Federated Bandit: A Gossiping Approach

ANONYMOUS AUTHOR(S)

In this paper, we study *Federated Bandit*, a decentralized Multi-Armed Bandit problem with a set of N agents, who can only communicate their local data with neighbors described by a connected graph G. Each agent makes a sequence of decisions on selecting an arm from M candidates, yet they only have access to local and potentially biased feedback/evaluation of the true reward for each action taken. Learning only locally will lead agents to sub-optimal actions while converging to a no-regret strategy requires a collection of distributed data. Motivated by the proposal of federated learning, we aim for a solution with which agents will never share their local observations with a central entity, and will be allowed to only share a private copy of his/her own information with their neighbors. We first propose a decentralized bandit algorithm `Gossip_UCB`, which is a coupling of variants of both the classical gossiping algorithm and the celebrated Upper Confidence Bound (UCB) bandit algorithm. We show that `Gossip_UCB` successfully adapts local bandit learning into a global gossiping process for sharing information among connected agents, and achieves guaranteed regret at the order of $O(\max\{\texttt{poly}(N,M)\log T, \texttt{poly}(N,M)\log_{\lambda_2^{-1}} N\})$ for all N agents, where $\lambda_2 \in (0,1)$ is the second largest eigenvalue of the expected gossip matrix, which is a function of G. We then propose `Fed_UCB`, a differentially private version of `Gossip_UCB`, in which the agents preserve ϵ-differential privacy of their local data while achieving $O(\max\{\frac{\texttt{poly}(N,M)}{\epsilon}\log^{2.5} T, \texttt{poly}(N,M)\log_{\lambda_2^{-1}} N\})$ regret.

Additional Key Words and Phrases: federated learning, differential privacy, decentralized multi-armed bandit, heterogeneous rewards

1 INTRODUCTION

When data resides at distributed ends, soliciting them to a single server to perform centralized learning might compromise users' privacy. Among all solutions, federated learning (FL) [14, 42] arises as a promising paradigm, where massive users are allowed to collaboratively train a model while keeping the training data decentralized at local. In this paper, we introduce `federated bandit` with fully decentralized users/decision-makers and heterogeneous rewards. Our aim is to provide a solution to enable collaborative learning among decentralized sequential decision-makers in the classical multi-armed bandit (MAB) setting, but with strong (i) regret guarantee even with heterogeneous reward observations, and (ii) privacy guarantees of each user's local data.

We are motivated by a federated learning scenario where multiple agents hold different and heterogeneous datasets for the same task. This heterogeneity exists in practice for multi-fold of reasons: it could be because of local observation and data collection errors, or it could be due to sampling biases, but the goal of each agent is to cooperatively smooth out these local biases and learn the true optimal action while protecting its private information. Collaborative research among countries is vital when facing global health emergencies, like COVID-19. The collected observations of the effectiveness of treatments might encode local biases due to the difference in the training of medical staff, the difference in following the protocols and equipment, and the difference in the underlying diseases of patients, etc. Therefore, the observed effect buries noise and local biases. In this case, sharing local models or estimates helps average out or smooth out the biases to obtain a more accurate model for diagnosis. Moreover, as mentioned in federated learning literature [14, 21, 42] and due to policy regulations, a centralized effort for data sharing and coordination is often challenging and sometimes prohibited. Therefore, the design of a decentralized federated learning system is both necessary and technically challenging.

Specifically, consider the following running example. Suppose that multiple hospitals decide to test the effectiveness of different treatment plans (*arms*). Due to the limitations such as data size, health condition and demographics of the patient population, and the details of how a treatment is used [21], each individual hospital may not be able to fully and truthfully observe the effect of treatments. In other words, individual hospitals will only observe locally biased feedback on the deployed treatment (*heterogeneous rewards*). Sharing observations across institutes is necessary for the decision-making process. However, due to privacy regulation, it is hard and expensive to call for centralized efforts to coordinate a transportation of data among hospitals. On the other hand, it is relatively easier for individual hospitals to reach agreements to share their observed treatment plan and effects with several others in an ad hoc way.

The hospital treatment selection problem mentioned above is effectively a sequential decision-making problem which can be abstracted as a MAB one. Formally, there is a group of N decision-makers facing a common set of arms. At each step $t = 1, 2, \ldots, T$, each decision-maker selects one arm in parallel. Decision-makers only have access to local *biased* rewards. Therefore, the agents' individually observed rewards do not fully reflect the true quality of each arm. Instead, the arms' true rewards are collectively decided by all decision-makers' local observations. In our heterogeneous setting, we consider a tractable scenario where the true quality of each treatment (arm) is the average of all hospitals' (agents') locally observed quality (in expectation). Each user aim to select the best arm via exchanging information only with their neighbors privately.

The key technical challenge of the above learning problem lies in the fully-decentralized information sharing and privacy protection with sequential observations. First, to reduce the communication overhead and privacy leakage during decentralized information sharing, we aim for a solution with only sharing the information over the adjacency matrix (graph) of agents in a gossiping way. However, classical gossiping methods [4] do not incorporate individual decision-maker's newly observed reward information. Thus, the resulting information at all other agents may not converge and reflect the true statistics of each arm adaptively, which is especially true in the fully-decentralized and heterogeneous settings. Secondly, even though the gossiping update is better than directly sharing data in terms of privacy, we still need a mechanism to ensure a specific privacy level in the worst case. We adopt the solution concept differential privacy (DP) [9–12] and extend our gossiping bandit results to a differentially private one.

In this paper, we attempt to solve the above federated bandit learning problem: (1) We introduce a novel extension of the classical MAB problem to a fully-decentralized federated learning setting with gossiping, where an individual decision-maker only has access to biased rewards, and agents have limited communication capacity and can only exchange their beliefs of rewards with neighbors; (2) We propose `Gossip_UCB` to solve the challenges for combining gossiping with bandit learning processes and develop novel proof techniques to guarantee its regret. (3) To ensure the differential privacy for each observation during the federated bandit learning process, we extend the proposed gossiping bandit algorithm to `Fed_UCB`, and prove agents preserve ϵ-differential privacy of their local data while achieving $O(\max\{\frac{\texttt{poly}(N,M)}{\epsilon}\log^{2.5}T, \texttt{poly}(N,M)\log_{\lambda_2^{-1}}N\})$ regret. `Fed_UCB` is also tested using real medical dataset [35]. (4) To the best of our knowledge, `Fed_UCB` is the first fully decentralized bandit learning framework that handles heterogeneous data sources with a privacy guarantee. The results lay the foundation to study more sophisticated and probably more practical settings (e.g., contextual bandit setting to further handle population biases at each local agent).

Most relevant to us are three lines of works:
Distributed MAB Recently, MAB problems have been studied within a multi-agent setting [2, 15, 20, 22, 28, 38]. But these works mostly either do not consider a consensus reaching in cheap communication setting (gossiping), or do not target on heterogeneous rewards where agents' observations incorporate local bias. For example, instead of reaching consensus among agents, [2, 15, 22, 28, 38] focused on avoiding the collision in wireless communication or cognitive radio. The homogeneous rewards were assumed in [6, 25, 33, 41]. However, the rewards in federated learning setting should be heterogeneous due to various limitations [21].

Information propagation and gossiping The idea of gossiping was originally proposed to solve the *consensus reaching* problem in distributed computation [16, 27, 31, 40], and questions about gossiping convergence rate were studied in [4, 5, 29]. It has also been used to solve distributed problems, such as convex optimization, ranking, and voting problems; and more recently to computing machine learning related statistics. Notable examples include [19] for calculating PCA, [8, 30] for computing U-Statistics, [23, 34] for computing gradients, [13] for federated learning, and [32] for reinforcement learning.

Federated learning and privacy preserving bandit Due to the high demand for privacy protection across different sectors such as financial, medical, and government systems, federated learning is becoming a trending solution that has been widely discussed [3, 14, 18, 35, 42]. Recently, differential privacy has also been adopted in solving MAB problems while ensuring privacy [24, 26, 39], but they either consider a single agent problem or use a homogeneous reward setting. We will follow the idea of DP in our work and offer a theoretically rigorous treatment for our federated bandit problem.

2 PROBLEM FORMULATION

Consider a network consisting of N agents. For ease of presentation, we label the agents from 1 through N. The agents are not aware of such a global labeling, but can differentiate between their neighbors. The set of agents is denoted by $[N] = \{1, 2, \ldots, N\}$. All agents face a common set of M arms, denoted by $[M] = \{1, 2, \ldots, M\}$. At each discrete time $t \in \{1, 2, \ldots, T\}$, each agent i makes a decision on which arm to select from the M options; the selected arm is denoted by $a_i(t) \in [M]$. When agent i selects an arm $k \in [M]$, the agent collects a reward which is generated according to a random variable $X_k(t)$.[1] But the agent cannot observe its exact reward; instead, it observes a locally biased "noisy" copy of the reward, which is generated according to another random variable $X_{i,k}(t)$. The unobservability of $X_k(t)$ can be due to local observational bias. We assume that $\{X_k(t)\}_{t=1}^T$ and $\{X_{i,k}(t)\}_{t=1}^T$ are i.i.d. random processes. For simplicity of analysis, we also assume that all X_k's and $X_{i,k}$'s have bounded support $[0, 1]$. The relationship between $X_k(t)$ and $X_{i,k}(t)$ is as follows. Let μ_k and $\mu_{i,k}$ be the mean of $X_k(t)$ and $X_{i,k}(t)$, respectively. For each $k \in [M]$, the mean of arm k's reward equals the average[2] of the means of all agents' observed rewards, i.e., $\underline{\mu_k := \frac{1}{N}\sum_{i=1}^N \mu_{i,k}}$, which implies that the true reward can be obtained by averaging and thus cancelling out local biases. Note the heterogeneous reward can model the systematic observation bias or the bias of datasets, which is more general and meaningful than the homogeneous reward, especially in FL settings [42] where each agent's systematic observation bias makes the locally optimal solution does not correspond to the real optimal action. Although the bias of datasets is probably more

[1]Different agents may select the same arm k at same time t. If this is the case, their rewards can be different as they may collect different realizations of $X_k(t)$.
[2]It can be generalized to the cases where the global reward is defined as any "convex combination" of all local rewards following the "push sum" idea [17].

suitable to be modeled as contextual bandits in practice, we currently focus on the classical bandit setting for a theoretically sound solution, which is also an essential foundation for future practically feasible extensions. Without loss of generality, suppose $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_M$, which implies that arm 1 is the best option. The difference of each arm's mean reward is denoted by $\Delta_k = \mu_1 - \mu_k$. The *federated bandit problem* is for each agent i to minimize the following (weak) *regret*:

$$R_i(T) = T\mu_1 - \sum_{t=1}^{T} \mathbb{E}\left[X_{a_i(t)}(t)\right] ,$$

with the goal of achieving $R_i(T) = o(T)$ (i.e., $R_i(T)/T \to 0$ as $T \to \infty$) for all $i \in [N]$. It is worth noting that each agent i only observes $X_{i,k}$, $k \in [M]$, and $\mu_{i,1}$ is *not* necessarily the largest among $\mu_{i,1}, \mu_{i,2}, \ldots, \mu_{i,M}$. A naive agent, which uses a standard centralized bandit algorithm, may not solve the problem without exchanging information with other agents. The heterogeneous reward structure is ready for extension to a contextual case by considering the feature $\boldsymbol{v}$ of each sample, where the regret could be $R_i(T) = \mathbb{E}_{\boldsymbol{v}}[T\mu_1(\boldsymbol{v}) - \sum_{t=1}^{T} \mathbb{E}_{X|\boldsymbol{v}}[X_{a_i(t)}(\boldsymbol{v}, t)]]$.

