Global and Local Differential Privacy for Collaborative Bandits

Huazheng Wang
hw7ww@virginia.edu
University of Virginia

Qian Zhao
qzhao101@bloomberg.net
Bloomberg L.P.

Qingyun Wu
qw2ky@virginia.edu
University of Virginia

Shubham Chopra
schopra31@bloomberg.net
Bloomberg L.P.

Abhinav Khaitan
akhaitan10@bloomberg.net
Bloomberg L.P.

Hongning Wang
hw5x@virginia.edu
University of Virginia

ABSTRACT

Collaborative bandit learning has become an emerging focus for personalized recommendation. It leverages user dependence for joint model estimation and recommendation. As such online learning solutions directly learn from users, e.g., result clicks, they bring in new challenges in privacy protection. Despite the existence of recent studies about privacy in contextual bandit algorithms, how to efficiently protect user privacy in a collaborative bandit learning environment remains unknown.

In this paper, we develop a general solution framework to achieve differential privacy in collaborative bandit algorithms, under the notion of global differential privacy and local differential privacy. The key idea is to inject noise in a bandit model's sufficient statistics (either on server side to achieve global differential privacy or client side to achieve local differential privacy) and calibrate the noise scale with respect to the *structure of collaboration* among users. We study two popularly used collaborative bandit algorithms to illustrate the application of our solution framework. Theoretical analysis proves our derived private algorithms reduce the added regret caused by privacy-preserving mechanism compared to its linear bandits counterparts, i.e., collaboration actually helps to achieve *stronger privacy* with the same amount of injected noise. We also empirically evaluate the algorithms on both synthetic and real-world datasets to demonstrate the trade-off between privacy and utility.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; • **Theory of computation** → **Sequential decision making**; **Online learning algorithms**.

KEYWORDS

Differential privacy; collaborative learning; contextual bandits

ACM Reference Format:
Huazheng Wang, Qian Zhao, Qingyun Wu, Shubham Chopra, Abhinav Khaitan, and Hongning Wang. 2020. Global and Local Differential Privacy for Collaborative Bandits. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412254

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412254

1 INTRODUCTION

Recommender system is an indispensable component to improve user engagement in modern online information services, such as e-commerce, online advertisement and search engines. As a reference solution for recommendation, collaborative filtering based algorithms achieved impressive success in practice [21, 25, 31]. However, the rapid appearance of new information and new users together with the ever-changing nature of content relevance make traditional offline learning of collaborative filtering incompetent. This motivates the recent developments in online collaborative learning for recommendation, especially contextual bandit based algorithms [1, 15, 23]. Collaborative bandit algorithms provide a principled solution to the *explore-exploit* dilemma, and enjoy the benefits of collaborative learning paradigm, such as alleviating the *cold-start* challenge. Recent advances in collaborative bandits include modeling user dependency (e.g., social influence) [6, 37], online user or item clustering [16, 17, 24], and estimating a low-rank structure with latent factors (i.e., matrix factorization based collaborative filtering) [20, 35, 36].

Nevertheless, personalized recommendation is a double-edged sword: the gained utility also comes with the risk of privacy violation. Overly personalized recommendations could be a potential source of privacy vulnerability, for adversaries to take advantage of, e.g., infer users' sensitive information. Real-world privacy breaches have been reported in Amazon's recommendation system [5] and Facebook's advertisement system [22], where an adversary learns considerable amount of information about a user solely based on the systems' recommendation sequences. Comparing to the offline learnt models, online learning methods directly interact with sensitive user data, e.g., user clicks or purchasing history, and timely update the models to adjust their output, which makes privacy an even more serious concern [3, 30, 33, 34]. Realizing its importance, private online learning has recently attracted increasing attention in the research community, with a goal to prevent the algorithm's sequential output from revealing a user's private information. While there is existing research on differentially private online convex optimization [18, 33] and contextual bandits [29, 30], private collaborative bandits have not been explored yet.

