$ is full rank with minimal eigenvalue λ_x. Meanwhile, for all time t, for any fixed unit vector $\theta \in \mathbb{R}^d$, $(\theta^\top x)^2$ has sub-Gaussian tail with variance σ_0^2.*

Learning efficiency. The goal of the learning agent is to accumulate as much reward as possible. The optimal item for user i_t at time t is $x_{i_t}^* = \arg\max_{x \in D_t} \theta_{i_t}^\top x$. The learning performance is typically measured by the regret, which is defined by

$$R(T) = \mathbb{E}[\sum_{t=1}^{T} r_t] = \mathbb{E}[\sum_{t=1}^{T} (\theta_{i_t}^\top x_{i_t}^* - \theta_{i_t}^\top x_t)], \quad (1)$$

where r_t is the regret at time t and the expectation is taken over the randomness of the algorithm and the environment regarding the users $i_1, ..., i_T$ and the item sets $D_1, ..., D_T$.

In addition to the regret, privacy protection and the communication cost are two important criterion in federated learning. In this work, we aim to ensure that the user data are protected under privacy constraints and the communication complexity is low.

Privacy requirements. To protect the user data, we introduce two privacy requirements. First, we desire the local server only uploads the user clusters' sufficient statistics (or clustered data) for the learning procedure, instead of individual users' raw data. Second, to protect the clustered data against third-party adversaries outside the local server, we adopt the notion of DP, which encodes the intuition that any observable output changes very little (in probability) when any input datum changes. Since existing DP notions are defined on *user-level* data Shariff and Sheffet [2018], Dubey and Pentland [2020], we introduce a new differential privacy requirement to protect the *cluster-level* data.

The contextual MAB problem involves two sets of variables against the adversaries outside the local server: the decision sets D_t and the observed rewards y_t. Since the users only receive and store observations regarding the chosen action x_t and the observed reward, it suffices to protect $(x_t, y_t)_{t\in[T]}$ to achieve DP requirements. Let $\tau_{\ell,j}$ be time slots when user in cluster j appear at local server ℓ, we denote two sequences $S_{\ell,j} = (x_t, y_t)_{t\in\tau_{\ell,j}}$ and $S'_{\ell,j} = (x'_t, y'_t)_{t\in\tau_{\ell,j}}$ as t-neighboring if $(x_\tau, y_\tau) = (x'_\tau, y'_\tau)$ for $\tau \neq t \in \tau_{\ell,j}$.

Definition 1 (Cluster Differential Privacy)**.** *In a FCLUB setting with L servers and (at most) m clusters, a federated contextual bandit algorithm $A = (A_{\ell,j})_{\ell\in[L],j\in[m]}$ is $(\varepsilon, \delta, L, m)$-CDP, if for any $(\ell, j), (\ell', j')$ s.t. $(\ell, j) \neq (\ell', j')$, any t and the set of sequences $\bar{S}_{\ell',j'} = \cup_{i\in[L],k\in[m]} S_{i,k}$ and $\bar{S}'_{\ell',j'} = \cup_{i\in[L]\setminus\ell',k\in[m]\setminus j'} S_{i,k} \cup S'_{\ell',j'}$ s.t. $\bar{S}_{\ell',j'}$ and $\bar{S}'_{\ell',j'}$ are t neighboring, and for any subset of actions $a_{\ell,j} \subset \prod_{\tau\in\tau_{\ell,j}} \mathcal{D}_\tau$ of actions, it holds that*

$$\Pr[A_{\ell,j}(\bar{S}_{\ell',j'}) \in a_{\ell,j}] \leq e^\varepsilon \Pr[A_{\ell,j}(\bar{S}'_{\ell',j'}) \in a_{\ell,j}] + \delta. \quad (2)$$

Note that our CDP notion formalizes the intuition that the action chosen by any local server (at the cluster level) must be

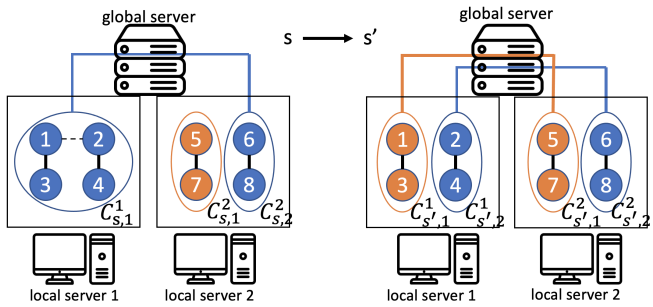


Figure 1: Illustration of how our algorithm detects clusters from phase s to s'. Local server 1 delete edge $(1, 2)$ and split $C^1_{s,1}$ into $C^1_{s',1}, C^1_{s',2}$. The global server merges local clusters as $\{(C^1_{s',1}, C^2_{s',1}), (C^1_{s',2}, C^2_{s',2})\}$.

sufficiently indistinguishable (in probability) to any single (x, y) pair from any other local server. Such a notion does not require each cluster is private to its own observations, i.e., each cluster of users can be trusted with its own data, which is different from the local DP [Zheng et al., 2020] or Joint DP [Shariff and Sheffet, 2018] that assume even itself cannot be trusted.

Communication complexity. To evaluate the communication complexity, we count one upload operation (or one download operation) between any local server and the global server as one communication round. Our communication complexity is the total number of communication rounds C_T over the time horizon T.

3 ALGORITHM

In this section, we introduce our phase-based federated clustering of bandit algorithms with CDP (FCLUB-CDP).

Identify the underlying cluster structure. To correctly identify the cluster structure, we design a phase-based clustering detection algorithm in Algorithm 2. The high level idea is to first conduct local-level clustering of bandits on each local server and merge the local clusters on the global server.

At the local level, each server ℓ maintains a profile of $(V_{t,i}, b_{t,i}, T_{t,i})$ for its own local users $i \in \mathcal{U}^\ell$, where $V_{t,i}$ is Gramian matrix, $b_{t,i}$ is the moment vector of regressand by the regressors, and $T_{t,i}$ is the number of times that i has appeared up to time t. At the beginning of s phase (or $t = 2^s + 1$), based on profiles $(V_{t,i}, b_{t,i}, T_{t,i})_{i \in \mathcal{U}^\ell}$, each server ℓ maintains an undirected graph structure $\mathcal{G}^\ell_s$, where nodes represent all local users $\mathcal{U}^\ell$ and a pair of users are connected by an edge if they are similar. We initialize the graph by a complete graph $\mathcal{G}^\ell_0$ and gradually delete edges at every beginning of each phase s. Specifically, in line 3, we delete edge between any user i_1 and i_2 if the distance between their estimated preference vector are larger than the following threshold.

$$\left\|\hat{\theta}_{t,i_1} - \hat{\theta}_{t,i_2}\right\|_2 > \alpha_1(F(\boldsymbol{T}_{t,i_1}) + F(\boldsymbol{T}_{t,i_2})), \qquad (3)$$

where $\hat{\theta}_{t,i} = (\lambda I + \boldsymbol{V}_{t,i})^{-1}\boldsymbol{b}_{t,i}$ and $F(x) = \sqrt{\frac{1+\ln(1+x)}{(1+x)}}$. After the deletion, users in connected components $j \in C(\mathcal{G}^\ell_s)$ are grouped into local cluster j. In line 5, server ℓ uploads the clustered information $I_{s,\ell} = (C^\ell_{s,j}, \tilde{V}^\ell_{s,j}, \tilde{b}^\ell_{s,j}, \tilde{T}^\ell_{s,j})_{j \in C(\mathcal{G}^\ell_s)}$ to the global server, which contains clustered information for each cluster $j \in C(\mathcal{G}^\ell_s)$. Note that $I_{s,\ell}$ are added with random perturbation to protect the users' data, which will be introduced shortly after.

In line 7, when the global server receives the privatized clustered information from all servers, it performs a merge operation to merge clusters from different servers whose estimated clustered preference vectors are close into a global clusters according to the following inequality.

$$\left\|\hat{\theta}^{\ell_1}_{s,j_1} - \hat{\theta}^{\ell_2}_{s,j_2}\right\|_2 < \alpha_2(F(\tilde{T}^{\ell_1}_{s,j_1}) + F(\tilde{T}^{\ell_2}_{s,j_2})), \qquad (4)$$

where $\hat{\theta}^\ell_{s,j} = (\tilde{V}^\ell_{s,j})^{-1}\tilde{b}^\ell_{s,j}$ and $F(x) = \sqrt{\frac{1+\ln(1+x)}{(1+x)}}$. P_s denotes the set of m_s global clusters. Note that the local clusters in the same global cluster indexed by $k \in [m_s]$ will communicate and share protected clustered information with each other in an asynchronous manner. At the beginning of phase s, if the new global cluster structure P_s is different from P_{s-1} at phase $s-1$, we will renew the shared global information $(S^g_{t,k}, u^g_{t,k}, T^g_{t,k})$ for $k \in [m_s]$. For local servers (ℓ, j) in the same global cluster $P_{s,k}$, the local synchronized information $(S^\ell_{t,j}, u^\ell_{t,j}, T^\ell_{t,j})$, the upload buffers $(\Delta S^\ell_{t,j}, \Delta u^\ell_{t,j}, \Delta T^\ell_{t,j})$ and download buffers $(\Delta S^\ell_{t,j}, \Delta u^\ell_{t,j}, \Delta T^\ell_{t,j})$ are renewed. We also generate a new tree-based privatizer PVT(ℓ, j) for each cluster j at server ℓ, which will be introduced later on.