2.1 Privacy Guarantee

The privacy of arms needs to be preserved in federated bandits. We aim to protect privacy from the source of data, i.e. the sequential observations $\{X_{i,k}(t)\}_{t=1}^{T}$. Differential privacy (DP) is one popular mechanism to ensure some privacy level of an algorithm $\mathcal{B}$ [10]. A DP mechanism can make the adversary hard to distinguish two adjacent streams $\{X_{i,k}(t)\}_{t=1}^{T}$ and $\{X'_{i,k}(t)\}_{t=1}^{T}$, which differ at each time t. Let C be the space of all possible outputs by Algorithm $\mathcal{B}$. DP is defined as:

Definition 1. *(Differential privacy [10]) A (randomized) algorithm $\mathcal{B}$ is ϵ-differentially private if for any adjacent streams $\{X_{i,k}(t)\}_{t=1}^{T}$ and $\{X'_{i,k}(t)\}_{t=1}^{T}$, and for all sets $O \in C$,*

$$\mathbb{P}\left[\mathcal{A}(\{X_{i,k}(t)\}_{t=1}^{T}) \in O\right] \leq e^{\epsilon} \cdot \mathbb{P}\left[\mathcal{A}(\{X'_{i,k}(t)\}_{t=1}^{T}) \in O\right].$$

2.2 Communication Graph

The neighbor relationships among the agents is described by a simple, undirected, connected graph $G = (V, E)$, whose vertices correspond to agents and whose edges depict neighbor relationships. Denote $\mathcal{N}_i$ as the set of nodes that are directly connected to agent i. Then, $\mathcal{N}_i$ is also the set of agent i's neighbors. We follow the setting in the classical gossiping [4] that at each time t, exactly one pair of two neighboring agents on an edge in E are activated and exchange information.

3 GOSSIP UCB

Before we offer the privacy-preserving solution, we first introduce an extension of the classical Upper Confidence Bound algorithm to a gossiping setting. As may be noticed, in the classical gossiping setting [4], the consensus is defined over initial data only. While in our setting, not only is the gossiping process required to incorporate with newly arrived data from each agent, but also the gossiped information will affect the arm to be selected and thus the observed data of an agent in the future. We present an algorithm, called `Gossip_UCB`, to solve the gossiping bandit problem. The algorithm hinges on combining and extending the classical gossiping algorithm and the celebrated UCB1 index policy[1]; yet our algorithm requires substantial changes for both the gossiping and bandit learning steps. While we will present the algorithm and analysis in detail, we outline here several crucial steps:

(1) Different from the classical gossiping algorithm, the gossiping procedure will incorporate new information from each agent's local sampling and observations at each step t. We adopt the classical gossip algorithm by adding "gradient" information at each step.

(2) Compared to standard bandit learning with only one decision maker where the traditional sample complexity bound can be employed, we need to cope with the uncertainties during gossiping. **(3)** A fully-decentralized structure requires designing a local information sharing mechanism. Besides, the delayed impact of local information sharing should be bounded analytically for computing the confidence bound locally.

3.1 Preliminaries

We define several quantities that will help us present our algorithm and analysis smoothly.

Sample counts: Each agent i maintains two sets of counters:
- $n_{i,k}(t)$: the number of times agent i has sampled arm k by time t;
- $\tilde{n}_{i,k}(t)$: agent i's local estimate of global maximum of pulls on arm k and is defined as

$$\tilde{n}_{i,k}(t+1) = \max\{n_{i,k}(t), \tilde{n}_{j,k}(t), j \in \mathcal{N}_i\}. \tag{1}$$

We assume agent i can observe $\tilde{n}_{j,k}(t),\ j \in \mathcal{N}_i$, thus is able to update $\tilde{n}_{i,k}(t)$ at each time.

Sample mean: Let $\mathbb{1}(\cdot)$ be an indicator function that returns 1 when the specific condition holds and 0 otherwise. Sample mean $\tilde{X}_{i,k}(t)$ is the average observation of agent i on arm k at time t:

$$\tilde{X}_{i,k}(t) = \frac{1}{n_{i,k}(t)} \sum_{\tau=1}^{t} \mathbb{1}(a_i(\tau) = k) \cdot X_{i,a_i(\tau)}(\tau). \tag{2}$$

Estimate of rewards: Each agent i maintains an estimate of the reward of arm k at time t, which is supposed to be unbiased and denoted by $\vartheta_{i,k}(t)$. The agents' goal is to narrow the gap between $\vartheta_{i,k}(t)$ and μ_k with sequential observations and gossiping.

Upper confidence bound: In the UCB algorithm, agent i's belief on each arm k relies on two terms: the estimate $\vartheta_{i,k}(t)$ and the upper confidence bound $C_{i,k}(t)$. The latter term denotes the uncertainty of belief. The arm to be pulled is selected as $a_i(t) = \arg\max_k\ \vartheta_{i,k}(t-1) + C_{i,k}(t)$.

Gossiping matrix: Denote the gossiping matrix over G as

$$W := \frac{1}{|E|} \sum_{(i,j)\in E} \left(I_N - \frac{1}{2}(e_i - e_j)(e_i - e_j)^\top \right),$$

which is a positive semi-definite matrix whose largest eigenvalue equals 1, and its second largest eigenvalue is denoted by $\lambda_2(W)$ and short-handed as λ_2 without ambiguity. Note $\lambda_2 < 1$ whenever G is connected [4].

3.2 Algorithm

`Gossip_UCB` is detailed in Algorithm 1. Each agent runs this algorithm in parallel. Note all the information is shared in a fully distributed fashion and the bandit estimate is updated in a gossiping way. There are two points worth noting.

Local information sharing Throughout the algorithm, agents need to share two local variables with their neighbors: the number of observations $n_{i,k}(t)$ and the estimate $\vartheta_{i,k}(t)$. The sample count $n_{i,k}(t)$ is shared to keep all the agents "in the same page". Note the bottleneck of a bandit problem is insufficient observations of a particular arm k, and the essential of UCB algorithms is encouraging the exploration of these "undersampled" arms. In the multi-agent scenario, we can take the advantage of neighboring agents and require some local consistency in sample counts. Particularly, we want to keep all agents' knowledge of arm k "at the same page" by encouraging $n_{i,k}(t) \geq \tilde{n}_{i,k}(t) - N$ in line 6. Recall that $\tilde{n}_{i,k}(t)$ is agent i's local estimate of global maximum

Algorithm 1: Gossip_UCB

Input: $G, T, C_{i,k}(t)$

1 **Initialization:** Each agent pulls each arm once, and receives a reward $X_{i,k}(0)$, $i \in [N]$, $k \in [M]$. Set $n_{i,k}(0) = 1$, $\vartheta_{i,k}(0) = \tilde{X}_{i,k}(0) = X_{i,k}(0)$.

2 **for** $t = 1, \ldots, T$ **do**

3 $\quad \mathcal{A}_i = \varnothing$

4 $\quad n_{i,k}(t) = n_{i,k}(t-1), \forall k \in [M]$

5 $\quad \tilde{n}_{i,k}(t+1) = \max\{n_{i,k}(t), \tilde{n}_{j,k}(t), j \in \mathcal{N}_i\}, \forall k \in [M]$

6 $\quad$ Put k into set $\mathcal{A}_i$ **if** $n_{i,k}(t) < \tilde{n}_{i,k}(t) - N, \forall k \in [M]$ *// local consistency requirements*

7 $\quad$ **if** $\mathcal{A}_i$ *is empty* **then**

8 $\quad\quad$ **for** $k = 1, \ldots, M$ **do**

9 $\quad\quad\quad Q_{i,k}(t) = \vartheta_{i,k}(t-1) + C_{i,k}(t)$ *// update the belief on each arm*

10 $\quad\quad\quad a_i(t) = \arg\max_k Q_{i,k}(t)$ *// select the best arm to pull*

11 $\quad\quad$ **end**

12 $\quad$ **else**

13 $\quad\quad a_i(t)$ is randomly selected from $\mathcal{A}_i$

14 $\quad$ **end**

15 $\quad$ Observe arm $a_i(t)$, get $X_{i,a_i(t)}(t)$, and update $\tilde{X}_{i,k}(t), \forall k$, following (2)

16 $\quad n_{i,a_i(t)} = n_{i,a_i(t)} + 1$

17 $\quad$ **if** *agent i is selected to gossip with agent j* **then**

18 $\quad\quad$ agent i sends $\vartheta_{i,k}(t-1)$ to agent j

19 $\quad\quad$ agent i receives $\vartheta_{j,k}(t-1)$ from agent j

20 $\quad\quad \vartheta_{i,k}(t) = \frac{\vartheta_{i,k}(t-1) + \vartheta_{j,k}(t-1)}{2} + \tilde{X}_{i,k}(t) - \tilde{X}_{i,k}(t-1)$ *// normal update*

21 $\quad$ **else**

22 $\quad\quad \vartheta_{i,k}(t) = \vartheta_{i,k}(t-1) + \tilde{X}_{i,k}(t) - \tilde{X}_{i,k}(t-1)$

23 $\quad$ **end**

24 $\quad$ *// gossiping update*

25 **end**

number of pulls, and is updated by local observations only as is defined in the previous section. Moreover, we will prove that, this requirement helps us get rid of relying on any global sample count so that $C_{i,k}(t)$ can be computed by each agent locally. The estimate $\vartheta_{i,k}(t)$ is updated in a gossiping way.