The challenges regarding the risk of privacy breach in a collaborative bandit based recommender system are unique. In such a system, the algorithm recommends an item to a user, and the

user provides feedback (e.g., click) based on his/her true preference. The feedback (reward) is then used to update not only the model's reward estimation on this user, but also on other users via the imposed dependency among users. As a result, any change in one user's feedback promptly leads to changes in the algorithm's output, e.g., different sequences of recommended items, potentially for *all* users. This is originally designed to improve subsequent recommendations collectively across all users. But a user's private information could thus be inferred and revealed simply by releasing the recommendation sequence, e.g., extraction attack, even if this user's feedback is kept private in the system.

In this work, we propose the first study to equip collaborative bandit algorithms with privacy guarantees, under the notion of *global differential privacy* [11] and *local differential privacy* [10]. Under global differential privacy, a user is assumed to trust (or say he/she has to trust) the system and provide real engagement data to the system, and the system outputs private recommendations; while under local differential privacy, each user provides perturbed statistics to the system and is no longer required to trust the system or the communication between him/her and system. As the very first study on private collaborative bandits, we focus on algorithms that leverage *known dependency* (e.g., social connections) among users, such as [6, 37]. Specifically, these algorithms propagate the reward collected from one user to update his/her peers' bandit models, according to a given and fixed user dependency structure.

One common practice to achieve privacy guarantee is to inject noise to perturb certain statistics derived from private information in the learning process, either on the server side to achieve global differential privacy or on the client side to achieve local differential privacy [8, 10, 11]. However, how to efficiently inject noise in the collaborative bandit learning setting is non-trivial, because of the inherent information sharing mechanism. Specifically, to preserve privacy in collaborative bandits, we apply the tree-based mechanism [7, 12] to add Laplace noise to the models' statistics to guarantee privacy on each user's *reward* feedback (e.g., user clicks). We conduct sensitivity analysis, to which the key is to calibrate the noise scale with respect to the *structure of collaboration* defined by the user dependency graph. Our insight is that a careful sensitivity analysis over the collaboration structure offers the opportunity to inject minimum amount of noise and better balance the *privacy and utility trade-off*. In this work, we study two popularly employed collaborative bandit algorithms, Collaborative LinUCB (CoLin) [37] and Gang of Bandits (GOBLin) [17], as the baseline algorithms, which represent two classic types of social network based collaboration structure. We develop their private versions to illustrate a general solution framework for private collaborative bandit. We prove the private algorithms reduce the added regret caused by privacy-preserving mechanism compared to its linear bandits counterparts, i.e., collaboration actually helps to achieve stronger privacy with the same amount of injected noise. We also empirically evaluate the algorithms on both synthetic and real-world public datasets to validate its effectiveness and show the improved trade-off between utility and privacy from our proposed solution framework.

2 RELATED WORK

Collaborative Bandits. Rooted in contextual bandits [1, 4, 23], collaborative bandit algorithms are recently developed to alleviate the cold-start problem in online recommendation. Wu et al. [37] modeled dependency among users (e.g., social influence) through a collaborative reward generation assumption; and Cesa-Bianchi et al. [6] leveraged the structure of user dependency as model regularization, where connected users are assumed to have similar model parameters. This type of collaborative bandits require the knowledge of user dependency structure beforehand. Correspondingly, online clustering of bandits studied in [17] avoided such a requirement. Li et al. [24] extended the online clustering to both users and items for collaborative filtering. Matrix factorization based collaborative bandits have been studied in [20, 35, 36], where the collaboration is achieved via a low rank structure over user and item latent factors. In this paper, we focus on privacy guarantees for the first type of collaborative bandits, which take a known collaboration structure as input, and leave the exploration of other types of collaborative bandits as our future work.

Differential privacy. Differential privacy [11] provides a formal notion to quantify the amount of information an adversary could obtain by observing the algorithm's output. The common practice is to add Laplace or Gaussian noise to the output; and the scale of noise depends on privacy budget (often denoted as ϵ) and *sensitivity*, which is the change of an algorithm's output caused by the change of input. Prior work has studied the problem of differential privacy for offline collaborative filtering methods [19, 26, 27, 38].