Asynchronous communication protocol. In this work, we design a novel asynchronous communication protocol to incorporate the randomly arriving users. To reduce the communication cost, our high-level idea is to use the delayed communication, where the feedback are temporarily stored in buffers and only if the stored information exceeds a threshold, the upload/download events are triggered. Such a threshold will ensure that the local information will not diverge too far from the global information, which in turn will not diverge too far from the scenario when information are fully synchronized. Also note that our communication is conducted in the asynchronous manner at the local cluster level. In other words, all local clusters indexed by $(\ell, j) \in P_{s,k}$ will establish connection with each other within the global cluster k. For each local cluster (ℓ, j), it stores a local copy of the sufficient statistics $(S^\ell_{t,j}, u^\ell_{t,j}, T^\ell_{t,j})$ and a upload buffer $(\Delta S^\ell_{t,j}, \Delta u^\ell_{t,j}, \Delta T^\ell_{t,j})$. For the global server, it prepares for each local local cluster (ℓ, j) a download buffer $(\Delta S^{-\ell}_{t,j}, \Delta u^{-\ell}_{t,j}, \Delta T^{-\ell}_{t,j})$, which will be used to send other local servers' information to the local cluster.

Our proposed communication framework consists of two components: the upload protocol (Algorithm 3) and the download protocol (Algorithm 4). For the upload protocol,

Algorithm 1 Phase-based FCLUB with CDP

Input: Failure probability α, deletion parameter $\alpha_1 > 0$, merge parameter $\alpha_2 > 0$, privacy parameters ε, δ.

1: User initialization: For $i \in [n]$, $\boldsymbol{V}_{0,i} = 0_{d\times d}, \boldsymbol{b}_{0,i} = 0_{d\times 1}, \boldsymbol{T}_{0,i} = 0$.
2: Local server initialization: For $\ell \in [L]$, set graph $\mathcal{G}_0^\ell = (\mathcal{U}^\ell, \mathcal{E}_0^\ell)$, local information, upload buffers, download buffers: $(S_{0,1}^\ell, u_{0,1}^\ell, T_{0,1}^\ell) = (\Delta S_{0,1}^\ell, \Delta u_{0,1}^\ell, \Delta T_{0,1}^\ell) = (\Delta S_{0,1}^{-\ell}, \Delta u_{0,1}^{-\ell}, \Delta T_{0,1}^{-\ell}) = (0_{d\times d}, 0_{d\times 1}, 0)$, perturbations $(\bar{H}_{t,1}^\ell, \bar{h}_{t,1}^\ell) = (H_{t,1}^\ell, h_{t,1}^\ell) = (0_{d\times d}, 0_{d\times 1})$.
3: Global server initialization: create one global cluster $P_0 = \{\{(1,1), ..., (L,1)\}\}$ and set the global information for $V_{0,1}^g = 0_{d\times d}$, the $u_{0,1}^g = 0_{d\times 1}, T_{0,1}^g = 0$.
4: **for** $s = 1, 2, ...,$ **do**
5: Detect and adjust clusters (Algorithm 2).
6: **for** $\tau = 1, ..., 2^s$ **do**
7: Compute the total time step $t = 2^s - 2 + \tau$.
8: Advance all parameter, e.g., $S_{t,j}^\ell = S_{t-1,j}^\ell$.
9: User $\boldsymbol{i}_t$ at local server $\boldsymbol{\ell}_t$ arrives and $\boldsymbol{\ell}_t$ gets the local cluster $\boldsymbol{j}_t$ that $\boldsymbol{i}_t$ belongs to based on $\mathcal{G}_s^\ell$.
10: Compute local $\beta_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$ according to Lemma 4.
11: Local server $\boldsymbol{\ell}_t$ receives feasible context set $\mathcal{D}_t = (x_{t,1}, ..., x_{t,K})$ and decides to recommend item $a_t = \arg\max_{i\in[K]} x_{t,i}^\top (S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})^{-1} u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} + \beta_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} \|x_{t,i}\|_{(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})^{-1}}$.
12: User $\boldsymbol{i}_t$ receives feedback $\boldsymbol{y}_t$, update the user $\boldsymbol{i}_t$'s information: $(\boldsymbol{T}_{t,\boldsymbol{i}_t}, \boldsymbol{V}_{t,\boldsymbol{i}_t}, \boldsymbol{b}_{t,\boldsymbol{i}_t}) \mathrel{+}= (1, x_{t,a_t}x_{t,a_t}^\top, y_t x_{t,a_t})$ and others unchanged.
13: Check upload event (Algorithm 3).
14: Check download event (Algorithm 4).
15: **end for**
16: **end for**

at each time step t, user i_t visits the server $\boldsymbol{\ell}_t$ and receives recommended item a_t. After the user i_t interacts with the environment and observes feedback (x_{t,a_t}, y_t), the local server updates the upload buffers in line 1 and checks the following condition to decide whether to upload the upload buffer:

$$\det(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} + \Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})/\det(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) \geq U, \tag{5}$$

where $H_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$ and $\bar{H}_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$ are tentative and current perturbation for privacy protection, respectively. If the condition is satisfied, the local server sends $(\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})$ to the global cluster $\boldsymbol{k}_t$. The global server then merges the uploaded information into the global information in line 4 and also sends it to download buffers for other local clusters $(\ell, j) \neq (\boldsymbol{\ell}_t, \boldsymbol{j}_t) \in P_{s,\boldsymbol{k}_t}$ in line 6. For local cluster $(\boldsymbol{\ell}_t, \boldsymbol{j}_t)$ itself, the local server updates the local statistics and initializes the upload buffer using the newly generated perturbation in lines 8 to 10. For the download protocol, at each time step t, the global server will check the deviation between global statistics and the local statistics via following condition:

$$\det(S_{t,\boldsymbol{k}_t}^g)/\det(S_{t,j}^\ell) \geq D \tag{6}$$

independently for local clusters (ℓ, j) in global cluster $P_{s,\boldsymbol{k}_t}$. If any cluster (ℓ, j) satisfies such condition, the global server sends the information from other clusters to (ℓ, j), which is used to update (ℓ, j)'s local statistics. Finally, the global server cleans the download buffer.

Tree-based privacy protocol. To ensure the uploaded information are privatized, we adopt the tree-based privatizer to generate random perturbations $H_{t,j}^\ell$ and $h_{t,j}^\ell$ whenever an upload event happens. Note that the privatier subroutine is at the local cluster level and a new privatizer is created if the cluster structure changes at the beginning of any phase s. Let $x_1, ..., x_T$ be a (matrix-valued) sequence of length T, and $s_i = \sum_{t=1}^i x_i$ be the partial sum of the first i elements that will be realised privately. Generally speaking, the tree-based mechanism Dwork et al. [2006] maintains a binary tree $\mathcal{T}$ of depth $1 + \lceil \log T \rceil$, where the leaf nodes contain the elements x_i and the parent node maintains the sum of its children. For each node with value n_i, the tree-base mechanism protects privacy by adding noise h_i to each node and release $n_i + h_i$ if queried. The key advantage is that such a tree only accesses $\nu = O(\log T)$ nodes to compute and release the partial sum s_i, which means the perturbation is at most $O(\nu)$ instead of $O(T)$.

Following this general idea, we implement the tree-based privatizer (ℓ, j) that satisfies the requirements of CDP. Recall that we only need to protect the information uploaded to the global server, it suffices to maintain a tree $\mathcal{T}_j^\ell$ of depth $\nu = O(1 + \lceil \log t_c \rceil)$, where t_c is the total number of uploads. To make the partial sums private, we insert a random noise matrix to each node in $\mathcal{T}_j^\ell$. Specifically, we sample a random matrix $\bar{N} \in \mathbb{R}^{(d+1)\times(d+1)}$ where each entry $\bar{N}_{p,q}$ is drawn from i.i.d. Gaussian distribution $\mathcal{N}(0, \sigma_{\text{noise}})$ and symmetrize it to get $N = (\bar{N}^\top + \bar{N})/\sqrt{2}$. It follows that in order to ensure the whole tree is (ε, δ)-DP, each node should preserve $(\varepsilon/\sqrt{8\nu \log(2/\delta)}, \delta/2)$-DP. In other words, it suffices to set the variance $\sigma_{\text{noise}} = 64\nu \log(2/\delta)^2/\epsilon^2$. Note that at each upload round t, the total noise added to the partial sum is the summation of at most ν random matrices, where the top-left $(d\times d)$-submatrix forms $H_{t,j}^\ell$ and the first d elements from the right-most $(d+1)\times 1$ vector forms $h_{t,j}^\ell$. By concentration, we have with probability at least $(1 - \frac{\alpha}{mL})$, the operator norm of $H_{t,j}^\ell$ is

$$\left\|H_{t,j}^\ell\right\|_{op} \leq \rho \triangleq 8\sqrt{2}\nu \log(4/\delta)(4\sqrt{d} + 2\log(2mL/\alpha))/\varepsilon. \tag{7}$$

for any $\ell \in [L], j \in [m], t \in [T]$.