Gossip bandit update The gossip updates are defined in line 20 and line 22. In traditional bandit problems, it is enough for each agent to maintain $\tilde{X}_{i,k}(t)$. However, in our concerned gossiping setting, solely relying on $\tilde{X}_{i,k}(t)$ may induce a biased estimate. The gossiping mechanism follows [23], where the difference $\tilde{X}_{i,k}(t) - \tilde{X}_{i,k}(t-1)$ can be seen as a gradient. Later we will show the effectiveness of the proposed gossip bandit update.

3.3 Theoretical Analysis

Note the arm selection in Algorithm 1 relies on the upper confidence bound $C_{i,k}(t)$. In this section, we would like to find an appropriate choice of $C_{i,k}(t)$ and derive the corresponding upper bound of each agent's regret by implementing Gossip_UCB.

Technical challenges The main technical challenge is tackling the *coupling effects of gossiping and bandit learning*. On a high level, classical technical results in gossiping assumed a *static* piece

of information that would not change much during the entire gossiping phase. The literature [37] often adopted a phased-based learning strategy (by caching the gossiped information) to avoid changes, which effectively delays the update of learned policy (the learning needs to wait for the gossiping to converge). Since agents only share information with their neighbors, globally, there is latency in receiving this data at the non-directly connected agents. Additionally, with the existence of multiple agents, ensuring local consistency as lines 6 and 13 will incur extra delay impacts when $|\mathcal{A}_i| > 1$. The *delayed impact* affects the immediate decisions taken by other agents and further affects the gossiping process in the near future. From the bandit learning's perspective, this delay might lead to inaccurate computation of the index policies. A poorly made decision will further have cascading effects to other receiving agents, which is especially challenging in heterogeneous settings. The key step to tackle this challenge is to firstly characterize the *delayed impact* and then find an upper bound of the optimal *variance proxy* of $\vartheta_{i,k}(t)$.

Intuitively speaking, the minimum number of selection over agents controls the quality of the average statistics. In line 6, we encourage $n_{i,k}(t) \geq \tilde{n}_{i,k}(t) - N$ to make sure all agents' knowledge of arm k is "at the same page". However, due to the limitation of the bandit feedback (one arm each time) as line 13, extra delays will occur thus $n_{i,k}(t) \geq \tilde{n}_{i,k}(t) - N$ cannot be guaranteed when $|\mathcal{A}_i| > 1$. We analyze the actually achieved local consistency in Lemma 1, Lemma 2, and Lemma 3. Particularly, denote by $d_{i,j}$ the distance (the number of directed edges in the shortest directed path) from vertex i to vertex j in the neighbor graph G. Note $d_{i,i} = 0$ and $d_{i,j} < N$ since G is strongly connected. W.l.o.g., let $n_{i,k}(t) = 0, \forall i \in [N], k \in [M]$ when $t < 0$. Then we have Lemma 1.

Lemma 1. *For any $i \in [N]$ and $k \in [M]$,*

$$\tilde{n}_{i,k}(t+1) = \max_{j \in [N]} \left\{ n_{j,k}(t - d_{j,i}) \right\}. \tag{3}$$

Proof. We will prove the lemma by induction on t. For the basis step, suppose that $t = 0$. In this case, $\tilde{n}_{i,k}(1) = \max\{n_{i,k}(0),\ \tilde{n}_{j,k}(0),\ j \in \mathcal{N}_i\} = 1$. Note that $\max_{j \in [N]}\{n_{j,k}(t - d_{j,i})\} = n_{i,k}(0) = 1$. Thus, (3) holds when $t = 0$.

For the inductive step, assume (3) holds at time t, and now consider time $t + 1$. Note that

$$\begin{aligned} \tilde{n}_{i,k}(t+1) &= \max\{n_{i,k}(t),\ \tilde{n}_{j,k}(t),\ j \in \mathcal{N}_i\} \\ &= \max\{n_{i,k}(t),\ n_{h,k}(t - d_{j,h} - 1), h \in [N],\ j \in \mathcal{N}_i\}. \end{aligned}$$

It is easy to see that $d_{h,i} \leq d_{j,i} + d_{h,j} = 1 + d_{h,j}$. Since $n_{i,k}(t)$ is a non-decreasing function of t by its definition,

$$\begin{aligned} \tilde{n}_{i,k}(t+1) &\leq \max\{n_{i,k}(t),\ n_{h,k}(t - d_{h,i}),\ h \in [N]\} \\ &= \max_{j \in [N]}\{n_{j,k}(t - d_{j,i})\}. \end{aligned} \tag{4}$$

Fix any vertex $j \in [N]$ and let $p = (j, v_{d_{j,i}}, \ldots, v_2, i)$ be a shortest directed path from j to i in G. From (1),

$$\begin{aligned} \tilde{n}_{i,k}(t+1) &\geq \tilde{n}_{v_2,k}(t) \geq \cdots \geq \tilde{n}_{v_{d_{j,i}},k}(t - d_{j,i} + 2) \\ &\geq \tilde{n}_{j,k}(t - d_{j,i} + 1) \geq n_{j,k}(t - d_{j,i}). \end{aligned} \tag{5}$$

Since j is arbitrarily chosen from $[N]$, we have $\tilde{n}_{i,k}(t+1) \geq \max_{j \in [N]}\{n_{j,k}(t - d_{j,i})\}$. Combining with (4), we have

$$\tilde{n}_{i,k}(t+1) = \max_{j \in [N]}\{n_{j,k}(t - d_{j,i})\}.$$

So (3) also holds at $t+1$, which completes the induction. □

With Lemma 1, we are ready to present the actual local consistency achieved by Algorithm 1.

Lemma 2. *$\forall i \in [N], k \in [M]$, we have $n_{i,k}(t) > \tilde{n}_{i,k}(t+1) - 3MN$.*

Proof. We will prove the lemma by contradiction. Suppose that, to the contrary, $\exists\ i, k_1$ such that $n_{i,k_1}(t) \leq \tilde{n}_{i,k_1}(t+1) - 3MN$. Let t' denote the first time at which the equality holds, i.e.,

$$n_{i,k_1}(t') = \tilde{n}_{i,k_1}(t'+1) - 3MN.$$

Here t' must exist, since when $t=0$, we have $n_{i,k}(0) > \tilde{n}_{i,k}(1) - 3MN$, since both $n_{i,k}(t)$ and $\tilde{n}_{i,k}(t)$ increase by 0 and 1 at each time instance, if there exists some t such that $n_{i,k_1}(t) < \tilde{n}_{i,k_1}(t+1) - 3MN$, there must exist a t' between 0 and t, such that $n_{i,k_1}(t') = \tilde{n}_{i,k_1}(t'+1) - 3MN$. According to Lemma 1, $\exists\ j \in [N]$ such that

$$\tilde{n}_{i,k_1}(t'+1) = n_{j,k_1}(t' - d_{j,i}). \tag{6}$$

Then,

$$n_{j,k_1}(t' - d_{j,i}) - n_{i,k_1}(t') = 3MN,$$

and t' is the earliest time instant at which $n_{j,k_1}(t' - d_{i,j}) - n_{i,k_1}(t') \geq 3MN$ holds. This implies that at time $t' - d_{i,j}$, agent j pulls arm k_1.

Since each agent must pull an arm at each time, we have $\sum_k n_{i,k}(t) = t,\ \forall i \in [N]$. Then,

$$\sum_{k \in [M] \setminus k_1} n_{i,k}(t') - \sum_{k \in [M] \setminus k_1} n_{j,k}(t' - d_{j,i}) = 3MN + d_{j,i}.$$

Applying the Pigeonhole principle, $\exists\ k_2 \in [M]$ such that

$$n_{i,k_2}(t') - n_{j,k_2}(t' - d_{j,i}) \geq \frac{3MN + d_{j,i}}{M-1} > 3N.$$

According to the definition of $n_{i,k}(t)$, it is non-decreasing and $n_{i,k}(t+1) \leq n_{i,k}(t) + 1$. Thus,

$$\begin{aligned} n_{i,k_2}(t' - 2d_{j,i}) - n_{j,k_2}(t' - d_{j,i}) &> 3N - 2d_{j,i} \\ &> N. \end{aligned}$$

Using (5), we have $\tilde{n}_{j,k_2}(t' - d_{j,i} + 1) \geq n_{i,k_2}(t' - d_{j,i} - d_{i,j})$. Thus,

$$\tilde{n}_{j,k_2}(t' - d_{j,i} + 1) - n_{j,k_2}(t' - d_{j,i}) > N.$$

From the above analysis, agent j must pull arm k_1 at time $t' - d_{j,i}$. According to the decision making step of the algorithm, there holds

$$\tilde{n}_{j,k_1}(t' - d_{j,i} + 1) - n_{j,k_1}(t' - d_{j,i}) \geq N > 0. \tag{7}$$

Note that from (5),

$$\tilde{n}_{i,k_1}(t'+1) \geq \tilde{n}_{j,k_1}(t' - d_{j,i} + 1). \tag{8}$$

Combining (6) – (8) together, we have

$$\begin{aligned} n_{j,k_1}(t' - d_{j,i}) &= \tilde{n}_{i,k_1}(t'+1) \\ &\geq \tilde{n}_{j,k_1}(t' - d_{j,i} + 1) \\ &> n_{j,k_1}(t' - d_{j,i}), \end{aligned}$$

which is a contradiction. Therefore, the statement of the lemma is true. □

With Lemma 1 and Lemma 2, we formally show how agents' knowledge are "at the same page".