Differential privacy was first extended to an online setting for stream data in [7, 12]. Differentially private online learning methods have been studied for online convex optimization [2, 3, 33] and bandit problems [28–30, 34]. The key technique of these solutions is the *tree-based mechanism*, which was proposed in [7, 12] for privately releasing *sum* statistics in stream data with finite time horizon T. Its key idea is to maintain a noisy binary tree where the T leaf nodes are the data points, and the internal nodes in the tree stores the sum of all the leaves in its sub-tree. Each node (which represents a partial sum) in the tree is protected with $\frac{\epsilon}{\log(T)}$-differential privacy. Since each sum statistic can be rewritten into $\lceil \log(T) \rceil$ partial sums, composition theorem of differential privacy [27] guarantees the sequence of output sum statistics is ϵ-differentially private.

Based on this tree-based mechanism, (globally) differentially private linear bandit was first studied in [29] with guaranteed privacy in collected user reward feedback. However, it is non-trivial to extend the private linear bandits to collaborative bandits setting, where one user's reward feedback directly contributes to other users' model update. In other words, the change of model's input from one user can be measured by the model's output in (potentially) all users. This propagation of information has to be carefully reflected in sensitivity analysis to avoid trivial solutions.

3 PRELIMINARIES

To prepare for the discussion of our proposed differentially private collaborative bandit solution framework, we provide a brief overview of contextual bandits, collaborative contextual bandits, and differential privacy in this section.

3.1 Contextual Bandits

In a multi-armed bandit problem, an algorithm sequentially selects an arm a_t from a candidate pool $\mathcal{A} = \{a_1, ..., a_k\}$ at time t, and receives the corresponding reward r_{a_t}. The goal is to maximize its cumulative reward over a finite time horizon T. In a typical contextual bandit setting, each arm a is associated with a d-dimensional context vector $\mathbf{x}_a$ and its expected reward is governed by a conjecture of the context vector and an unknown bandit model, parameterized as $\boldsymbol{\theta}^*$. For example, in a linear contextual bandit setting, it is assumed that $r_a \sim N(\mathbf{x}_a^\intercal \boldsymbol{\theta}^*, \sigma^2)$. A bandit algorithm is evaluated by its pseudo-regret with respect to the optimal arm choice in T rounds of interactions, which is defined as,

$$\text{Regret}(T) = \sum_{t=1}^{T} \left(\mathbf{E}[r_{a_t^*}] - \mathbf{E}[r_{a_t}] \right) \tag{1}$$

where a_t^* is the optimal arm to select at time t according to $\boldsymbol{\theta}^*$. When selecting an arm, the algorithm needs to carefully balance the need for exploitation (trusting the current estimate of $\boldsymbol{\theta}^*$) and the need for exploration (testing new hypotheses to improve the estimation of $\boldsymbol{\theta}^*$).

3.2 Collaborative Contextual Bandits

When applied to personalized online recommendation settings, the unknown bandit model parameter $\boldsymbol{\theta}^*$ is usually attached to each user to reflect their corresponding personalized preferences. We use $\boldsymbol{\theta}_u^*$ to denote the personalized bandit model parameter for user u. In a vanilla contextual bandit setting, $\boldsymbol{\theta}_u^*$ for $u \in \mathcal{U}$ are independently estimated based on the observations from the corresponding users. In a collaborative environment, the expected reward on arm a from user u is assumed to be correlated with those from other users. Collaborative contextual bandit algorithms aim to leverage such dependency information for improved online model estimation and arm selection. Several different ways have been developed to utilize such dependency. The most effective ways can be roughly summarized into the following three categories: 1). additive weighted reward sharing [37]; 2). graph Laplacian based model regularization [6]; 3). online clustering based model sharing [16, 17, 24]. Among these three categories, the first two can be considered as *explicit* collaboration, as both of them require specific input about how information is shared across users; and the last one can be considered as *implicit* collaboration, since the structure of collaboration also needs to be inferred from observations by the algorithm. In this paper, we focus on the first two types of collaborative bandit algorithms, and elaborate their details later.