Recommendation Procedure. At each time step t, the recommended action a_t for user $\boldsymbol{i}_t$ is selected as follows. When the current cluster is correct (which is guaranteed after $O(\log T)$ rounds and to be proved later), the estimated $\hat{\theta}_t = (S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})^{-1} u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$ is computed using the local information $S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$ and $u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$. Since by Lemma 4, $\left\|\theta_{\boldsymbol{i}_t} - \hat{\theta}_t\right\|_2 \leq \beta_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}$, the confidence radius is $\beta_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} \|x_{i,t}\|_{(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})^{-1}}$, which character-

Algorithm 2 Phase-based Cluster Detection and Adjustment

$t = 2^s - 1$.
for $\ell \in [L]$ **do**
  Set $\mathcal{G}_s^\ell$ by deleting any edge $(i_1, i_2) \in \mathcal{G}_{s-1}^\ell$ if Equation (3) holds.
  For $\ell \in [L], j \in C(\mathcal{G}_s^\ell)$, generate new perturbation $H_{t,j}^\ell, h_{t,j}^\ell$ using PVT(ℓ, j) in Algorithm 5 and set historical $\bar{H}_{t,j}^\ell = H_{t,j}^\ell, h_{t,j}^\ell = \bar{h}_{t,j}^\ell$.
  For $\ell \in [L]$, upload the local clustered information $I_{s,\ell} = (C_{s,j}^\ell, \tilde{V}_{s,j}^\ell, \tilde{b}_{s,j}^\ell, \tilde{T}_{s,j}^\ell)_{j \in C(\mathcal{G}_s^\ell)}$ to the global server, where $(\tilde{V}_{s,j}^\ell, \tilde{b}_{s,j}^\ell, \tilde{T}_{s,j}^\ell) = (2\rho I + H_{t,j}^\ell, h_{t,j}^\ell, 0) + \sum_{i \in C_{s,j}^\ell} (V_{t,i}, b_{t,i}, T_{t,i})$.
end for
The global server does global merge based on I_s and get m_s global clusters $P_s = \{P_{s,1}, ..., P_{s,m_s}\}$, where the two local clusters $C_{t,j_1}^{\ell_1}$, $C_{t,j_2}^{\ell_2}$ (with $\ell_1 \neq \ell_2$) are merged together in $P_{s,k}$ if Equation (4) holds.
if $s = 0$ or $P_s \neq P_{s-1}$ **then**
  //Renew the cluster information.
  for $k \in [m_s]$ **do**
    Set global gram matrix $(S_{t,k}^g, u_{t,k}^g, T_{t,k}^g) = \sum_{(\ell,j) \in P_{s,k}} (\tilde{V}_{s,j}^\ell, \tilde{b}_{s,j}^\ell, \tilde{T}_{s,j}^\ell)$.
    for $(\ell, j) \in P_{s,k}$ **do**
      Set $(S_{t,j}^\ell, b_{t,j}^\ell, T_{t,j}^\ell) = (S_{t,k}^g, b_{t,k}^g, T_{t,k}^g)$.
      Create new perturbation $H_{t,j}^\ell, h_{t,j}^\ell$ using PVT(ℓ, j) in Algorithm 5.
      Set new $(\Delta S_{t,j}^\ell, \Delta u_{t,j}^\ell, \Delta T_{t,j}^\ell) = (3\rho I + H_{t,j}^\ell - \bar{H}_{t,j}^\ell, h_{t,j}^\ell - \bar{h}_{t,j}^\ell, 0)$
      Set new $(\Delta S_{t,j}^{-\ell}, \Delta u_{t,j}^{-\ell}, \Delta T_{t,j}^{-\ell}) = (0, 0, 0)$.
    end for
  end for
end if

izes the exploration bonus for item $i \in [K]$.

4 RESULTS

Recall that perturbations $\bar{H}_{t,j}^\ell, \bar{H}_{t,j}^\ell$ are designed to satisfy the $(\varepsilon, \delta, L, m)$-CDP requirement. In particular, the privacy budget (ε, δ) affects the regret and communication bounds via the following quantities $(\rho_{\max}, \rho_{\min}, \kappa)$, which can be treated as spectral bounds for $\bar{H}_{t,j}^\ell, \bar{H}_{t,j}^\ell$. Let $\tilde{H}_{t,j}^\ell = 2\rho I + 3\rho c_{j,t}^\ell I + \bar{H}_{t,j}^\ell$, where $c_{j,t}^\ell$ is the number of uploads for local server ℓ and cluster j.

Definition 2 (Approximately-accurate $\rho_{\min}, \rho_{\max}$ and κ)**.** *The bounds $0 < \rho_{t,\min} \leq \rho_{t,\max}$ and $\kappa > 0$ are $(\alpha/(mL))$-accurate for $(\bar{H}_{t,j}^\ell)$ for any $\ell \in [L], j \in [m]$ and $t \in [T]$:*

$$\left\|\tilde{H}_{t,j}^\ell\right\|_{op} \leq \rho_{\max}, \left\|(\tilde{H}_{t,j}^\ell)^{-1}\right\|_{op} \leq \frac{1}{\rho_{\min}}, \left\|\bar{h}_{t,j}^\ell\right\|_{(\tilde{H}_{t,j}^\ell)^{-1}} \leq \kappa \tag{8}$$

with probability at least $(1 - \frac{\alpha}{mL})$. As we will later show, our communication ensures $c_{j,t}^\ell \in [0, d \log T / \log(\min\{U, D\})]$, so $\rho_{\min} = \rho$, $\rho_{\max} = 3\rho + 3\rho d \log T / \log(\min\{U, D\})$, and $\kappa = \left\|\bar{h}_{t,j}^\ell\right\| / \sqrt{\rho}$, where $\rho \triangleq 8\sqrt{2}\nu \log(4/\delta)(4\sqrt{d} + 2\log(2mL/\alpha))/\varepsilon$ is given by our privatizer.

Algorithm 3 Check Upload Event

Update upload buffer $(\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta \boldsymbol{T}_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) \mathrel{+}= (x_{t,a_t} x_{t,a_t}^\top, y_t x_{t,a_t}, 1)$.
if $\det(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} + \Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) / \det(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) \geq U$ **then**
  The global cluster finds $\boldsymbol{k}_t$ so that $(\boldsymbol{\ell}_t, \boldsymbol{j}_t) \in P_{s,\boldsymbol{k}_t}$.
  Update global information $(S_{t,\boldsymbol{k}_t}^g, u_{t,\boldsymbol{k}_t}^g, T_{t,\boldsymbol{k}_t}^g) \mathrel{+}= (\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})$.
  for $(\ell, j) \neq (\boldsymbol{\ell}_t, \boldsymbol{j}_t) \in P_{s,\boldsymbol{k}_t}$ **do**
    Global server updates other servers' download buffer $(\Delta S_{t,j}^{-\ell}, \Delta u_{t,j}^{-\ell}, \Delta T_{t,j}^{-\ell}) \mathrel{+}= (\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})$.
  end for
  Local server $\boldsymbol{\ell}_t$ updates the local statistics: $(S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) \mathrel{+}= (\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t})$.
  Local server $\boldsymbol{\ell}_t$ sets $(\bar{H}_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \bar{h}_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) = (H_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, h_{t,j}^{\boldsymbol{\ell}_t})$ and creates new perturbation $H_{t,\boldsymbol{j}_t}^\ell, h_{t,\boldsymbol{j}_t}^\ell$ using the tree-based privatizer PVT$(\boldsymbol{\ell}_t, \boldsymbol{j}_t)$.
  Local server $\boldsymbol{\ell}_t$ initializes the upload buffer using the new perturbation: $(\Delta S_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta u_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, \Delta T_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}) = (3\rho I + H_{t,j}^{\boldsymbol{\ell}_t} - H_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, h_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t} - \bar{h}_{t,\boldsymbol{j}_t}^{\boldsymbol{\ell}_t}, 0)$.
end if

Algorithm 4 Check Download Event

for $(l, j) \in P_{s,\boldsymbol{k}_t}$ **do**
  if $\det(S_{t,\boldsymbol{k}_t}^g) / \det(S_{t,j}^\ell) \geq D$ **then**
    Local server receives $(S_{t,j}^\ell, u_{t,j}^\ell, T_{t,j}^\ell) \mathrel{+}= (\Delta S_{t,j}^{-\ell}, \Delta u_{t,j}^{-\ell}, \Delta T_{t,j}^{-\ell})$.
    Global server cleans the download buffer: $(\Delta S_{t,j}^{-\ell}, \Delta u_{t,j}^{-\ell}, \Delta T_{t,j}^{-\ell}) = (0, 0, 0)$.
  end if
end for

In the following, we will give general regret and communication bounds using $(\rho_{\max}, \rho_{\min}, \kappa)$ and replace them with their exact values.

4.1 REGRET BOUND

We give the following theorem as our main result for the regret bound.

Theorem 1. *Suppose the cluster structure over the users and items satisfy the assumptions in Section 2 with gap parameter $\gamma > 0$ and item regularity parameter $1 \geq \lambda_x > 0$. If the privatizer produces random perturbation that are $(1/(8mLT))$-accurate as in Definition 2, the regret is bounded with probability by*

$$R(T) \leq \tilde{O}\Big(n\big(\frac{\log T}{\lambda_x^2} + \frac{\sigma_0^2 d \log T}{\lambda_x \gamma^2} + \frac{\log(1/\delta)\log T}{\lambda_x \varepsilon \gamma^2}\big) + dL\sqrt{mT\frac{\log(1/\delta)}{\varepsilon}}\log^{1.5} T\Big)$$

We will give the proof sketch for the above theorem 1.

Algorithm 5 Privatizer PVT(ℓ, j) for cluster j at server ℓ

1: **Input:** Privacy budget ε, δ, number of uploads t_c.
2: Create a binary tree $\mathcal{T}$ of depth $\nu = \lceil \log(t_c + 1) \rceil + 1$.
3: For each node, we generate a perturbation matrix matrix $N \in \mathbb{R}^{(d+1)\times(d+1)}$, where $N = (\bar{N} + \bar{N}^\top)/\sqrt{2}$ and $\bar{N} \in \mathbb{R}^{(d+1)\times(d+1)}$ with $\bar{N}_{p,q} \sim \mathcal{N}(0, 64\nu \frac{\log(2/\delta)^2}{\varepsilon^2})$.
4: Calculate a queue of $\mathcal{Q} = (H_i, h_i)_{i=1,\dots,t_c+1}$ for partial sums $s_0, ..., s_{t_c}$.
5: Sequentially pop one pair of $\mathcal{Q}$ if PVT(ℓ, j) is called.

Proof. Our proof mainly consists of two parts. The first part bounds the number of exploration rounds $2T_0$ after which the overall user clusters are correctly detected at the global server. The second part is to bound the regret for the asynchronous contextual linear bandits after the clusters are partitioned correctly.

Different from standard online clustering bandits, the key technical challenge is to take care of the additional random Gaussian noise produced by the privater, which perturbs the true observation that is needed for global cluster detection and the regret analysis for contextual linear bandits. Moreover, such perturbed observation are also lagged behind the instant observation, since FCLUB-CDP adopts the "delayed" asynchronous communication where upload and download are triggered occasionally. This makes standard contextual bandit analysis no longer works and requires new proof techniques to handle the gap between instant observation and the lagged (and perturbed) observation.