Lemma 3. *$\forall i \in [N], k \in [M]$, when $n_{i,k}(t) \geq (3M+1)N$, we have $\max_{j\in[N]} n_{j,k}(t) \leq 2n_{i,k}(t)$.*

Proof. From (5), $\tilde{n}_{i,k}(t+1) \geq n_{h,k}(t-d_{h,i}),\ \forall h \in [N]$. Combining with $n_{i,k}(t+1) \leq n_{i,k}(t)+1$, we have

$$\tilde{n}_{i,k}(t+1) \geq n_{h,k}(t) - d_{h,i} \geq n_{h,k}(t) - N.$$

From Lemma 2, we have

$$n_{i,k}(t) \geq n_{h,k}(t) - (3M+1)N,\ \forall\ h \in [N].$$

Since h is arbitrarily chosen and $n_{i,k}(t) \geq (3M+1)N$,

$$\max_{j\in[N]} n_{j,k}(t) \leq n_{i,k}(t) + (3M+1)N \leq 2n_{i,k}(t),$$

which completes the proof. □

By analyzing the delayed impacts on $\vartheta_{i,k}(t)$, we have the following bound on its variance proxy.

Lemma 4. *When $n_{i,k}(t) \geq \max\{L, (3M+1)N\}, \forall i \in [N]$, with probability at least $1-2p_0$, the optimal variance proxy of $\vartheta_{i,k}(t)$ is bounded by $\frac{N}{2n_{i,k}(t)}$, where $p_0 = \lambda_2^{n_{j,k}(t)/6}/(1-\lambda_2^{1/3})$, and L is the value which makes $\lambda_2^{t/6}/(1-\lambda_2^{1/3}) < 2(Nt)^{-1}$ hold for all $t \geq L$.*

Proof. We can infer from the algorithm that each $\vartheta_{i,k}(t)$ is a linear combination of $X_{j,k}(\tau)$, for all $j \in [N], \tau \in \{0,\dots,t\}$. Define $c_{i,k,j}^{(\tau)}$ as the corresponding coefficient of such $X_{j,k}(\tau)$ in $\vartheta_{i,k}(t)$. To find the variance proxy of $\vartheta_{i,k}(t)$, which is $\frac{1}{4}\sum_{j=1}^{N}\sum_{\tau=1}^{t}|c_{i,k,j}^{(\tau)}|^2$ according to Property 2 and Property 3 in Appendix A, we will estimate the value of $c_{i,k,j}^{(\tau)}$ in the following content.

The vector form of the iteration process can be expressed as:

$$\vartheta_k(t) = W(t)\vartheta_k(t-1) + \tilde{X}_k(t) - \tilde{X}_k(t-1), \tag{9}$$

for all $k \in [M]$, where $W(t) = I - \frac{1}{2}(e_i - e_j)(e_i - e_j)^\top$ is the gossip updating matrix when agent i and j are selected to exchange information at time t. According to (9), we have:

$$\begin{aligned}\vartheta_k(t) &= W(t)\vartheta_k(t-1) + \tilde{X}_k(t) - \tilde{X}_k(t-1)\\ &= \left(W(t)\cdots W(1) - W(t)\cdots W(2)\right)X_k(0) + \left(W(t)\cdots W(2) - W(t)\cdots W(3)\right)\tilde{X}_k(1)\\ &\quad + \cdots + (W(t) - I)\tilde{X}_k(t-1) + \tilde{X}_k(t).\end{aligned}$$

Define $\tau_1,\dots,\tau_{n_{j,k}(t)}$ as the time instance before t when agent j pulls arm k, so $\tau_1 = 0$, and we can get

$$\begin{aligned}\vartheta_{i,k}(t) = \sum_j \Bigg(&[W(t)\cdots W(\tau_1+1) - W(t)\cdots W(\tau_2+1)]_{i,j}\tilde{X}_{j,k}(\tau_1)\\ &+ \cdots + [W(t)\cdots W(\tau_{n_{j,k}(t)-1}+1) - W(t)\cdots W(\tau_{n_{j,k}(t)}+1)]_{i,j}\tilde{X}_{j,k}(\tau_{n_{j,k}(t)-1})\\ &+ [W(t)\cdots W(\tau_{n_{j,k}(t)}+1)]_{i,j}\tilde{X}_{j,k}(\tau_{n_{j,k}(t)})\Bigg),\end{aligned} \tag{10}$$

and

$$c_{i,k,j}^{(0)} = \Bigg[(W(t)\cdots W(\tau_1+1) - W(t)\cdots W(\tau_2+1)) + \cdots + \frac{W(t)\cdots W(\tau_{n_{j,k}(t)-1}+1) - W(t)\cdots W(\tau_{n_{j,k}(t)}+1)}{n_{j,k}(t)-1} + \frac{W(t)\cdots W(\tau_{n_{j,k}(t)}+1)}{n_{j,k}(t)} \Bigg]_{i,j}, \tag{11}$$

where $[\cdot]_{i,j}$ denotes the entry in the i-th row and the j-th column. Notice due to the delay in updating, there exists some h', when $h > h'$, $\tilde{X}_{j,k}(\tau_h)$ can never be transmitted to agent i till time t, in this case, the coefficient of such $\tilde{X}_{j,k}(\tau_h)$ in (10) is zero. The number of non-zero terms in (11) is always no larger than $\sum_{j=1}^{N} n_{j,k}(t)$.

We can also write (11) as

$$c_{i,k,j}^{(0)} = \left[W(t)\cdots W(\tau_1+1) - \sum_{h=2}^{n_{j,k}(t)} \frac{W(t)\cdots W(\tau_h+1)}{(h-1)h} \right]_{i,j}. \tag{12}$$

We want to estimate the value of $W(t)\cdots W(\tau_h+1)$. According to [4], we have

$$\mathbb{P}\left(\left| [W(t)\cdots W(\tau_h+1)]_{i,j} - \frac{1}{N} \right| > \lambda_2^{\frac{1}{3}(t-\tau_h)} \right) < \lambda_2^{\frac{1}{3}(t-\tau_h)}, \tag{13}$$

where $\lambda_2 \in (0,1)$ is the second largest eigenvalue of the expected gossip matrix W. Then with probability at least $1-p_0$, we have

$$\begin{aligned}
c_{i,k,j}^{(0)} =& \left[W(t)\cdots W(\tau_1+1) - \left(\sum_{h=2}^{\frac{n_{j,k}(t)}{2}} + \sum_{h=\frac{n_{j,k}(t)}{2}+1}^{n_{j,k}(t)} \right) \frac{W(t)\cdots W(\tau_h+1)}{(h-1)h} \right]_{i,j} \\
>& \frac{1}{N} - \lambda_2^{\frac{1}{3}t} - \left[\sum_{h=2}^{\frac{n_{j,k}(t)}{2}} W(t)\cdots W(\tau_h+1) \right]_{i,j} - \sum_{h=\frac{n_{j,k}(t)}{2}+1}^{n_{j,k}(t)} \frac{1}{(h-1)h} \\
=& \frac{1}{N}\left(1 - \sum_{h=2}^{\frac{n_{j,k}(t)}{2}} \frac{1}{(h-1)h} \right) - \sum_{h=1}^{\frac{n_{j,k}(t)}{2}} \lambda_2^{\frac{1}{3}(t-\tau_h)} - \sum_{h=\frac{n_{j,k}(t)}{2}+1}^{n_{j,k}(t)} \frac{1}{(h-1)h} \\
>& \frac{1}{N}\left(1 - \sum_{h=2}^{\frac{n_{j,k}(t)}{2}} \frac{1}{(h-1)h} \right) - \sum_{h=\frac{n_{j,k}(t)}{2}}^{n_{j,k}(t)} \lambda_2^{\frac{1}{3}h} - \sum_{h=\frac{n_{j,k}(t)}{2}+1}^{n_{j,k}(t)} \frac{1}{(h-1)h} \\
>& \frac{1}{N}\frac{2}{n_{j,k}(t)} - \frac{\lambda_2^{\frac{n_{j,k}(t)}{6}}}{1-\lambda_2^{\frac{1}{3}}} - \frac{1}{n_{j,k}(t)},
\end{aligned}$$

and

$$c_{i,k,j}^{(0)} \leq \left[W(t)\cdots W(\tau_1+1) - \sum_{h=2}^{\frac{n_{j,k}(t)}{2}} \frac{W(t)\cdots W(\tau_h)}{(h-1)h} \right]_{i,j}$$
$$< \frac{1}{N}\left(1 - \sum_{h=2}^{\frac{n_{j,k}(t)}{2}} \frac{1}{(h-1)h}\right) + \sum_{h=\frac{n_{j,k}(t)}{2}}^{n_{j,k}(t)} \lambda_2^{\frac{1}{3}h} < \frac{1}{N}\frac{2}{n_{j,k}(t)} + \frac{\lambda_2^{\frac{n_{j,k}(t)}{6}}}{1-\lambda_2^{\frac{1}{3}}},$$

where

$$p_0 = \sum_{h=\frac{n_{j,k}(t)}{2}}^{n_{j,k}(t)} \lambda_2^{\frac{1}{3}h} = \frac{\lambda_2^{\frac{n_{j,k}(t)}{6}}}{1-\lambda_2^{\frac{1}{3}}}.$$

Thus, with probability at least $1-p_0$:

$$|c_{i,k,j}^{(0)}| < \left(1 - \frac{2}{N}\right)\frac{1}{n_{j,k}(t)} + p_0.$$

Let L be the value which makes $\frac{\lambda_2^{\frac{t}{6}}}{1-\lambda_2^{\frac{1}{3}}} < \frac{2}{Nt}$ hold for all $t \geq L$. So when $n_{j,k}(t) \geq L$, we have

$$|c_{i,k,j}^{(0)}| < \frac{1}{n_{j,k}(t)}.$$

It is easy to see other $c_{i,k,j}^{(\tau_h)}$ is the last $n_{j,k}(t)-h+1$ terms in $c_{i,k,j}^{(0)}$ which is defined in (11), so similarly, $c_{i,k,j}^{(\tau_h)}$ also have above property, for all $h \leq n_{j,k}(t)$. Thus when $n_{i,k}(t) \geq \max\{L, (3M+1)N\}, \forall i$, using Lemma 3, we know, with probability at least $1-2p_0$,

$$\sum_{j=1}^{N}\sum_{h=1}^{n_{j,k}(t)} |c_{i,k,j}^{(\tau_h)}|^2 < \sum_{j=1}^{N} \frac{n_{j,k}(t)}{n_{j,k}^2(t)} < \frac{2N}{n_{i,k}(t)}.$$

And further we can get

$$\sigma_{i,k}^2(t) = \frac{1}{4}\sum_{j=1}^{N}\sum_{h=1}^{n_{j,k}(t)} |c_{i,k,j}^{(\tau_h)}|^2 < \frac{N}{2n_{i,k}(t)},$$

which proves the lemma. □

Based on Lemma 4, we have the following main result:

Theorem 1. *(Main Result for* `Gossip_UCB`*) For the* `Gossip_UCB` *algorithm with bounded reward over* $[0,1]$*, and*

$$C_{i,k}(t) = \sqrt{\frac{2N}{n_{i,k}(t)}\log t} + C_1, \tag{14}$$

the regret of each agent i until time T satisfies

$$R_i(T) < \sum_{\Delta_k>0} \Delta_k \left(\max\left\{ \frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2}\log T, L, (3M+1)N \right\} + C_2 \right),$$

where $C_1 = \frac{64}{N^{17}}, C_2 = \frac{2\pi^2}{3} + \frac{12}{(1-\lambda_2^{1/3})\log\lambda_2^{-1}}$.