3.3 Differential Privacy

For a contextual bandit algorithm that interacts with users over time horizon T, denote $S = \{r_t\}_{t=1}^T$ as the reward sequence, where r_t is the reward feedback from user u_t at time t. S' is considered as an adjacent neighboring sequence of S, if it only differs from S at one point of reward r_i. The output of a bandit algorithm O (which is observed by the adversary) is the sequence of its selected arms, i.e., $\{a_t\}_{t=1}^T$.

Definition 1 (Global Differential Privacy (DP) [11]). *A randomized mechanism $\mathcal{M}$ is ϵ-differentially private if for any adjacent neighboring sequences $\{S, S'\}$ and output, $\mathbb{P}(\mathcal{M}(S) \in O) \leq e^\epsilon \mathbb{P}(\mathcal{M}(S') \in O)$.*

Global differential privacy ensures the adversary observes almost the same output from a private algorithm, in a probabilistic sense, if only one input data point is changed. The difference between the corresponding output is characterized by ϵ. Laplace or Gaussian noise is commonly introduced to disguise the output, where the noise scale is related to the privacy budget ϵ and the *sensitivity* of $\mathcal{M}$. We formally define sensitivity below.

Definition 2 (Sensitivity [11]). *For any adjacent neighboring sequences $\{S, S'\}$, global sensitivity of a function $f(\cdot)$ is defined as $\Delta_f = \max_{S,S'} |f(S) - f(S')|$.*

Global differential privacy protects sensitive user data from an adversary who has access to the algorithm's output. But it requires the user to send his/her authentic data to the server. Thus, the server and the communication between user and server have to be trusted. To lift the trust needed from the user, local differential privacy (LDP) is proposed [10]. The key idea is that the privacy mechanism needs to perturb the sensitive statistics on the client side before sending it to the server for further computation. Local differential privacy has been adopted in many real-world applications, such as the RAPPOR system developed by Google to collect web browsing behaviour [14], and Apple provides this privacy protection when collecting users' usage and typing history [32]. Note that the input and output of a local differential privacy mechanism could be different from the global differential privacy mechanism, even for the same problem, as they impose different privacy requirements. Let S_i be the reward sequence of user u_i such that $\bigcup_i S_i = S$. The formal definition of local differential privacy is provided below, where a user perturbs his/her private statistics S_i using mechanism L locally, and then send the noisy statistics to the server.

Definition 3 (Local Differential Privacy (LDP) [10]). *A randomized mechanism $\mathcal{M}$ is ϵ-locally differentially private if for any input $\{S_i, S_i'\}$ and output O, $\mathbb{P}(\mathcal{M}(S_i) \in O) \leq e^\epsilon \mathbb{P}\left(\mathcal{M}(S_i') \in O\right)$*

The key difference between LDP and DP is that a DP mechanism takes all users' data S as input and requires the output to be indifferentiable, while LDP mechanism takes only one user's data S_i as input and generates randomized responses per user (locally) for downstream tasks.

4 DIFFERENTIALLY PRIVATE COLLABORATIVE BANDITS: STARTING FROM COLIN

In this work, we aim to develop a general framework to guarantee global and local differential privacy for collaborative contextual bandits. Due to the intrinsic complexity of the problem, in this section we first develop the private version for a state-of-the-art collaborative bandit algorithm Collaborative LinUCB (CoLin) [37] as an example. To note, our solution framework is general and can be applied to other collaborative contextual bandits algorithms. In the next section, we will provide the full picture of our framework and show how it can be applied to another collaborative contextual bandit algorithm Gang of Bandits (GOBLin) [17] with minimum modification in the procedures and analysis.