For the first cluster detection part, by the assumption of item regularity, we prove that after $t \geq O(n(\frac{\log T}{\lambda_x^2} + \frac{d\sigma_0^2 \log T}{\lambda_x \gamma^2}))$ rounds, the local estimates are accurate enough so that the local clusters are correctly identified, similar to that of Li and Zhang [2018]. Specifically, the 2-norm distance between local estimate $\hat{\theta}_{t,i}$ and the truth θ_i for any user i is less than $\gamma/4$. Thus the local clusters are split correctly for all local servers. Now for the global cluster detection, the global server receives the aggregated observation from correctly partitioned local clusters, in which random Gaussian noises are added. Based on spectra property of Gaussian noise matrices (definition 2), the global server will spend additional $O(\frac{n \log(1/\delta) \log T}{\lambda_x \varepsilon \gamma^2})$ rounds so that the perturbed estimate $\hat{\theta}^l_{s,j}$ are accurate enough at the beginning of phase $s = \lceil \log_2 T_0 \rceil$, where $T_0 = O(n(\frac{\log T}{\lambda_x^2} + \frac{d\sigma_0^2 \log T}{\lambda_x \gamma^2} + \frac{\log(1/\delta) \log T}{\lambda_x \varepsilon \gamma^2}))$. Therefore, after $t > 2T_0$, the overall user clusters are partitioned correctly.

For the regret after $2T_0$, we use the delayed update technique from [Abbasi-Yadkori et al., 2011, Section 5.1], which only recomputes the confidence radius only $O(\log T)$ times and hence saves computation. The same strategy can also be applied for the delayed communication. The key analysis relies on using the upload and download condition in eq. (5) and eq. (6), so that the actually-used cluster confidence radius is at most Γ times larger than that if all local servers upload their perturbed observations in a fully synchronized manner, where $\Gamma = \sqrt{D(1 + (L-1)(U-1)) + U - 1}$. This will give a $\Gamma R_j(T_j)$ regret for the second part, where $R_j(T_j)$ is the private-version regret for the cluster j if all observation are synchronized at each round. The full proof is put in Appendix A.1. ■

4.2 COMMUNICATION COST

We give the following theorem to bound the total communication cost.

Theorem 2. *Under the CDP setting, the total communication cost satisfies:*

$$C_T \leq O(\frac{dmL \log T}{\log(\min\{U, D\}})$$

Proof. The total communication cost also has two parts: the upload at the beginning of each phase for global cluster detection and the asynchronous communication within each phase for information sharing. For the first part, the algorithm has at most $\log T$ phases and at each phase, there are total mL local clusters uploading the clustered information, hence the total communication cost is $O(mL \log T)$. For the second part, recall that we adopt the delayed asynchronous communication protocol and the total number of uploads and downloads can be bounded by $O(dmL \log T)$. See Appendix A.2 for the detailed proofs. ■

4.3 PRIVACY GUARANTEE

Theorem 3. *Algorithm 1 preserves $(\varepsilon, \delta, L, m)$-CDP as defined in Definition 1.*

Proof. The CDP condition is satisfied by assigning the right amount of Gaussian noise in each tree node of our tree-based privacy protocol in Section 3. ■

4.4 DISCUSSION AND COMPARISON

For the regret bound, our result has two terms: the regret before the clusters are correctly partitioned $n(\frac{\log T}{\lambda_x^2} + \frac{\sigma_0^2 d \log T}{\lambda_x \gamma^2} + \frac{\log(1/\delta) \log T}{\lambda_x \varepsilon \gamma^2})$ and the regret after the clusters are correctly partition $O(dL\sqrt{mT \frac{\log(1/\delta)}{\varepsilon}} \log^{1.5} T)$. We will compare our results with several degenerate cases given we are the first work to study the federated clustering of bandits setting. For these cases, the additional CDP causes at most $O(\sqrt{\frac{\log(1/\delta)}{\varepsilon}})$ factor and asynchronous communication protocol causes at most $O(\sqrt{dL \log T})$ factor in general.

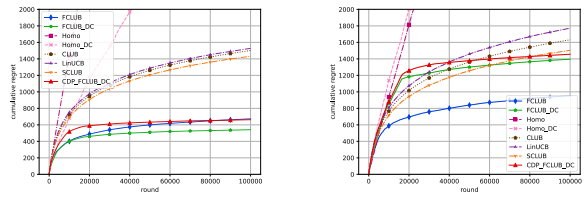
First, when $m = 1, L = 1$, our setting degenerates to the linear bandits with DP where all users share the same underlying parameter. Compared to Shariff and Sheffet [2018] which gives a $O(\sqrt{d \frac{\log(1/\delta)}{\varepsilon}} \sqrt{T} \log^{1.5} T)$ regret

with 0 communication, our bound has a $O(\sqrt{d})$ additional factor (or more precisely $O(\sqrt{d \log \log T})$ factor) for the second term, which stems from the larger perturbation in order to protect total $O(d \log T)$ communication rounds. Second, when $L = 1$, our setting reduces to the online clustering bandits with DP, Li and Zhang [2018] gives a $O(n(\frac{\log T}{\lambda_x^2} + \frac{d\sigma_0^2 \log T}{\lambda_x \gamma^2}) + d\sqrt{mT} \log T)$ for the non-DP version. Since CDP mechanism requires random perturbation, the clustering process suffers an additional $\frac{n \log T \log(1/\delta)}{\lambda_x \varepsilon \gamma^2}$ for the first term and the second regret term now has a new $\sqrt{\frac{\log(1/\delta) \log T}{\varepsilon}}$ leading factor due to the CDP requirements. Third, when $m = 1$ and if we consider the special case when each local server only has one user and all users come in a round-robin manner, our setting reduces to the distributed linear bandits with DP. Dubey and Pentland [2020] provides a synchronized algorithm that achieves $O(\sqrt{dLT \frac{\log(1/\delta)}{\varepsilon}} \log^{1.5} T)$, our second term has an additional $\sqrt{dL}$ factor because of different communication protocol, which enables asynchronous communication at the cost of the larger $O(dL \log T)$ (compared with $O(L \log T)$) communication rounds and an additional $\Gamma = O(\sqrt{L})$ factor in the confidence radius.

Our communication cost also has two terms: the first $O(mL \log T)$ term for identifying clusters at the beginning of each phase and the leading $O(\frac{dmL \log T}{\log(\min\{U,D\}})$ term for our asynchronous communication protocol. Compared with Dubey and Pentland [2020] when $m = 1$ and users come at the round-robin manner, our communication has an additional $O(d)$ factor. Due to the specialty of the user arrival, the same paper can achieve communication cost independent of T at the cost of $O(\log(LT))$ additional factor in the regret. Though our total communication cost can not be reduced below $O(mL \log T)$ due to the first term, it will be interesting to consider whether the similar trade-off works for our asynchronous protocol in the future work.

5 EXPERIMENTS

To validate our theoretical findings, we conduct experiments a synthetic dataset and a real-world MovieLens dataset. Algorithm 1 is denoted as FCLUB-CDP-DC and we also consider its non-private version FCLUB-DC and its non-private, synchronized version FCLUB (by instantly uploading the observations). The benchmark algorithms include CLUB which uses a separate CLUB Gentile et al. [2014] for each local server, SCLUB which uses a separate SCLUB Li et al. [2019] for each server and LinUCB which uses a separate for each user. We also consider the synchronized and asynchronous algorithm (denoted as Homo and Homo-DC) by treating users are identical with the same preference vector. Note that all results are averaged over 10 random seeds and we provide mean result with one standard derivation for each curve. Due to the space limit, we provide the detailed experimental settings (including data generation and processing) in Appendix A.4.1, the parameter study Appendix A.4.2, the communication cost Appendix A.4.3 and running time results in Appendix A.4.4, respectively.



(a) Comparison with baselines on the synthetic dataset (b) Comparison with baselines on the Movielens dataset

Figure 2: Comparative experiment with baselines

5.1 SYNTHETIC DATASET

We first experiment on a synthetic dataset. In Figure 2a, we compare our algorithm FCLUB-DC-CDP with the baselines listed above. The vertical axis indicates the cumulative regret and the horizontal axis indicates the round t. In Homo-DC, we assume all users are in the same cluster, we can verify that our algorithm FCLUB-DC-CDP can complete cluster detection well by comparing it with Homo-DC. Besides, since CLUB and SCLUB only perform merge or split operations at the local cluster level, through comparison with SCLUB and CLUB we can verify the correctness of our algorithm's merge operation at the global level. In general, our algorithm FCLUB-DC-CDP's performance has a clear advantage over baseline SCLUB, CLUB, LinUCB, Homo and Homo-DC. FCLUB-DC-CDP performs worse than FCLUB and FCLUB-DC due to the delay communication and cluster differential privacy(CDP).

5.2 MOVIELENS DATASET

In this section, we compare our algorithm FCLUB-DC-CDP with the baselines listed above on movie recommendations with MovieLens dataset. The performances are shown in Figure 2b. Our algorithm FCLUB-DC-CDP's performance has a advantage over baseline SCLUB, CLUB, LinUCB, Homo and Homo-DC in general. Figure 2b shows FCLUB-DC-CDP performs worse than FCLUB and FCLUB-DC due to the delay communication and cluster differential privacy(CDP) as we have explained in synthetic dataset part. In the early stage, our algorithm needs more time to identify the underlying cluster structure. But after all user clusters are correctly detected at the global server, our algorithm performs better than LinUCB-one that assume users are homogeneous (Homo-DC), CLUB/SCLUB on each local server and LinUCB-ind on each user.

6 FUTURE WORK

There are many compelling directions for the future study. For example, it would be interesting to study our problem where local differential privacy is considered. One could also study the more efficient protocol to further save the communication cost.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29 (4):731–745, 2011.