Proof. We need to find the confidence bound $C_{i,k}(t)$ such that $\mathbb{P}(|\vartheta_{i,k}(t) - \mu_k| \geq C_{i,k}(t))$ could be upper bounded in a sufficiently small order. Note

$$\mathbb{P}(|\vartheta_{i,k}(t) - \mu_k| \geq C_{i,k}(t)) = \mathbb{P}(|\vartheta_{i,k}(t) - \mathbb{E}(\vartheta_{i,k}(t)) + \mathbb{E}(\vartheta_{i,k}(t)) - \mu_k| \geq C_{i,k}(t))$$

We need to bound $|\vartheta_{i,k}(t) - \mathbb{E}(\vartheta_{i,k}(t))|$ and $|\mathbb{E}(\vartheta_{i,k}(t)) - \mu_k|$, respectively.

According to the Hoeffding's inequality for sub-Gaussian random variables (specified in Property 1 of Appendix A) and Lemma 4,

$$\mathbb{P}\left(|\vartheta_{i,k}(t) - \mathbb{E}[\vartheta_{i,k}(t)]| \geq \sqrt{\frac{2N}{n_{i,k}(t)}\log t}\right) \leq 2\exp\frac{\frac{2N}{n_{i,k}(t)}\log t}{2\sigma_{i,k}^2(t)} < \frac{2}{t^2}.$$

Taking expectation on both sides of (9) yields:

$$\begin{aligned}\mathbb{E}(\vartheta_k(t)) &= W\mathbb{E}(\vartheta_{i,k}(t-1)) + \mathbb{E}(\tilde{X}_k(t) - \tilde{X}_k(t-1)) \\ &= W\mathbb{E}(\vartheta_{i,k}(t-1)) \\ &= W^t\mathbb{E}(\vartheta_k(0))\end{aligned}$$

Since

$$\|W^t - \frac{1}{N}1\cdot 1^\top\|_2 = \lambda_2^t,$$

denoting 1_i by the $N\times 1$ column vector with the i-th element being 1 and others being 0, we have

$$\begin{aligned}&|\mathbb{E}(\vartheta_{i,k}(t)) - \mu_k| \\ =&|1_i^\top(\mathbb{E}(\boldsymbol{\vartheta}_k(t)) - \frac{1}{N}1\cdot 1^\top\cdot\mathbb{E}(\boldsymbol{\vartheta}_k(0))| \\ =&|1_i^\top(W^t - \frac{1}{N}1\cdot 1^\top)\mathbb{E}(\boldsymbol{\vartheta}_k(0))| \\ \leq&\|1_i^\top(W^t - \frac{1}{N}1\cdot 1^\top)\|_2\|\mathbb{E}(\boldsymbol{\vartheta}_k(0))\|_2 \qquad \text{(Hölder's inequality)} \\ \leq&\sqrt{N}\|1_i\|_2\|W^t - \frac{1}{N}1\cdot 1^\top\|_2 \\ \leq&\sqrt{N}\lambda_2^t\end{aligned} \tag{15}$$

holds for all $i \in [N], k \in [M]$. Thus

$$\begin{aligned}\mathbb{P}(|\vartheta_{i,k}(t) - \mu_k| \geq C_{i,k}(t)) =& \mathbb{P}(|\vartheta_{i,k}(t) - \mathbb{E}(\vartheta_{i,k}(t)) + \mathbb{E}(\vartheta_{i,k}(t)) - \mu_k| \geq C_{i,k}(t)) \\ \leq& \mathbb{P}(|\vartheta_{i,k}(t) - \mathbb{E}(\vartheta_{i,k}(t))| \geq C_{i,k}(t) - \sqrt{N}\lambda_2^t).\end{aligned}$$

When $n_{i,k}(t) \geq \max\{L, (3M+1)N\}, \forall i \in [N]$, we have

$$\sqrt{N}\lambda_2^t < \frac{64}{N^{17}} \triangleq C_1.$$

Set $C_{i,k}(t) = \sqrt{\frac{2N}{n_{i,k}(t)}\log t} + C_1$, then with probability at least $1 - p_0$,

$$\begin{aligned}\mathbb{P}(\vartheta_{i,k}(t) - \mu_k \geq C_{i,k}(t)) <& \mathbb{P}\left(\vartheta_{i,k}(t) - \mathbb{E}(\vartheta_{i,k}(t)) \geq C_{i,k}(t) - C_1\right) \\ <& \frac{1}{t^2}.\end{aligned}$$

Similarly,

$$\mathbb{P}(\mu_k - \vartheta_{i,k}(t) \geq C_{i,k}(t)) < \frac{1}{t^2}.$$

Let us now look into the standard UCB algorithm, it requires, if at time t, agent i chooses arm k instead of arm 1, there are only four possible cases [1]:

(1) $k \in \mathcal{A}_i$;

(2) $\vartheta_{i,k}(t) - \mu_k \geq C_{i,k}(t)$;

(3) $\mu_1 - \vartheta_{i,1}(t) \geq C_{i,1}(t)$;

(4) $\mu_1 - \mu_k < 2C_{i,k}(t)$.

It is easy to verify when

$$n_{i,k}(t) \geq \frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2} \log T,$$

case (4) does not hold. Define t' as the time when $n_{i,k}(t')$ reach the above value, then

$$\sum_{t>t'} \mathbb{P}(\vartheta_{i,k}(t) - \mu_k \geq C_{i,k}(t)) \leq \sum_{t>t'} \frac{1}{t^2} \leq \frac{\pi^2}{6},$$
$$\sum_{t>t'} \mathbb{P}(\mu_1 - \vartheta_{i,1}(t) \geq C_{i,1}(t)) \leq \sum_{t>t'} \frac{1}{t^2} \leq \frac{\pi^2}{6}.$$

So after t', the expected time agent i pulls arm k which is due to Case 2 and 3 is no larger $\frac{\pi^2}{3}$. So the expected time Case 1 happens should also be no larger than $\frac{\pi^2}{3}$. Together we have

$$\mathbb{E}[n_{i,k}(T) - n_{i,k}(t')] \leq \frac{2\pi^2}{3}.$$

Thus with probability at least $1 - 2p_0$,

$$\begin{aligned}\mathbb{E}[n_{i,k}(T)] &\leq \mathbb{E}[n_{i,k}(t')] + \mathbb{E}[n_{i,k}(T) - n_{i,k}(t')] \\ &< \max\left\{\frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2} \log T, L, (3M+1)N\right\} + \frac{2\pi^2}{3}.\end{aligned}$$

So the regret of agent until time T satisfies

$$\begin{aligned}R_i(T) &= \sum_{\Delta_k>0} \Delta_k \cdot \mathbb{E}[n_{i,k}(T)] \\ &< \sum_{\Delta_k>0} \Delta_k \left(\max\left\{\frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2} \log T, L, (3M+1)N\right\} + \frac{2\pi^2}{3} + 2p_0 \cdot n_{i,k}(t)\right) \\ &< \sum_{\Delta_k>0} \Delta_k \left(\max\left\{\frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2} \log T, L, (3M+1)N\right\} + \frac{2\pi^2}{3} + \frac{12}{(1-\lambda_2^{\frac{1}{3}}) \log \lambda_2^{-1}}\right).\end{aligned}$$

Let $C_2 = \frac{2\pi^2}{3} + \frac{12}{(1-\lambda_2^{\frac{1}{3}}) \log \lambda_2^{-1}}$, we have

$$R_i(t) < \sum_{\Delta_k>0} \Delta_k \left(\max\left\{\frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2} \log T, L, (3M+1)N\right\} + C_2\right).$$

□

Algorithm 2: Create partial sums

Input: Observations $\{X(\tau)\}_{\tau=1}^{t}$, $q(t) = t$, $p = 0$

Output: P partial sums: $\{\widehat{X}_1^{\mathsf{ps}}, \widehat{X}_2^{\mathsf{ps}}, \cdots, \widehat{X}_P^{\mathsf{ps}}\}$

while $t \neq 0$ **do**
  $d = \max\{i : \mathsf{bin}(t)[i] = 1\}$ *// convert t as a binary string and find the rightmost digit that is 1*
  $\mathsf{bin}(t)[d] = 0, q(t) = \mathsf{int}(\mathsf{bin}(t))$ *// flip that digit to 0, convert is back to decimal and get $q(t)$*
  $\widehat{X}_p^{\mathsf{ps}} = \sum_{\tau=q(t)+1}^{t} \widehat{X}_{i,k}(\tau)$ *// get a partial sum*
  $t = q(t), p = p + 1$
end

We have Remark 1 for the order of regret $R_i(T)$.

Remark 1. *There are two important terms affecting the order of regret: $\frac{2N}{(\frac{1}{2}\Delta_k - C_1)^2}\log T$ and L. The order of the former term is $O(N \log T)$, and the order of the latter term does not depend on T since L is determined by λ_2 and N. Let $t = 6\gamma \log_{\lambda_2^{-1}} N$. The inequality $\lambda_2^{t/6}/(1-\lambda_2^{1/3}) < 2(Nt)^{-1}$ becomes:*

$$\frac{3N}{1-\lambda_2^{1/3}}\frac{1}{N^{\gamma}} < \frac{1}{\log_{\lambda_2^{-1}} N^{\gamma}}.$$

There always exists a positive γ such that the above inequality holds. Thus the order of $R_i(T)$ is $O(\max\{NM \log T, M \log_{\lambda_2^{-1}} N\})$.