4.1 Global Differential Privacy for CoLin

In Collaborative LinUCB (CoLin [37]), contextual bandit models are placed on a weighted graph $G = (\mathcal{V}, \mathcal{E})$, which encodes the affinity relationship among users. Specifically, each node $v_i \in \mathcal{V}$ in G hosts a bandit model parameterized by $\boldsymbol{\theta}_i$ for user i; and the edges in $\mathcal{E}$ represent the affinity relation over pairs of users. This graph is encoded as an $N \times N$ stochastic matrix $\mathbf{W}$, in which each element w_{ij} is nonnegative and proportional to the influence that user i has on user j. $\mathbf{W}$ is normalized such that $\sum_{i=1}^{N} w_{ij} = 1$ for $j \in \{1, ..., N\}$, and it is assumed to be time-invariant and known to the learner beforehand. Accordingly, CoLin postulates an *additive* reward generation assumption: the expected reward $\mathbf{E}[r_{a_t,u_t}]$ is not only determined by user u_t's own preference on the arm a_t, but also by that from the neighbors who have influence on u_t as $\mathbf{E}[r_{a_t,u_t}] = \sum_{j=1}^{N} w_{u_t j} \mathbf{x}_{a_t,j}^{\mathsf{T}} \boldsymbol{\theta}_j$; or equivalently this can be described as,

$$r_{a_t,u_t} \sim N\big(\mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})^{\mathsf{T}}\mathrm{Vec}(\boldsymbol{\Theta}), \sigma^2\big) \tag{2}$$

where $\mathrm{Vec}(\cdot)$ is the matrix vectorization operation, $\boldsymbol{\Theta}$ is a $d \times N$ matrix consisting of parameters from all the bandits in the graph: $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, and $\mathring{\mathbf{X}}_{a_t,u_t}$ is a $d \times N$ matrix with only the column corresponding to user u_t at time t set to $\mathbf{x}_{a_t,u_t}^{\mathsf{T}}$ and all the other columns set to zero. By defining $\tilde{\mathbf{x}}_{a_t,u_t} = \mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})$ and $\boldsymbol{\vartheta} = \mathrm{Vec}(\boldsymbol{\Theta})$, Eq (2) can be re-written as $r_{a_t,u_t} \sim N(\tilde{\mathbf{x}}_{a_t,u_t}^{\mathsf{T}}\boldsymbol{\vartheta}, \sigma^2)$.

With such a collaborative reward generation assumption, CoLin appeals to ridge regression for estimating the global bandit parameter matrix $\boldsymbol{\vartheta}_t$ over all the users at time t. It has a closed-form solution $\hat{\boldsymbol{\vartheta}}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$, in which $\mathbf{A}_t = \lambda\mathbf{I}_{dN} + \sum_{t'=1}^{t-1} \tilde{\mathbf{x}}_{a_{t'},u_{t'}}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}^{\mathsf{T}}$ and $\mathbf{b}_t = \sum_{t'=1}^{t-1} \tilde{\mathbf{x}}_{a_{t'},u_{t'}} r_{a_{t'},u_{t'}}$. $\mathbf{I}_{dN}$ is an identity matrix and λ is the trade-off parameter for the L2 regularization in ridge regression.

The required information sharing in CoLin brings unique challenges in protecting users' reward feedback, i.e., the change in one user's reward feedback can be effectively inferred from all users' observed recommendation sequences. The recommendation sequences for all users thus have to be perturbed to obtain differential privacy. But instead of directly adding noise to the model's output, i.e., its choice of arms, we choose to add noise η_t to the sufficient statistics $\mathbf{b}_t = \sum_{t'}^{t-1} \tilde{\mathbf{x}}_{a_{t'}} r_{t'}$ in CoLin, where we sample η_t from a tree-based mechanism [7, 12]. Because differential privacy is immune to post-processing [13], this ensures differential privacy on the algorithm's output. We name this private derivation of CoLin as (Globally) Differentially Private CoLin (DP-CoLin), and provide its details in Algorithm 1.