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

Ilai Bistritz and Amir Leshem. Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems*, 31, 2018.

Etienne Boursier and Vianney Perchet. Sic-mmab: synchronisation involves communication in multiplayer multi-armed bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

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

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765. PMLR, 2014.

Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *International Conference on Machine Learning*, pages 1253–1262. PMLR, 2017.

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*, 14(1–2):1–210, 2021.

Nathan Korda, Balazs Szorenyi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*, pages 1301–1309. PMLR, 2016.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Chuanhao Li and Hongning Wang. Asynchronous upper confidence bound algorithms for federated linear bandits. *arXiv preprint arXiv:2110.01463*, 2021.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Shuai Li and Shengyu Zhang. Online clustering of contextual cascading bandits. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

Ye Li, Hong Xie, Yishi Lin, and John CS Lui. Unifying offline causal inference and online bandit learning for data driven decision. In *Proceedings of the Web Conference 2021*, pages 2291–2303, 2021.

Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*, pages 155–163. PMLR, 2016.

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

Chengshuai Shi and Cong Shen. Federated multi-armed bandits. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Xiaoying Zhang, Hong Xie, Hang Li, and John CS Lui. Conversational contextual bandit: Algorithm and application. In *Proceedings of The Web Conference 2020*, pages 662–672, 2020.

Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. Locally differentially private (contextual) bandits learning. *arXiv preprint arXiv:2006.00701*, 2020.

Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pages 3–4, 2021.

SUPPLEMENTARY MATERIAL

A THEORETICAL ANALYSIS

A.1 REGRET

Let $\beta_i(T_{t,i},\alpha)=\sigma_0\sqrt{2\log(\frac{1}{\alpha})+d\log(1+\frac{T_{t,i}}{\lambda d})}+\sqrt{\lambda}$, $\beta_j^\ell(T_{t,j}^\ell,0,\alpha)=\sigma_0\sqrt{2\log(\frac{1}{\alpha})+d\log(3+\frac{T_{t,j}^\ell}{\rho_{\min}d})}+\sqrt{\rho_{0,\max}}+\kappa$, $\beta_j^\ell(T_{t,j}^\ell,L,\alpha)=\sigma_0\sqrt{2\log(\frac{1}{\alpha})+d\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{T_{t,j}^\ell}{\rho_{\min}d})}+\sqrt{L\rho_{\max}}+\sqrt{L}\kappa$. We define a series of events as follows which will be helpful for the proof.

$\mathcal{B}_0(\alpha)=\{(\rho_{\min},\rho_{\max},\kappa)$ are $\alpha/(mL)$-accurate bounds for $\tilde{H}_{t,j}^\ell,$ for any $t\in[T],j\in[m],\ell\in[L]\}$.

$\mathcal{B}_1(\alpha)=\{\left\|\theta_i-\hat{\theta}_{t,i}\right\|_{M_{t,i}}\le\beta_i(T_{t,i},\alpha/n)$ for any $t\in[T],i\in[n]\}$.

$\mathcal{B}_2(\alpha)=\{\lambda_{\min}(V_{t,i})\ge T_{t,i}\lambda_x/8,$ for all $T_{t,i}\ge\frac{1024}{\lambda_x^2}\log\frac{512d}{\lambda_x^2\alpha/n},$ for any $i\in[n]\}$.

$\mathcal{B}_3(\alpha)=\{\left\|\theta_j-\hat{\theta}_{t,j}^\ell\right\|_{\tilde{V}_{t,j}^\ell}\le\beta_j^\ell(\tilde{T}_{t,j}^\ell,0,\alpha/(mL))$ for any $t\in[T],j\in[m],\ell\in[L]\}$.

$\mathcal{B}_4(\alpha)=\{\left\|\theta^*-\hat{\theta}_{t,j}^\ell\right\|_{S_{t,j}^\ell}\le\beta_j^\ell(\tilde{T}_{t,j}^\ell,L,\alpha/(mL))$ for any $t\in[T],j\in[m],\ell\in[L]\}$.

For each of event $B_i\alpha,i=0,...,4$, the probability $\Pr\{B_i(\alpha)\}\ge1-\alpha$, where $B_0(\alpha)$ is by the definition in Equation (8), $B_1(\alpha)$ is by [Abbasi-Yadkori et al., 2011, Theorem 2], $B_2(\alpha)$ is by [Li and Zhang, 2018, Lemma 7], $B_3(\alpha)$ and $B_4(\alpha)$ are by Lemma 4. The detailed proofs are postponed to later sections.

A.1.1 Correctness of the Cluster Detection

Lemma 1. *Let* $\bar{T}_i(\alpha)=\max\Big\{A(\alpha),B,C,D(\alpha),E(\alpha)\Big\}$, $A(\alpha)=\frac{(96\sigma_0)^2\log(\frac{2}{\alpha})}{\lambda_x\gamma^2}$, $B=\frac{(96\sigma_0)^2d}{\lambda_x\gamma^2}\log(\frac{4608\sigma_0^2}{\lambda_x\gamma^2})$, $C=\frac{192^2\times6\sqrt{2}\nu(\log(\frac{4}{\delta}))\sqrt{d}}{\varepsilon\lambda_x\gamma^2}$, $D(\alpha)=\frac{192^2\times3\sqrt{2}\nu(\log\frac{4}{\delta})\log(\frac{2mL}{\alpha})}{\varepsilon\lambda_x\gamma^2}$, $E(\alpha)=\frac{1024}{\lambda_x^2}\log\frac{512nd}{\lambda_x^2\alpha}$. *When* $T>2T_0(\alpha)\triangleq2\times(16n\log(\frac{T}{\alpha})+4n\bar{T}_i(\alpha))$, *then with probability* $1-6\alpha$, *(a): All local clusters are partitioned correctly;*

(b) All local clusters are correctly merged at the global server.

Proof. For any user i, it suffices to show that when $T_{t,i}$ is larger than some threshold $\bar{T}_i(\alpha)$ (whose value will be settled later), then we have the following two results hold.

(a): Under event $\mathcal{B}_1(\alpha)$ and $\mathcal{B}_2(\alpha)$, which happens at least $1-2\alpha$,

$$\left\|\theta_i-\hat{\theta}_{t,i}\right\|_2\le\frac{\left\|\theta_i-\hat{\theta}_{t,i}\right\|_{M_{t,i}}}{\sqrt{\lambda_{\min}(M_{t,i})}}\le\frac{\beta_i(T_{t,i},\alpha/n)}{\sqrt{\lambda+T_{t,i}\lambda_x/8}}\le\frac{\sigma_0\sqrt{2\log(\frac{n}{\alpha})+d\log(1+\frac{T_{t,i}}{\lambda d})}+\sqrt{\lambda}}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{4}\tag{9}$$

where $M_{t,i}=\lambda I+V_{t,i}$ and the last inequality is valid when $T_{t,i}\ge\bar{T}_i(\alpha)$.

When Equation (9) holds, for any two user i_1,i_2 who belong to different clusters, i.e., $\|\theta_{i_1}-\theta_{i_2}\|_2\ge\gamma$, $\left\|\hat{\theta}_{i_1}-\hat{\theta}_{i_1}\right\|_2\ge\|\theta_{i_1}-\theta_{i_2}\|_2-\left\|\hat{\theta}_{i_2}-\theta_{i_2}\right\|_2-\left\|\hat{\theta}_{i_1}-\theta_{i_1}\right\|_2>\gamma/2\ge\frac{\beta_i(T_{t,i_1},\alpha/n)}{\sqrt{\lambda+T_{t,i_1}\lambda_x/8}}+\frac{\beta_i(T_{t,i_2},\alpha/n)}{\sqrt{\lambda+T_{t,i_2}\lambda_x/8}}$, which will trigger the condition in Line 3 in Algorithm 2. On the other hand, when the condition in Line 3 in Algorithm 2 holds, $\frac{\beta_i(T_{t,i_1},\alpha/n)}{\sqrt{\lambda+T_{t,i_1}\lambda_x/8}}+\frac{\beta_i(T_{t,i_2},\alpha/n)}{\sqrt{\lambda+T_{t,i_2}\lambda_x/8}}<\left\|\hat{\theta}_{i_1}-\hat{\theta}_{i_1}\right\|_2\le\|\theta_{i_1}-\theta_{i_2}\|_2+\left\|\hat{\theta}_{i_2}-\theta_{i_2}\right\|_2+\left\|\hat{\theta}_{i_1}-\theta_{i_1}\right\|_2\le\|\theta_{i_1}-\theta_{i_2}\|_2+\frac{\beta_i(T_{t,i_1},\alpha/n)}{\sqrt{\lambda+T_{t,i_1}\lambda_x/8}}+\frac{\beta_i(T_{t,i_2},\alpha/n)}{\sqrt{\lambda+T_{t,i_2}\lambda_x/8}}$, which implies $\|\theta_{i_1}-\theta_{i_2}\|_2>0$. In other words, all local clusters are correct after $T_{t,i}\ge\bar{T}_i(\alpha)$ for all users.

(b): Moreover, under event $\mathcal{B}_0(\alpha),\mathcal{B}_2(\alpha)$ and $\mathcal{B}_3(\alpha)$, for any correctly partitioned local cluster C_j at server ℓ with true

parameter θ_j, with probability at least $1-3\alpha$,

$$\left\|\theta_j-\hat{\theta}^{\ell}_{t,j}\right\| \leq \frac{\left\|\theta_j-\hat{\theta}^{\ell}_{t,j}\right\|_{\tilde{V}^{\ell}_{t,j}}}{\sqrt{\lambda_{\min}(\tilde{V}^{\ell}_{t,j})}} \leq \frac{\beta^{\ell}_j(\tilde{T}^{\ell}_{t,j},0,\alpha/(mL))}{\sqrt{\rho_{\min}+\tilde{T}^{\ell}_{t,j}\lambda_x/8}} \leq \frac{\sigma_0\sqrt{2\log(\frac{mL}{\alpha})+d\log(3+\frac{\tilde{T}^{\ell}_{t,j}}{\rho_{\min}d})}+\sqrt{\rho_{0,\max}}+\kappa_t}{\sqrt{\tilde{T}^{\ell}_{t,j}\lambda_x/8}} < \frac{\gamma}{4} \tag{10}$$

, where $\tilde{T}^{\ell}_{t,j}=\sum_{i\in C_j}T_{t,i}$ and $\hat{\theta}_{t,j}=(\tilde{V}^{\ell}_{t,j})^{-1}\tilde{b}^{l}_{t,j}$.