As for the lower bound of the regret, consider a trivial case where the graph G is fully connected. Then easily we can show the regret of our algorithm will be lower bounded by an ideal setting where all agents will receive the reward information from everyone else simultaneously with no delay. This setting reduces to a centralized bandit setting and by calling the classical results we know the regret is lower bounded by $\Omega(\log T)$. It remains a challenging and interesting question to understand the tightness of our bound in terms of the number of agents N and the graph G.

4 FED_UCB: PRIVACY PRESERVING GOSSIP_UCB

Noting directly leaking some information that might appear to be "anonymized" can be used to cross-reference with other datasets to breach privacy [36], we seek for a solution with worst-case privacy guarantees (even with arbitrarily power adversary). When guaranteeing an ϵ-differential privacy in one-shot, adding Laplacian noise $\gamma \sim \mathsf{Lap}(\frac{1}{\epsilon})$ to the observation often suffices, where a larger ϵ indicate a lower privacy level. However, preserving privacy in a sequential setting is in general hard due to the continual and sequential revelation of observations. That is, in addition to preserving the privacy of $X_{i,k}(t)$, we also need to protect it in each $\tau = t, t+1, ..., T$ steps.

To preserve at least ϵ-DP in T time slots, a naive extension of the Laplace mechanism [10] is adding Laplacian noise $\mathsf{Lap}(\frac{T}{\epsilon})$ to each observation $X_{i,k}(t)$. The noise introduced in each time step grows linearly w.r.t. T, i.e. $O(\frac{T}{\epsilon})$. To add a mild noise and maintain the same privacy level at the same time, we apply the partial sums idea [7] to $\tilde{X}_{i,k}(t)$. Since both the gossiping information $\theta_{i,k}(t)$ and the selection information $n_{i,k}(t)$ are functions of $X_{i,k}(t)$, this approach also preserves the privacy for Gossip_UCB by data processing inequality.

Algorithm 3: Add Laplacian Noise to Partial Sums

Input: Observations $\{X(\tau)\}_{\tau=1}^{t}$, mapping from partial sums to noise: $\mathcal{P}$, privacy level ϵ'

Get a list with P partial sums: $\{\widehat{X}_1^{\mathsf{ps}}, \widehat{X}_2^{\mathsf{ps}}, \cdots, \widehat{X}_P^{\mathsf{ps}}\}$ using Algorithm 2, $\tilde{X} = 0$

for $p = 1, \ldots, P$ **do**

  $\mathbb{1}_{\mathsf{p}} = \mathbb{1}(\widehat{X}_p^{\mathsf{ps}}$ is defined in $\mathcal{P})$ *// set as 1 when Laplacian noise is already added to* $\widehat{X}_p^{\mathsf{ps}}$

  / add i.i.d Laplacian noise to partial sum* $\widehat{X}_p^{\mathsf{ps}}$ *if it appears for the first time */*

  $\tilde{X} \leftarrow \tilde{X} + \widehat{X}_p^{\mathsf{ps}} + \mathbb{1}_{\mathsf{p}}\mathcal{P}(\widehat{X}_p^{\mathsf{ps}}) + (1 - \mathbb{1}_{\mathsf{p}})\mathsf{Lap}(1/\epsilon')$

end

Return $\tilde{X}/n_{i,k}(t)$

Algorithm 4: Fed_UCB

Input: $G, T, C_{i,k}(t)$

Initialization: Each agent pulls each arm exactly once, and receives a reward $X_{i,k}(0)$, $i \in [N]$, $j \in [M]$. Set $n_{i,k}(0) = 1$, $\vartheta_{i,k}(0) = \tilde{X}_{i,k}(0) = X_{i,k}(0)$.

for $t = 1, \ldots, T$ **do**

  $\mathcal{A}_i = \varnothing$

  $n_{i,k}(t) = n_{i,k}(t-1), \forall k \in [M]$

  $\tilde{n}_{i,k}(t+1) = \max\{n_{i,k}(t), \tilde{n}_{j,k}(t), j \in \mathcal{N}_i\}, \forall k \in [M]$

  Put k into set $\mathcal{A}_i$ **if** $n_{i,k}(t) < \tilde{n}_{i,k}(t) - N, \forall k \in [M]$ *// local consistency requirements*

  if $\mathcal{A}_i$ *is empty* **then**

    for $k = 1, \ldots, M$ **do**

      $Q_{i,k}(t) = \vartheta_{i,k}(t-1) + C_{i,k}(t)$ *// update the belief on each arm,* $C_{i,k}(t)$ *is chosen as (16)*

      $a_i(t) = \arg\max_k Q_{i,k}(t)$ *// select the best arm to pull*

    end

  else

    $a_i(t)$ is randomly selected from $\mathcal{A}_i$

  end

  / Lines 16, 17: Get DP-observations with input* $\{\widehat{X}_{i,k}(\tau)\}_{\tau=1}^{t}$, $\mathcal{P}$, *and* ϵ **/*

  Get noisy observation $\tilde{X}_{i,a_i(t)}(t)$ following Algorithm 3, update $\mathcal{P}$

  $\tilde{X}_{i,k}(t) = \tilde{X}_{i,k}(t-1), \forall k \neq a_i(t)$

  if *agent* i *is selected to gossip with agent* j **then**

    agent i sends $\vartheta_{i,k}(t-1)$ to agent j, at the same time, receives $\vartheta_{j,k}(t-1)$ from agent j

    $\vartheta_{i,k}(t) = \frac{\vartheta_{i,k}(t-1) + \vartheta_{j,k}(t-1)}{2} + \tilde{X}_{i,k}(t) - \tilde{X}_{i,k}(t-1)$ *// gossiping update*

  else

    $\vartheta_{i,k}(t) = \vartheta_{i,k}(t-1) + \tilde{X}_{i,k}(t) - \tilde{X}_{i,k}(t-1)$ *// normal update*

  end

end

Denote by $\widehat{X}_{i,k}(\tau) := \mathbb{1}(a_i(\tau) = k)X_{i,a_i(\tau)}(\tau)$. The partial sum is constructed as in Algorithm 2. As a result, $\tilde{X}_{i,k}(t)$ can be written as the summation of no more than $\lceil \log t \rceil$ partial sums:

$$\tilde{X}_{i,k}(t) = \frac{1}{n_{i,k}(t)}\left(\sum_{\tau=q(t)+1}^{t} \widehat{X}_{i,k}(\tau) + \sum_{\tau=q(q(t))+1}^{q(t)} \widehat{X}_{i,k}(\tau) + \ldots + \sum_{\tau=1}^{q(\cdots(q(t)))} \widehat{X}_{i,k}(\tau)\right).$$

Independent Laplacian noise $\gamma \sim \mathsf{Lap}(1/\epsilon')$ is added to each partial sum if there exists observations in that partial sum. Thus the total privacy guarantee is given by $\lceil \log t \rceil \epsilon$. Set $\epsilon' := \epsilon \frac{1}{\lceil \log T \rceil}$, where ϵ is a pre-set privacy level we want to achieve. Then we will achieve at least a total $\epsilon \frac{1}{\lceil \log T \rceil} \lceil \log t \rceil \leq \epsilon$ differential privacy. The algorithm for adding Laplacian noise to partial sums is shown in Algorithm 3. To make the algorithm preserve ϵ-DP, we need to modify the observation procedure in line 14 of Algorithm 1. Specifically, instead of directly getting $X_{i,a_i(t)}(t)$, Algorithm 3 is implemented with observations $\{\widehat{X}_{i,k}(\tau)\}_{\tau=1}^{t}$, the mapping $\mathcal{P}$ recording the noise added to the previous partial sums, and the privacy level ϵ. Besides, the confidence bound is set following (16). Note `Gossip_UCB` is a special case of `Fed_UCB` when ϵ is infinity.

We summarize `Fed_UCB` in Algorithm 4. Note the time complexity of line 16 is $O(\log t)$.

With the existence of Laplacian noise, Theorem 1 can be extended for `Fed_UCB` as follows.

Theorem 2. *For the ϵ-differentially private* `Fed_UCB` *algorithm with bounded reward over* $[0, 1]$, *and*

$$C_{i,k}(t) = C_1 + \sqrt{2N\left(\frac{128N\log^2 T \log n_{i,k}(t) \log t}{n_{i,k}^2(t)\epsilon^2} + \frac{1}{n_{i,k}(t)}\right)\log t}, \tag{16}$$

the regret of each agent i until time T satisfies the bound

$$R_i(T) < \sum_{\Delta_k > 0} \Delta_k \left(\max\left\{ \frac{N \log T \left(1 + \sqrt{1 + \left(16(\frac{\Delta_k}{2} - C_1)/\epsilon\right)^2 \log^3 T}\right)}{(\frac{\Delta_k}{2} - C_1)^2}, L, (3M+1)N \right\} + C_2 + 2N \right). \tag{17}$$

Proof. Now we are onto the noise analysis. Recall $\epsilon' := \epsilon \frac{1}{\lceil \log T \rceil}$. Up to each time t, there are at most $\lceil \log t \rceil$ noise terms added to each $\tilde{X}_{i,k}(t)$. Let $\gamma_\tau \sim \mathsf{Lap}(1/\epsilon')$ and

$$\Gamma := \sum_{\tau=1}^{\lceil \log t \rceil} \gamma_\tau.$$

The sum of these Laplace noise terms satisfy the following inequality [10]:

$$\mathbb{P}(\Gamma \geq \lambda) \leq \exp\left(-\frac{\lambda^2}{8\sum_\tau \frac{1}{\epsilon'^2}}\right)$$

Set $\lambda := \frac{4 \cdot \log T \cdot \sqrt{\log t \cdot \log n_{i,k}(t)}}{\epsilon}$, we have

$$\begin{aligned}
\exp\left(-\frac{\lambda^2}{8\sum_\tau \frac{1}{\epsilon'^2}}\right) &\leq \exp\left(-\frac{\lambda^2}{8\log t \frac{\log^2 T}{\epsilon^2}}\right) \\
&= \exp\left(-\frac{\frac{16\log^2 T \log t \log n_{i,k}(t)}{\epsilon^2}}{8\log t \frac{\log^2 T}{\epsilon^2}}\right) \\
&= n_{i,k}^{-2}(t).
\end{aligned}$$

That is with probability at most $\frac{1}{n_{i,k}^2(t)}$, the noise term added to each empirical mean $\tilde{X}$ is bounded by $\frac{\lambda}{n_{i,k}(t)}$.