The key in DP-CoLin is to derive the sensitivity of CoLin. Analyzing sensitivity in a linear bandit is straightforward [29], as the sensitivity on $\mathbf{b}_t$ can be directly bounded by $\|\mathbf{x}_a\|_2|r_a - r'_a| \le L$, where the reward difference is bounded by 1 and the norm of context vector is bounded by L. However, for collaborative bandits, the context vectors encode user dependency and have a higher dimension $\tilde{\mathbf{x}}_{a,u} \in \mathbb{R}^{dN}$. A trivial bound is $\|\tilde{\mathbf{x}}_{a,u}\|_2|r_a - r'_a| \le NL$; but we argue this is not tight enough and unnecessarily introduces large noise. Below we analyze the privacy guarantee of DP-CoLin with a tighter sensitivity bound, which calibrates the noise with respect to the structure of collaboration embedded in $\mathbf{W}$.

Algorithm 1 Differentially Private CoLin (DP-CoLin)

1: **Inputs:** $\delta \in \mathbb{R}_+$, $\lambda \in [0, 1]$, $\mathbf{W} \in \mathbb{R}^{N\times N}$, Δ
2: **Initialize:** $\mathbf{A}_1 \leftarrow \lambda\mathbf{I}_{dN\times dN}$, $\mathbf{b}_1 \leftarrow \mathbf{0}$, $\hat{\boldsymbol{\vartheta}}_1^p \leftarrow \mathbf{A}_1^{-1}\mathbf{b}_1$,
3: **for** $t = 1$ to T **do**
4: Receive user u_t, observe context vectors, $\mathbf{x}_{a_t,u_t} \in \mathbb{R}^d$ and construct $\tilde{\mathbf{x}}_{a_t,u_t} = \mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})$ for $\forall a \in \mathcal{A}$
5: Take action $a_t = \arg\max_{a\in\mathcal{A}} \tilde{\mathbf{x}}_{a_t,u_t}^{\mathsf{T}}\hat{\boldsymbol{\vartheta}}_t^p + \alpha_t\sqrt{\tilde{\mathbf{x}}_{a_t,u_t}^{\mathsf{T}}\mathbf{A}_t^{-1}\tilde{\mathbf{x}}_{a_t,u_t}}$, where α_t is given by Lemma 2.
6: Observe payoff r_{a_t,u_t}
7: $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t + \tilde{\mathbf{x}}_{a_t,u_t}\tilde{\mathbf{x}}_{a_t,u_t}^{\mathsf{T}}$, $\mathbf{b}_{t+1} \leftarrow \mathbf{b}_t + \tilde{\mathbf{x}}_{a_t,u_t}r_{a_t,u_t}$
8: Sample noise $\eta_t \sim$ TreeMechanism(Δ, ϵ), in which $\Delta = \max_i L\|\mathbf{W}_i\|_2$
9: $\mathbf{b}_{t+1}^p \leftarrow \mathbf{b}_{t+1} + \eta_t$, $\hat{\boldsymbol{\vartheta}}_{t+1}^p \leftarrow \mathbf{A}_{t+1}^{-1}\mathbf{b}_{t+1}^p$
10: **end for**

4.1.1 Privacy Analysis of DP-CoLin. Lemma 1 provides the sensitivity of model statistics $\mathbf{b}_t$ in CoLin, based on which we develop the privacy guarantee of DP-CoLin.