Here $\beta_{\tilde{T}^{\ell}_{t,j}}$ is irrelevant to L or m because the uploaded information only contains the local observations and local noise added at the beginning of each phase, but doesn't contain any information from other local servers. According to similar argument, we can show for any two cluster j_1, j_2 who belong to the same cluster, i.e., $\|\theta_{j_1}-\theta_{j_2}\|_2=0$, we have $\left\|\hat{\theta}_{j_1}-\hat{\theta}_{j_2}\right\|_2 \leq \frac{\beta^{\ell}_j(\tilde{T}^{\ell}_{t,j_1},0,\alpha/(mL))}{\sqrt{\rho_{\min}+\tilde{T}^{\ell}_{t,j_1}\lambda_x/8}}+\frac{\beta^{\ell}_j(\tilde{T}^{\ell}_{t,j_2},0,\alpha/(mL))}{\sqrt{\rho_{\min}+\tilde{T}^{\ell}_{t,j_2}\lambda_x/8}}$, which will trigger the condition in Line 7 in Algorithm 2. On the other hand, if the above condition is triggered, then it implies $\|\theta_{j_1}-\theta_{j_2}\|_2<\gamma$. In other words, all global clusters are merged correctly.

Now for the last inequality of Equation (9) and Equation (10), we only need to prove the following inequality holds for any $i\in[n]$,

$$\frac{\sigma_0\sqrt{2\log(\frac{n}{\alpha})+d\log(3+\frac{T_{t,i}}{d})}+\sqrt{\rho_{0,\max}}+\kappa_t}{\sqrt{T_{t,i}\lambda_x/8}} \leq \frac{\gamma}{4} \tag{11}$$

This comes from the observation that $\lambda_{0,\max}>\lambda\geq 1$ and $\kappa_t>0$ for Equation (9) and $n\geq mL, \lambda_{0,\min}\geq 1, \tilde{T}^{\ell}_{t,j}=\sum_{i\in C_j}T_{t,i}$ for Equation (10).

We prove the sufficient $\bar{T}_i(\alpha)$ as shown in Lemma 2 so that Equation (11) holds in Lemma 2. Now by [Li and Zhang, 2018, Lemma 8], at global time $T_0(\alpha)=16n\log(\frac{T}{\alpha})+4n\bar{T}_i(\alpha)$, Equation (11) is correct with probability at least $1-\alpha$. Then at the next cluster detection (which occurs at most two times of $T_0(\alpha)$), the global server will partition the global clusters correctly with probability at least $1-6\alpha$ (by using union bounds of all corresponding events).

By lemma 2 eq. (11) is correct when $T_{t,i}>\bar{T}_i(\alpha)$, and by [Li and Zhang, 2018, Lemma8] this can be satisfied when $T>16n\log(\frac{T}{\alpha})+4n\bar{T}_i(\alpha)$. Then at next phase, (b) in the lemma 1 should be satisfied.

■

Lemma 2.

$$\frac{\sigma_0\sqrt{2\log(\frac{n}{\alpha})+d\log(3+\frac{T_{t,i}}{d})}+\sqrt{\rho_{0,\max}}+\kappa_t}{\sqrt{T_{t,i}\lambda_x/8}} < \frac{\gamma}{4} \tag{12}$$

can be satisfied when $T_{t,i}>\bar{T}_i(\alpha)$*, where* $\bar{T}_i(\alpha)=\max\left\{A(\alpha),B,C,D(\alpha),E(\alpha)\right\}$, $A(\alpha)=\frac{(96\sigma_0)^2\log(\frac{2}{\alpha})}{\lambda_x\gamma^2}$, $B=\frac{(96\sigma_0)^2d}{\lambda_x\gamma^2}\log(\frac{4608\sigma_0^2}{\lambda_x\gamma^2})$, $C=\frac{192^2\times 6\sqrt{2}\nu(\log(\frac{4}{\delta}))\sqrt{d}}{\varepsilon\lambda_x\gamma^2}$, $D(\alpha)=\frac{192^2\times 3\sqrt{2}\nu(\log\frac{4}{\delta})\log(\frac{2mL}{\alpha})}{\varepsilon\lambda_x\gamma^2}$, $E(\alpha)=\frac{1024}{\lambda_x^2}\log\frac{512nd}{\lambda_x^2\alpha}$.

Proof. We can divide this proof into three parts: $\frac{\sigma_0\sqrt{2\log(\frac{n}{\alpha})+d\log(3+\frac{T_{t,i}}{d})}}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{12}$, $\frac{\sqrt{\rho_{0,max}}}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{12}$ and $\frac{\kappa_t}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{12}$.

The first part can be satisfied by (a)$\frac{\sigma_0\sqrt{2\log(n/\alpha)}}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{24}$ and (b) $\frac{\sigma_0\sqrt{d\log(3+T_{t,i}/d)}}{\sqrt{T_{t,i}\lambda_x/8}}<\frac{\gamma}{24}$. And (a) can be satisfied by $T_{t,i}>\frac{(96\sigma_0)^2\log(\frac{2}{\alpha})}{\gamma^2\lambda_x}$, (b) by [Li and Zhang, 2018, Lemma 9] can be satisfied by $T_{t,i}>\frac{(96\sigma_0)^2d}{\lambda_x\gamma^2}\log(\frac{4608\sigma_0^2}{\lambda_x\gamma^2})$.

Then consider the second part. Since $\rho_{0,max}\leq 24\sqrt{2}\nu\log(\frac{4}{\delta})(4\sqrt{d}+2\log(\frac{2mL}{\alpha}))/\varepsilon$, we can prove this part by (c) $\frac{96\times 8\sqrt{2}\nu\log(\frac{4}{\delta})(\sqrt{d})}{T_{t,i}\varepsilon\lambda_x}<\frac{\gamma^2}{2\times 12^2}$ and (d) $\frac{48\times 8\sqrt{2}\nu\log(\frac{4}{\delta})(\log(\frac{mL}{\alpha}))}{T_{t,i}\varepsilon\lambda_x}<\frac{\gamma^2}{2\times 12^2}$. By some math calculation, (c) can be satisfied by $T_{t,i}>\frac{192^2\times 6\sqrt{2}\nu(\log(\frac{4}{\delta}))\sqrt{d}}{\varepsilon\lambda_x\gamma^2}$ and (d) can be satisfied when $T_{t,i}>\frac{192^2\times 3\sqrt{2}\nu(\log\frac{4}{\delta})\log(\frac{2mL}{\alpha})}{\varepsilon\lambda_x\gamma^2}$.

Finally we prove the third part, since $\kappa_t=\sqrt{2\nu(\sqrt{d}+2\log(2mL/\alpha))/(\sqrt{2}\varepsilon)}$, the third part has a similar form to the second part, so their proof are similar. We can get the third part could be satisfied by $T_{t,i}>\frac{48^2\sqrt{2}\nu\sqrt{d}}{\varepsilon\lambda_x\gamma^2}$ and $T_{t,i}>\frac{48^2\times 2\sqrt{2}\nu\log(\frac{2mL}{\alpha})}{\varepsilon\lambda_x\gamma^2}$.

Considering the $T_{t,i} \geq \frac{1024}{\lambda_x^2} \log \frac{512d}{\lambda_x^2 \alpha/n}$ required by $B_1(\alpha)$, and put all these together the proof is completed. ■

A.1.2 Regret Bound After Clusters are Correctly Detected

Proposition 1. *After $2T_0(\alpha/8)$ rounds, with probability at least $(1-\frac{3\alpha}{4})$, the clusters on both local servers and global servers are correctly partitioned, and the partition won't change in later rounds. Then fix any true global cluster $j \in [m]$ whose local clusters are scattered on $L' \leq L$ servers, let users belong to j appear total T_j times, then with probability $(1-\frac{\alpha}{4})$, the total regret is bounded by*

$$
\begin{aligned}
R_j(T_j) &\leq (\sigma_0\sqrt{2\log(\frac{8mL}{\alpha}) + d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{T_j}{d\rho_{\min}})} + \sqrt{L\rho_{max}} + \kappa\sqrt{L})(\sqrt{d\log(1+\frac{T}{d\rho_{\min}})})\Gamma\sqrt{T_j} \\
&\leq O\Big(\sqrt{L\frac{d\log T}{\log\min\{U,D\}}\frac{\log\log T\log(1/\delta)(\sqrt{d}+\log T)}{\varepsilon}}\Gamma\sqrt{d\log T}\sqrt{T_j}\Big) \\
&\leq \tilde{O}\Big(dL\sqrt{mT\frac{\log(1/\delta)}{\varepsilon}}\log^{1.5}T\Big)
\end{aligned}
$$

where $\Gamma = \sqrt{D(1+(L'-1)(U-1))+U-1}$ and $\sqrt{d} < \log T$.