Now we want to figure out how the added noise would affect the regret, more specifically, the $\vartheta_{i,k}(t)$. According to (10), $\vartheta_{i,k}(t)$ is a linear combination of $\tilde{X}_{i,k}(t)$, denote $\Delta\vartheta_{i,k}(t)$ as the total noise added in $\tilde{\vartheta}_{i,k}(t)$, $S_{i,k}(t)$ as the cumulative noise added in $\sum_{\tau=1}^{t}\tilde{\theta}(\tau)$. Then $\Delta\vartheta_{i,k}(t)$ is a linear combination of $S_{i,k}(t)$ and we have with probability at least $1-p_1$,

$$\begin{aligned}
&|\Delta\vartheta_{i,k}(t)| \\
=&\Bigg|\sum_j\Bigg(\left[W(t)\cdots W(\tau_1+1)-W(t)\cdots W(\tau_2+1)\right]_{i,j}S_{j,k}(\tau_1)+\cdots \\
&+\frac{\left[W(t)\cdots W(\tau_{n_{j,k}(t)-1}+1)-W(t)\cdots W(\tau_{n_{j,k}(t)}+1)\right]_{i,j}}{n_{j,k}(t)-1} \\
&\cdot S_{j,k}(\tau_{n_{j,k}(t)-1})+\frac{\left[W(t)\cdots W(\tau_{n_{j,k}(t)}+1)\right]_{i,j}}{n_{j,k}(t)}S_{j,k}(\tau_{n_{j,k}(t)})\Bigg)\Bigg| \\
\le&\sum_j|c_{i,k,j}^{(0)}|\cdot 4\log T\sqrt{\log n_{i,k}(t)\log t}/\epsilon \\
\le&\frac{8N}{n_{i,k}(t)}\log T\sqrt{\log n_{i,k}(t)\log t}/\epsilon,
\end{aligned}$$

where

$$p_1=2Nn_{i,k}(t)\cdot\frac{1}{n_{i,k}(t)^2}+2p_0=\frac{2N}{n_{i,k}(t)}+2p_0.$$

Define $\tilde{\sigma}_{i,k}^2(t)$ as the variance proxy of $\tilde{\vartheta}_{i,k}(t)$. Since noise and reward are always independent, $\Delta\vartheta_{i,k}(t)$ should also be independent with each $X_{j,k}(\tau)$, for all $j\in[N],\tau\in\{0,\dots,t\}$. Thus when $n_{i,k}(t)\ge\max\{L,(3M+1)N\},\forall i\in[N]$:

$$\tilde{\sigma}_{i,k}^2(t)=\sigma_{i,k}^2(t)+\sigma_\Delta^2<\frac{64N^2\log^2 T\log n_{i,k}(t)\log t}{n_{i,k}^2(t)\epsilon^2}+\frac{N}{2n_{i,k}(t)}.$$

Note $\mathrm{E}[\Delta\vartheta_{i,k}(t)]=0$. According to (15), we have

$$|\mathrm{E}[\tilde{\vartheta}_{i,k}(t)]-\mu_k|=|\mathrm{E}[\vartheta_{i,k}(t)]-\mu_k|<\lambda_2^t.$$

Set

$$C_{i,k}(t)=C_1+\sqrt{2N\left(\frac{128N\log^2 T\log n_{i,k}(t)\log t}{n_{i,k}^2(t)\epsilon^2}+\frac{1}{n_{i,k}(t)}\right)\log t},$$

then we have (with probability at least $1-p_1$)

$$\mathbb{P}(\tilde{\vartheta}_{i,k}(t)-\mu_k\ge C_{i,k}(t))\le\mathbb{P}(\tilde{\vartheta}_{i,k}(t)-\mathrm{E}[\tilde{\vartheta}_{i,k}(t)]\ge C_{i,k}(t)-C_1)<\frac{1}{t^2},$$

and similarly,

$$\mathbb{P}(\mu_k-\tilde{\vartheta}_{i,k}(t)\ge C_{i,k}(t))<\frac{1}{t^2}.$$

Notice

$$C_{i,k}(t) < \sqrt{\left(\frac{128N\log^4 T}{n_{i,k}(t)\epsilon^2}+1\right)\cdot\frac{2N\log t}{n_{i,k}(t)}} + C_1,$$

following the similar steps in the analyses of Gossip_UCB, we get with probability at least $1-p_1$, after time t' which makes

$$n_{i,k}(t') = \frac{N\log T\left(1+\sqrt{1+\left(16(\frac{\Delta_k}{2}-C_1)/\epsilon\right)^2\log^3 T}\right)}{(\frac{\Delta_k}{2}-C_1)^2}$$

hold, the expected error resulting from agent i selecting arm k is bounded by $\frac{2\pi^2}{3}$. So the expected number of non-optimal arms is bounded as

$$\mathrm{E}[n_{i,k}(T)] < \max\left\{\frac{N\log T\left(1+\sqrt{1+\left(16(\frac{\Delta_k}{2}-C_1)/\epsilon\right)^2\log^3 T}\right)}{(\frac{\Delta_k}{2}-C_1)^2}, L, (3M+1)N\right\} + \frac{2\pi^2}{3}. \quad (18)$$

Using union bound, the regret of agent i until time T is bounded as

$$\begin{aligned} R_i(t) &< \sum_{\Delta_k>0}\Delta_k\left(\max\left\{\frac{N\log T\left(1+\sqrt{1+\left(16(\frac{\Delta_k}{2}-C_1)/\epsilon\right)^2\log^3 T}\right)}{(\frac{\Delta_k}{2}-C_1)^2}, L, (3M+1)N\right\} + \frac{2\pi^2}{3} + p_1\cdot n_{i,k}(t)\right) \\ &< \sum_{\Delta_k>0}\Delta_k\left(\max\left\{\frac{N\log T\left(1+\sqrt{1+\left(16(\frac{\Delta_k}{2}-C_1)/\epsilon\right)^2\log^3 T}\right)}{(\frac{\Delta_k}{2}-C_1)^2}, L, (3M+1)N\right\} + C_2 + 2N\right). \end{aligned} \quad (19)$$

□

Remark 2. *The order of $R_i(T)$ is $O(\max\{NM\log^{2.5} T/\epsilon, M\log_{\lambda_2^{-1}} N\})$.*

5 EXPERIMENTS

In addition to the theoretical guarantees, we test the performance of Fed_UCB in multi-agent stochastic multi-arm bandits problems on both synthetic datasets and real datasets. Each experiment is run for 100 times with i.i.d. noise and the regret is averaged over 100 trials and N agents. Shadow areas indicate the range of regret fluctuations (minimum and maximum in 100 trials). When $\epsilon = \infty$, there is no extra privacy preservation approach as discussed in Gossip_UCB. To show the effect of ϵ-DP preservation on the regret of Fed_UCB, we add Laplacian noise following Algorithm 3. The total privacy levels are set as $\epsilon = 1, 2, 5$, respectively.

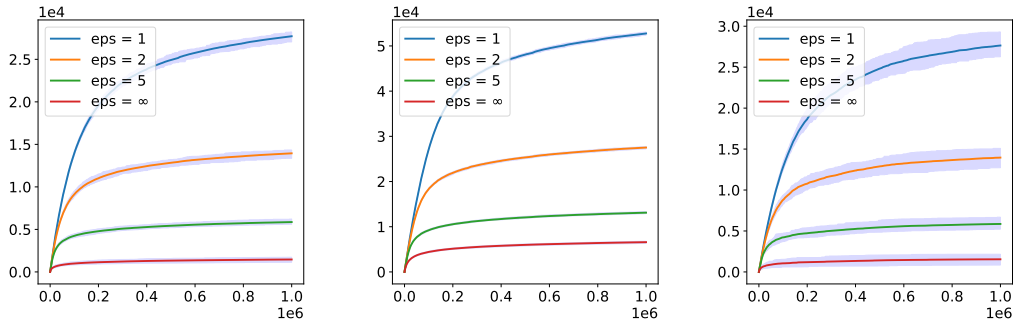

Fig. 1. Regret of Fed_UCB over time when the algorithm preserves ϵ-DP. Higher ϵ indicates lower privacy level. (a) Synthetic data with $N = 3, M = 5, \lambda_2 = 0.5$; (b) Synthetic data with $N = 10, M = 10, \lambda_2 \approx 0.99$; (c) Real hospital data with $N = 3, M = 5, \lambda_2 = 0.5$.

5.1 Synthetic datasets

In the synthetic data, the expected belief of agent i toward arm k, i.e. $\mu_{i,k}$, is generated randomly in $[0, 1]$. The rank of arms from each agent's perspective is different. Zero-mean Gaussian noise with variance 1 is added to each observation. Figure 1(a) and Figure 1(b) show the simulation results in different settings. When the increase of regret is stable, e.g. t =6e5, in both figures, the ratios of regret with $\epsilon_i \in \{1, 2, 5\}$ are approximately $1 : 1/2 : 1/5 = 1/\epsilon_1 : 1/\epsilon_2 : 1/\epsilon_3$, which is consistent with our bound on the regret of Fed_UCB.

5.2 Real-world datasets

We use the UCI diabetes dataset [35], which is a medical dataset including 101,766 inpatient records in US hospitals. There are over 50 features representing patient and hospital outcomes, e.g., results of some laboratory tests, personal information of patients, and the specific medications administered during the encounter. The label encodes three readmission states of patients: no record, readmitted in more than 30 days, and readmitted in less than 30 days.