Lemma 1 (Sensitivity of $\mathbf{b}_t$ in CoLin). *Sensitivity of $\mathbf{b}_t$ is $\Delta = \max_i L\|\mathbf{W}_i\|_2$, where $\mathbf{W}_i$ is the i-th row of user dependecy matrix $\mathbf{W}$ and L is the norm of context vector $\mathbf{x}$.*

The proof of this lemma is provided in Appendix. Note that the sensitivity Δ of CoLin is related to the structure of $\mathbf{W}$; and we discuss two extreme cases of $\mathbf{W}$ to illustrate its effect on privacy protection. Consider when $\mathbf{W}$ is an identity matrix, the resulting sensitivity by our Lemma 1 is L, which is the same as in linear bandits, since there is no influence among users. When $\mathbf{W}$ is a uniform matrix, i.e., users have homogeneous influence among each other and $w_{ij} = \frac{1}{N}$, Lemma 1 shows the sensitivity is $\frac{L}{\sqrt{N}}$. This result is significant: stronger user dependency in CoLin not only leads to lower regret [37], but also smaller sensitivity of $\mathbf{b}_t$, which directly reduces the level of required noise to guarantee privacy. This result is also intuitive: when every user has uniform influence on each other, it becomes harder to tell whose action causes the observed change in the algorithm's output. Less perturbation is thus needed to protect a single user's privacy. This improvement can hardly be obtained by directly applying existing conclusions on linear bandits.

Based on the above sensitivity analysis, we prove privacy guarantee of DP-CoLin in the following.

Theorem 1 (Privacy of DP-CoLin). *Algorithm 1 with global sensitivity Δ defined in Lemma 1 is ϵ-differentially private.*

Proof. By applying tree-based mechanism [7, 12] with privacy budget ϵ and sensitivity Δ as shown in line 9-11 of Algorithm 1, the perturbed statistics $\mathbf{b}_t^p$ is ϵ-differentially private. Since differential privacy is immune to post-processing [13], this consequently makes the model parameter $\hat{\boldsymbol{\vartheta}}_t^p$ and the sequence of recommendation $\{a_t : t \in [1..T]\}$ produced by $\hat{\boldsymbol{\vartheta}}_t^p$ also ϵ-differentially private. □

4.1.2 Regret Analysis of DP-CoLin. We first prove the corresponding confidence bound of parameter estimation in DP-CoLin, i.e., α_t in line 5 of Algorithm 1, which governs its upper confidence bound

based arm selection for online learning. In the following discussion, we use $\|\mathbf{B}\|_{\mathbf{A}} = \sqrt{\mathbf{B}^\top \mathbf{A}\mathbf{B}}$ to denote the matrix norm of vector $\mathbf{B}$.

Lemma 2 (Confidence Bound of DP-CoLin). *For any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters in DP-CoLin is bounded by,*

$$\|\hat{\boldsymbol{\vartheta}}_t^p - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\Big) - 2\log(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \frac{\Delta}{\epsilon}\log T\sqrt{\log t}\log\frac{1}{\delta}$$

The proof is provided in Appendix. The right-hand side of the inequality in Lemma 2 gives us α_t that is used in line 5 of Algorithm 1 for arm selection. We notice that in order to maintain a private bandit model $\hat{\boldsymbol{\vartheta}}_t^p$, the parameter estimation error of DP-CoLin suffers from an additional term $\frac{\Delta}{\epsilon}\log T\sqrt{\log t}\log\frac{1}{\delta}$ comparing to that in CoLin due to the added noise η_t. Based on Lemma 2, we have the following theorem about the upper regret bound of the DP-CoLin algorithm, which shows the trade-off between privacy budget ϵ and regret.

Theorem 2 (Regret of DP-CoLin). *With probability at least $1-\delta$, the cumulative regret of DP-CoLin algorithm satisfies,*

$$\boldsymbol{R}(T) \leq 2\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}\Bigg(\sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\Big) - 2\log(\delta)} + \frac{\max_i L\|\mathbf{W}_i\|_2}{\epsilon}\log^{1.5} T\log\frac{1}{\delta}\Bigg) \tag{3}$$

Specifically, the added regret of DP-CoLin comparing to the CoLin is the last term, i.e.,

$$\frac{2\max_i L\|\mathbf{W}_i\|_2}{\epsilon}\log^{1.5} T\log\frac{1}{\delta}\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}$$