Proof. We consider the case when all true clusters are correctly identified. Suppose the user belong to global cluster j whose underlying preference vectors are θ^* comes to the server at time slots $\{t_1, ..., t_{T_j}\}$. At time $t = t_k$, user i_t in server ℓ_t and local cluster j_t comes with candidate feature vector sets $(x_{t,1}, ..., x_{t,K})$. Recall that $(\tilde{\theta}_t, a_t) = \arg\max_{(\theta,i)\in\mathcal{C}_t\times[K]} \langle\theta, x_{t,i}\rangle$ is the action selected by our algorithm and $\mathcal{C}_t = \{\theta \in \mathbb{R}^d : \left\|\hat{\theta}_{t,i_t} - \theta\right\|_{S^{\ell_t}_{t,j_t}} \leq \beta^{\ell_t}_{t,j_t}\}$ as defined in lemma 4. Denote the full-synchronized gram matrix from all servers for cluster j as V_t (same for b_t and T_t). Let $x_{t,*} = \arg\max_{i\in[K]} \langle\theta^*, x_{t,i}\rangle$. The instantaneous regret r_t can be written as:

$$
\begin{aligned}
r_{t,j} &= \langle\theta^*, x_{t,*}\rangle - \langle\theta^*, x_{t,a_t}\rangle && (13)\\
&\leq \left\langle\tilde{\theta}_t, x_{t,a_t}\right\rangle - \langle\theta^*, x_{t,a_t}\rangle && (14)\\
&= \left\langle\tilde{\theta}_t - \hat{\theta}_{t,i_t}, x_{t,a_t}\right\rangle + \left\langle\hat{\theta}_{t,i_t} - \theta^*, x_{t,a_t}\right\rangle && (15)\\
&\leq \left\|\tilde{\theta}_t - \hat{\theta}_{t,i_t}\right\|_{(V_t)} \|x_{t,a_t}\|_{V_t^{-1}} + \left\|\hat{\theta}_{t,i_t} - \theta^*\right\|_{V_t} \|x_{t,a_t}\|_{V_t^{-1}} && (16)\\
&= \left\|\tilde{\theta}_t - \hat{\theta}_{t,i_t}\right\|_{S^{l_t}_{t,j_t}} \|x_{t,a_t}\|_{V_t^{-1}} \frac{\left\|\tilde{\theta}_t - \hat{\theta}_{t,i_t}\right\|_{V_t}}{\left\|\tilde{\theta}_t - \hat{\theta}_{t,i_t}\right\|_{S^{l_t}_{t,j_t}}} + \left\|\hat{\theta}_{t,i_t} - \theta^*\right\|_{S^{l_t}_{t,j_t}} \|x_{t,a_t}\|_{V_t^{-1}} \frac{\left\|\hat{\theta}_{t,i_t} - \theta^*\right\|_{V_t}}{\left\|\hat{\theta}_{t,i_t} - \theta^*\right\|_{S^{l_t}_{t,j_t}}} && (17)\\
&\leq 2P_t\beta^{l_t}_{t,j_t} \|x_{t,a_t}\|_{V_t^{-1}} \leq 2\Gamma\beta^{l_t}_{t,j_t} \|x_{t,a_t}\|_{V_t^{-1}}, && (18)
\end{aligned}
$$

where Equation (14) is because the definition of the selected action a_t, Equation (16) is by Cauchy Schwarz inequality and Equation (18) is by Lemma 3.

By summing over all instant regrets, we have

$$
\begin{aligned}
R_j(T_j) &\leq \sum_{t\in[T_j]} r_{t,j} && (19)\\
&\leq 2\Gamma\beta^{\ell_t}_{t,j_t}\left\|x_t, a_t\right\|_{V_t^{-1}} && (20)\\
&\leq 2\Gamma\beta^{\ell}_{j}(T, L, \alpha/(mL))\sqrt{T_j\sum_{t\in[T]}\left\|x_{t,a_t}\right\|^2_{V_t^{-1}}} && (21)\\
&\leq 2\Gamma\beta^{\ell}_{j}(T, L, \alpha/(mL))\sqrt{T_j}\sqrt{d\log(1+\frac{T}{d\rho_{\min}})} && (22)\\
&\leq (\sigma_0\sqrt{2\log(\frac{8mL}{\alpha})+d\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{T_j}{d\rho_{\min}})}+\sqrt{L\rho_{max}}+\kappa\sqrt{L})(\sqrt{d\log(1+\frac{T}{d\rho_{\min}})})\Gamma\sqrt{T_j} && (23)
\end{aligned}
$$

where Equation (21) is by [Shariff and Sheffet, 2018, Lemma 22] and Equation (23) is by lemma 4.

■

Lemma 3. *Let* $P_t = \max_{\theta\neq 0}\sqrt{\frac{\theta^\top V_t\theta}{\theta^\top S^{l_t}_{t,j_t}\theta}}$*, then*

$$
P_t \leq \Gamma \triangleq \sqrt{D(1+(L'-1)(U-1))+U-1} \tag{24}
$$

Proof. Without loss of generality, let the local cluster ids are $(l_1, j_1), ..., (l_{L'}, j_{L'})$ with $(l_t, j_t) = (l_1, j_1)$ and the global gram matrix is S^g_{t,k_t}, we have

$$
\begin{aligned}
\frac{\theta^\top V_t\theta}{\theta^\top S^{l_1}_{t,j_1}\theta} &= \frac{\theta^\top S^g_{t,j_t}\theta+\sum_{k=1}^{L'}\theta^\top\Delta S^{l_k}_{j_k,t}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta} && (25)\\
&= \frac{\theta^\top S^g_{t,k_t}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta}+\frac{\theta^\top\Delta S^{l_1}_{t,j_1}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta}+\sum_{k=2}^{L'}\frac{\theta^\top\Delta S^{l_k}_{t,j_k}\theta}{\theta^\top S^{l_k}_{t,j_k}\theta}\frac{\theta^\top S^{l_k}_{t,j_k}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta} && (26)\\
&\leq \frac{\theta^\top S^g_{t,k_t}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta}+\frac{\theta^\top\Delta S^{l_1}_{t,j_1}\theta}{\theta^\top S^{l_1}_{t,j_1}\theta}+\sum_{k=2}^{L'}\frac{\theta^\top\Delta S^{l_k}_{t,j_k}\theta}{\theta^\top S^{l_k}_{t,j_k}\theta}\frac{\theta^\top S^{l_k}_{t,j_k}\theta}{\theta^\top S^{g}_{t,k_t}\theta} && (27)\\
&\leq \frac{\det(S^g_{t,k_t})}{\det(S^{l_1}_{t,j_1})}+(\frac{\det(S^{l_1}_{t,j_1}+\Delta S^{l_1}_{t,j_1})}{\det(S^{l_1}_{t,j_1})}-1)+\sum_{k=2}^{L'}(\frac{\det(S^{l_k}_{t,j_k}+\Delta S^{l_k}_{t,j_k})}{\det(S^{l_k}_{t,j_k})}-1)\frac{\det(S^{l_k}_{t,j_k})}{\det(S^{g}_{t,k_t})} && (28)\\
&\leq D+(U-1)+(L'-1)D(U-1), && (29)
\end{aligned}
$$

■

where Equation (27) *is due to the fact that* $\sup_{x\neq 0}\frac{x^\top Ax}{x^\top Bx}\leq\frac{\det(A)}{\det(B)}$ *for any positive semi-definite matrices* A, B, C *s.t.* $A = B + C$ *[Abbasi-Yadkori et al., 2011, Lemma 12] and Equation* (28) *is because of the condition of upload Equation* (5) *and download Equation* (6)*.*

Lemma 4. *Under the setting of CDP and when all (global) clusters are correct, fix a local cluster* j *located at the server* ℓ *which shares the information with* L' *clusters (including itself). At time* t*, let the true preference vector be* θ^* *and the true cluster be* j^**, let* $\hat{\theta}^\ell_{t,j} = (S^\ell_{t,j})^{-1}u^\ell_{t,j}$*, then with probability at least* $(1-2\alpha)$*,*

$$
\left\|\theta^*-\hat{\theta}^\ell_{t,j}\right\|_{S^\ell_{t,j}}\leq\beta^\ell_{t,j}=\beta^\ell_j(\tilde{T}^\ell_{t,j}, L, \alpha/(mL))\triangleq\sigma_0\sqrt{2\log(\frac{mL}{\alpha})+d\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{T^\ell_{t,j}}{dL'\rho_{\min}})}+\sqrt{L'\rho_{\max}}+\sqrt{L'}\kappa \tag{30}
$$

Proof. Let τ be the time slot when the last upload (or download) event happens before time t and sequence $\{t_1, ..., t_{T^\ell_{\tau,j}}\}$ be the time slots when user i_{t_i} that belongs to cluster j^* appears. Denote the L' clusters that belongs to cluster j^* to be

$\{(\ell_1, j_1), ..., (\ell_{L'}, j_{L'})\}$. With a little abuse of notation, we denote $\sum_{p=1}^{L'} \tilde{H}_{\tau,j_p}^{\ell_p}$ and $\sum_{p=1}^{L'} \bar{h}_{\tau,j_p}^{\ell_p}$ by $\tilde{H}_\tau$ and $\bar{h}_\tau$. Recall that by definition, $S_{t,j}^\ell = \tilde{H}_\tau + \sum_{k=1}^{T_{t,j}^\ell} x_{t_k} x_{t_k}^\top, u_{t,j}^\ell = \bar{h}_\tau + \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k} y_{t_k}$. Then we have

$$\begin{aligned}
\theta^* - \hat{\theta}_{t,j}^\ell &= \theta^* - (S_{t,j}^\ell)^{-1}(\sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k} y_{t_k} + \bar{h}_\tau) \\
&= \theta^* - (S_{t,j}^\ell)^{-1}((\sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k} x_{t_k}^\top)\theta^* + \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k} + \bar{h}_\tau) \\
&= \theta^* - (S_{t,j}^\ell)^{-1}(S_{t,j}^\ell \theta^* - \tilde{H}_\tau \theta^* + \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k} + \bar{h}_\tau) \\
&= (S_{t,j}^\ell)^{-1}(\tilde{H}_\tau \theta^* - \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k} - \bar{h}_\tau)
\end{aligned}$$