Construction of arms: There are 23 diabetes medications: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, Each medication contains four sub-features, i.e., *No*, *Down*, *Up*, *Steady*, indicating whether the drug was prescribed or there was a change in the dosage. The number of each sub-feature is pretty unbalanced. Generally, there are much less samples with sub-features *Down* and *Up* compared with *No* and *Steady*. To ensure a sufficient number of samples, we only focus on the sub-feature *No* and *Steady*. Additionally, noting the number of *Steady* samples in different medications is also very imbalanced, we combine some medications with similar rewards and get 5 arms. They are:

- *Arm* 1: Use steady dosage of insulin;
- *Arm* 2: Use steady dosage of metformin;
- *Arm* 3: Use steady dosage of repaglinide,glipizide, rosiglitazone, acarbose, miglitol or troglitazone;
- *Arm* 4: Use steady dosage of nateglinide, chlorpropamide, glyburide, examide, glimepiride,

acetohexamide, glyburide, tolbutamide, pioglitazone, sitagliptin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, or metformin-pioglitazone;
• *Arm* 5: No action in any medications.

Construction of rewards: There are 3 readmission states: *no record*, *readmitted in more than* 30 *days*, and *readmitted in less than* 30 *days*. If the patient is readmitted in less than 30 days, we can infer the treatment (medication) does not work. On the contrary, if the patient is readmitted in more than 30 days, we can infer the treatment (medication) works in some sense. If there is no record for this patient, it may because the patient chooses other hospitals or is cured, which makes it difficult to determine the effectiveness of the treatment. Additionally, we cannot simply discard these samples since they account for a large population. In our experiments, we assume *no record* means the treatment does not work well. There are several evidences:
• *Evidence 1*: Diabetes are chronic diseases, it is hard to be cured in one hospital stay;
• *Evidence 2*: By simple logistic regression, we can find the readmission states are highly-related to the discharge disposition, including transferring other hospitals or medical institutions. This phenomenon supports our speculation that *no record* mainly shows bad treatments;
• *Evidence 3*: From our numerical results, assuming *no record* indicating the positive effect of treatments leads to a unreasonable result: nearly all steady medications have negative effects.

Therefore, we map *no record* and *readmitted in less than* 30 *days* to reward 0, and map *readmitted in more than* 30 *days* to reward 1.

In this experiment, we focus on the relationship between medications and readmission states. Noting no one wants to be known about their readmission status, we consider a federated bandit setting where hospitals (agents) do not directly share their rewards (readmission states) due to privacy concerns. On one hand, the privacy can be protected in some sense the gossiping procedure since the variable for gossiping is a weighted combination of the protected data with complicated (and possibly unknown) weights. On the other hand, the hospital can only use a noisy copy of readmission states in the DP setting. Each hospital protects their patient's privacy in this mechanism[3]. We map 23 types of medications into 5 arms, and randomly divide the samples into 3 equal-sized set belonging to three hospitals (agents). From our numerical results, local biased indeed occurs when the dataset is separated into several parts according to the order in which the samples appear. We leave detailed data processing procedures in supplementary materials. The feasibility of federated bandit in a real hospital dataset is testified in Figure 1(c).

6 CONCLUSION

In this paper, we have proposed `Gossip_UCB` for solving a gossiping bandit learning problem, where a network of agents aim to learn to converge to selecting the best arm both locally and globally through gossiping, and its differentially private variant, `Fed_UCB`, for preserving ϵ-differential privacy of the agents' local data. We have shown that both `Gossip_UCB` and `Fed_UCB` achieve weak regret at an order depending on the size of agents, the number of arms, time horizon, and connectivity of the graph. In addition to the theoretical bounds on regret, experiments on both synthetic data and real data also verify the feasibility of the proposed gossiping approach of federated bandit.

[3]To further protect the patient's privacy, the Laplacian noise can be added by some trust-worthy institutions such that hospitals cannot get the clean data.

REFERENCES

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

[2] Ilai Bistritz and Amir Leshem. 2018. Distributed Multi-Player Bandits - a Game of Thrones Approach. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7222–7232.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482* (2016).

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. 2006. Randomized Gossip Algorithms. *IEEE Transactions on Information Theory* 52, 6 (2006), 2508–2530.

[5] M. Cao, D. A. Spielman, and A. S. Morse. 2005. A lower bound on convergence of a distributed network consencus algorithm. In *Proceedings of the 44th IEEE Conference on Decision and Control*. 2356–2361.

[6] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. 2017. Coordinated Versus Decentralized Exploration In Multi-Agent Multi-Armed Bandits.. In *IJCAI*. 164–170.

[7] T-H Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)* 14, 3 (2011), 26.

[8] Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Clémençon. 2015. Extending Gossip Algorithms to Distributed Estimation of U-Statistics. In *Advances in Neural Information Processing Systems*. 271–279.

[9] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*. 3571–3580.

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

[11] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[12] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.

[13] István Hegedűs, Gábor Danner, and Márk Jelasity. 2019. Gossip learning as a decentralized alternative to federated learning. In *IFIP International Conference on Distributed Applications and Interoperable Systems*. Springer, 74–90.

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

[15] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. 2014. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory* 60, 4 (2014), 2331–2345.

[16] A. Kashyap, T. Başar, and R. Srikant. 2007. Quantized consensus. *Automatica* 43, 7 (2007), 1192–1203.

[17] David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE, 482–491.

[18] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*. https://arxiv.org/abs/1610.05492

[19] Satish Babu Korada, Andrea Montanari, and Sewoong Oh. 2011. Gossip PCA. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 209–220.

[20] P. Landgren, V. Srivastava, and N. E. Leonard. 2016. Distributed Cooperative Decision-making in Multiarmed Bandits: Frequentist and Bayesian Algorithms. In *Proceedings of the 55th IEEE Conference on Decision and Control*. 167–172.

[21] Qinbin Li, Zeyi Wen, and Bingsheng He. 2019. Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693* (2019).

[22] Keqin Liu and Qing Zhao. 2010. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing* 58, 11 (2010), 5667–5681.

[23] Yang Liu, Ji Liu, and Tamer Basar. 2018. Differentially private gossip gradient descent. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2777–2782.

[24] Mohammad Malekzadeh, Dimitrios Athanasakis, Hamed Haddadi, and Ben Livshits. 2020. Privacy-Preserving Bandits. In *Proceedings of Machine Learning and Systems 2020*. 350–362.

[25] David Martínez-Rubio, Varun Kanade, and Patrick Rebeschini. 2019. Decentralized Cooperative Stochastic Bandits. In *Advances in Neural Information Processing Systems*. 4531–4542.

[26] Nikita Mishra and Abhradeep Thakurta. 2014. Private Stochastic Multi-arm Bandits: From Theory to Practice.

[27] L. Moreau. 2005. Stability of multi-agent systems with time-dependent communication links. *IEEE Trans. Automat. Control* 50, 2 (2005), 169–182.
[28] N. Nayyar, D. Kalathil, and R. Jain. 2016. On Regret-optimal Learning in Decentralized Multi-player Multi-armed Bandits. *IEEE Transactions on Control of Network Systems* 5, 1 (2016), 597–606.
[29] A. Olshevsky and J. N. Tsitsiklis. 2009. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization* 48, 1 (2009), 33–55.
[30] Kristiaan Pelckmans and Johan AK Suykens. 2009. Gossip algorithms for computing U-statistics. *IFAC Proceedings Volumes* 42, 20 (2009), 48–53.
[31] W. Ren and R. W. Beard. 2005. Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies. *IEEE Trans. Automat. Control* 50, 5 (2005), 655–661.
[32] Joshua Romoff, Nicolas Ballas, Joelle Pineau, Mike Rabbat, et al. 2019. Gossip-based Actor-Learner Architectures for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*. 13299–13309.
[33] Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. 2019. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 3 (2019), 1–35.
[34] Benjamin Sirb and Xiaojing Ye. 2018. Decentralized consensus algorithm with delayed and stochastic gradients. *SIAM Journal on Optimization* 28, 2 (2018), 1232–1254.
[35] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. 2014. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international* 2014 (2014).
[36] Latanya Sweeney. 2000. Simple demographics often identify people uniquely. *Health (San Francisco)* 671, 2000 (2000), 1–34.
[37] Balazs Szorenyi, Róbert Busa-Fekete, István Hegedus, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. 2013. Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*. 19–27.
[38] Aristide CY Tossou and Christos Dimitrakakis. 2015. Differentially private, multi-agent multi-armed bandits. In *European Workshop on Reinforcement Learning (EWRL)*.
[39] Aristide CY Tossou and Christos Dimitrakakis. 2016. Algorithms for differentially private multi-armed bandits. In *Thirtieth AAAI Conference on Artificial Intelligence*.
[40] B. Touri and A. Nedić. 2014. Product of random stochastic matrices. *IEEE Trans. Automat. Control* 59, 2 (2014), 437–448.
[41] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. 2020. Distributed Bandit Learning: Near-Optimal Regret with Efficient Communication. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJxZnR4YvB
[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

The analyses are based on sub-Gaussian random variables. We present the following preliminaries.

A PRELIMINARIES OF SUB-GAUSSIAN RANDOM VARIABLES

Definition 2. *A random variable X with $\mu = \mathbb{E}[X]$ is called σ^2 sub-Gaussian if there is a positive σ such that*

$$\mathbb{E}[e^{\lambda(X-\mu)}] \leq e^{\frac{\sigma^2\lambda^2}{2}}, \ \forall \lambda \in \mathbb{R},$$

where such σ^2 is called a variance proxy, and the smallest variance proxy is called the optimal variance proxy.

Property 1. *(Inequality) A sub-Gaussian random variable X satisfies:*

$$\mathbb{P}(X - \mu \geq a) \leq e^{-\frac{a^2}{2\sigma^2}},$$

$$\mathbb{P}(\mu - X \geq a) \leq e^{-\frac{a^2}{2\sigma^2}}.$$

Property 2. *(Sufficient condition) If X is a random variable with finite mean μ and $a \leq X \leq b$ almost surely, then X is $\frac{(b-a)^2}{4}$ sub-Gaussian.*

Property 3. *(Additivity) If X_1 is σ_1^2 sub-Gaussian and for $2 \leq i \leq n$, $(X_i|X_1, \ldots, X_{i-1})$ is σ_i^2 sub-Gaussian with σ_i being free of $X_1, \ldots, X_{i-1}$, then $X_1 + \cdots + X_i$ is sub-Gaussian with $\sigma_1^2 + \cdots + \sigma_i^2$ being one of its variance proxy.*