Then we multiply the both sides by $(S_{t,j}^\ell)^{1/2}$,

$$\begin{aligned}
\left\|\theta^* - \hat{\theta}_{t,j}^\ell\right\|_{S_{t,j}^\ell} &= \left\|\tilde{H}_\tau \theta^* - \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k} - \bar{h}_\tau\right\|_{(S_{t,j}^\ell)^{-1}} && (31) \\
&\leq \left\|\tilde{H}_\tau \theta^*\right\|_{(S_{t,j}^\ell)^{-1}} + \left\|\sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k}\right\|_{(S_{t,j}^\ell)^{-1}} + \left\|\bar{h}_\tau\right\|_{(S_{t,j}^\ell)^{-1}} && (32) \\
&\leq \left\|\tilde{H}_\tau \theta^*\right\|_{\tilde{H}_\tau^{-1}} + \left\|\sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k}\eta_{t_k}\right\|_{\left(\sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k} x_{t_k}^\top + L'\rho_{\min} I\right)^{-1}} + \left\|\bar{h}_\tau\right\|_{\tilde{H}_\tau^{-1}} && (33) \\
&\leq \sqrt{L'\rho_{\max}} + \sigma_0 \sqrt{2\log(\frac{2}{\alpha}) + d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{T_{t,j}^\ell}{dL'\rho_{\min}})} + \sqrt{L'}\kappa && (34)
\end{aligned}$$

where Equation (33) uses $\tilde{H}_\tau \preceq S_{t,j}^\ell + \sum_{k=1}^{T_{\tau,j}^\ell} x_{t_k} x_{t_k}^\top \preceq S_{t,j}^\ell + L'\rho_{\min} I$, Equation (34) uses the Definition 2 and the self-normalized bound in [Abbasi-Yadkori et al., 2011, Theorem 2].

■

A.2 COMMUNICATION COST

Proposition 2. *Under the CDP setting, the total communication cost satisfies:*

$$C_T \leq mL\left(\log T + \frac{d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{T}{d\rho_{\min}})}{\log(\min\{U, D\})} + \frac{d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{2T_0}{d\rho_{\min}})\log(2T_0)}{\log(\min\{U, D\})}\right) \tag{35}$$

Proof. We consider two cases, when $t > 2T_0$ and $t \leq 2T_0$.

Case 1: $t > 2T_0$**.**

Recall that when $t > 2T_0$, the cluster structure are correct and there will be no changing of clusters. Fix any local cluster j at local server ℓ, suppose it shares information with total L' clusters (including itself). Let $\{t_1, t_2, ...t_{n'}\}$ be the sequence of time when upload event or download event happen and the corresponding sequence of local Gram matrix after each event be $\{S_{t_1,j}^\ell, S_{t_2,j}^\ell, ..., S_{t'_n,j}^\ell\}$.

Then we have:

$$\log(\frac{\det(S^{\ell}_{t'_n,j})}{\det(S^{\ell}_{t_1,j})}) = \log(\frac{\det(S^{\ell}_{t_2,j})}{\det(S^{\ell}_{t_1,j})}) + \log(\frac{\det(S^{\ell}_{t_3,j})}{\det(S^{\ell}_{t_2,j})}) + ... + \log(\frac{\det(S^{\ell}_{t_{n'},j})}{\det(S^{\ell}_{t_{n'-1},j})}) \leq \log(\frac{\det(V_T)}{\det(L'\rho_{0,\min}I)}) \tag{36}$$

On one hand, by the definition of the upload and download event, we have $\log(\frac{\det(S^{\ell}_{t_{k+1},j})}{\det(S^{\ell}_{t_k,j})}) \geq \log(\min\{U, D\})$ for any $k \in [n'-1]$. On the other hand, $\log(\frac{\det(V_T)}{\det(L'\rho_{\min}I)}) \leq d\log((L'd\rho_{\max}+T)/d) - d\log(L'\rho_{\min}) = d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{T}{dL'\rho_{\min}}) \leq d\log(\frac{\rho_{\max}}{\rho_{\min}} + \frac{T}{d\rho_{\min}})$, where the last inequality uses the fact that $L' \geq 1$. So $n' \leq \frac{d\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{T}{d\rho_{\min}})}{\log(\min\{U,D\})}$. Since we at most have mL local clusters, the total communication is $\frac{mLd\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{T}{d\rho_{\min}})}{\log(\min\{U,D\})}$.

Case 2: $t < 2T_0$**.** Equivalently, we are in phase $s \leq \lceil \log T_0 \rceil$, then each phase we will communicate $\frac{mLd\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{2T_0}{d\rho_{\min}})}{\log(\min\{U,D\})}$ using the similar argument. Then the total communication is $\frac{mLd\log(\frac{\rho_{\max}}{\rho_{\min}}+\frac{2T_0}{d\rho_{\min}})\log(2T_0)}{\log(\min\{U,D\})}$.

For the communication during the beginning of each phase in order to detect the cluster structure, we know there will be at most $mL\log T$ communications, which concludes the lemma.

■

A.3 PRIVACY GUARANTEE

To guarantee the algorithm preserves the CDP, we use a tree based algorithm where each tree node preserves $(\varepsilon/\sqrt{8\nu\ln(2/\delta)}, \delta/2)$-DP.

A.4 SUPPLEMENTAL EXPERIMENTS

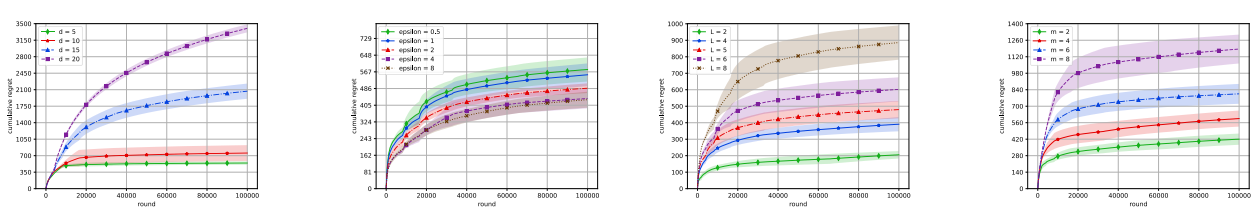
A.4.1 Setting

Synthetic Dataset. We consider a setting of $n = 40$ users with $m = 4$ global clusters and $L = 5$ local servers. The weight vectors and item vectors are generated randomly with dimension $d = 10$. For each global cluster $j \in [m]$, we fix a weight vector θ_j and we guarantee that all weight vectors are orthogonal and the difference gap γ between them is $\sqrt{2}$. We stipulate that the default privacy bugdet is $(\epsilon, \delta) = (1, 0.1)$ and communication upload/download rate $U = D = 1.01$. In each round, a random user comes and the algorithm selects one of $K = 10$ items to recommend to the user. After we receive the feedback and compare it with the best action since we know the true weight vectors, we evaluate the performance of the algorithm by calculating the cumulative regret. The vertical axis indicates the cumulative regret and the horizontal axis indicates the round t.

MovieLens Dataset. We randomly draw 10^3 movies and 10^3 users as preparation for the experiment. Similar to Li and Zhang [2018], we generate the weight vectors on the basis of the 10^3 users and the movies they rated. We randomly select 40 users from these 10^3 users for the experiment. In this experiment, we assume that there are $n = 40$ users with $m = 4$ global clusters and $L = 5$ local servers and the weight vectors and item vectors are generated randomly with dimension $d = 10$. The users are selected randomly from these 10^3 users for the experiment. Besides, we stipulate that the default privacy bugdet is $(\epsilon, \delta) = (1, 0.1)$ and communication upload/download rate $U = D = 1.01$.

A.4.2 Parameter Study

We experiment on a synthetic dataset. Our goal in this section is to numerically investigate the influence of parameters' change on the performance of FCLUB-CDP-DC algorithm.



(a) Vary dimension d and fix other variables (b) Vary privacy budget ϵ and fix other variables (c) Vary the number of local servers L and fix other variables (d) Vary the number of global cluster m and fix other variables

Figure 3: Comparative experiment on synthetic dataset

In Figure 3a, we vary the dimension d from 5 to 20 and fix other parameters. It can be seen that with the increase of dimension d, the convergence speed of regret will slow down. In Figure 3b, we change the privacy budget ϵ from 0.5 to 8. It is obvious that when ϵ is small, the cumulative regret is better because at this time the privacy will have less impact on the recommendation. In Figure 3c and Figure 3d, we vary the number of local server L from 2 to 8 and the number of global cluster m from 2 to 8 respectively. The results show that the value of L and m are positively correlated with the cumulative regret, which is consistent with our conclusion in Theorem 1.

A.4.3 communication cost

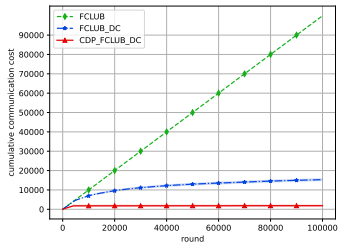


Figure 4: Comparison of communication costs

Communication cost is one of the most important factors we consider when designing the algorithm. In Figure 4, we compare the communication cost of FCLUB using full communication with that of FCLUB-DC and FCLUB-DC-CDP using the delayed communication. The results clearly indicate that our asynchronous communication protocol can reduce the communication cost effectively.

A.4.4 Running Time

Table 1: Comparison of run time

	FCLUB-DC-CDP	FCLUB-DC	FCLUB	SCLUB	CLUB	LinUCB	Homo-DC	Homo
run time (ms)	1.707	1.814	58.870	3.805	0.779	0.647	1.277	29.265
run time (ms)	1.667	1.892	83.492	3.855	0.776	0.654	1.301	35.645

We also compare the average running time (ms) of each round between our algorithm FCLUB-CDP-DC and baselines on synthetic dataset and MovieLens dataset (the first line is synthetic dataset and the second line is MovieLens dataset). It can be seen from Table 1 that due to the use of delay communication, our algorithm has a great improvement in communication cost compared with FCLUB and our run time cost is even lower than SCLUB. However, because of the existence of communication, our run time cost is still higher than CLUB and LinUCB.