We illustrate the proof details in Appendix. From Theorem 2, we can find that the dependency structure plays an important role in the added regret, and again we discuss those two extreme cases of $\mathbf{W}$ to explain its effect. If $\mathbf{W}$ is an identity matrix, DP-CoLin algorithm is *equivalent to* running N independent private LinUCB [29] for each user and the added regret is in the order of $O\big(\frac{\sqrt{N}}{\epsilon}\log^{1.5} T\sqrt{\log\frac{T}{N}}\sqrt{T}\log\frac{1}{\delta}\big)$. If $\mathbf{W}$ is a uniform matrix, the added regret is in the order of $O\big(\frac{1}{\epsilon}\log^{1.5} T\sqrt{\log\frac{T}{N}}\sqrt{T}\log\frac{1}{\delta}\big)$. It is important to note that the collaboration structure also helps reduce the added regret *by a factor of* $\frac{1}{\sqrt{N}}$.

4.2 Local Differential Privacy for CoLin

Global differential privacy for CoLin requires each user to send true reward (e.g., clicks) to the server, which then aggregates the data, injects noise, and publishes a privacy preserving output. Local differential privacy lifts the trust on the server by asking each user to perturb his/her data locally, before any disclosure to non-trustful server or the communication. Intuitively, this stronger privacy guarantee is at the cost of worse utility.

We present the Locally Differentially Private CoLin algorithm (LDP-CoLin) in Algorithm 2 in Appendix. LDP-CoLin requires a different communication mechanism: instead of directly sending reward r_{a_t,u_t} to the server, each user u maintains $\mathbf{b}_{u,t} = \sum_{t'=1}^{t_u-1}\tilde{\mathbf{x}}_{a_{t'},u} r_{a_{t'},u}$ locally as shown in line 8 of Algorithm 2. Each user perturbs their own $\mathbf{b}_{u,t}$ by a tree-based mechanism, where noise scales with per-user sensitivity Δ_u (line 8-9), and then sends it to the server. The server aggregates the received statistics to get $\mathbf{b}_t^p$ as shown in line 12, and uses it for model estimation and subsequent recommendations. Again in LDP-CoLin the key is to analyze the sensitivity, which controls the minimum amount of noise needed for privacy protection.

4.2.1 Privacy Analysis of LDP-CoLin. We first analyze the sensitivity Δ_u of $\mathbf{b}_{u,t}$ for each user u, and then show that Algorithm 2 is locally differentially private using this per-user sensitivity.

Lemma 3 (Sensitivity of $\mathbf{b}_{u,t}$ in CoLin). *Sensitivity of $\mathbf{b}_{u,t}$ for user u is $\Delta_u = L\|\mathbf{W}_u\|_2$.*

The proof is similar to Lemma 1 and the details are provided in Appendix. The main difference is that sensitivity Δ_u is for a specific user u, which only relies on his/her dependent neighbors, i.e., $\mathbf{W}_u$.

Theorem 3 (Privacy of DP-CoLin). *Randomized response $\mathbf{b}_{u,t}^p$ in Algorithm 2 with sensitivity Δ_u defined in Lemma 3 is ϵ-locally differentially private.*

The proof is similar to DP-CoLin but works in the local setting: as shown in line 8-9 of Algorithm 2, each user u maintains his/her own tree-based mechanism with privacy level ϵ and sensitivity Δ_u *locally*. The local statistics $\mathbf{b}_{u,t}$ are perturbed by the tree-based mechanism thus is ϵ-locally differentially private, and thus are $\hat{\boldsymbol{\vartheta}}_t^p$ and the resulting recommendation sequence.

4.2.2 Regret Analysis of LDP-CoLin. Due to local noise injection, the server's arm selection strategy has to be revised accordingly, which can be guided by the following lemma.

Lemma 4 (Confidence Bound of LDP-CoLin). *Let t_i be the number of times where user i interacts with the system up to time t, i.e., $\sum_i t_i = t$. For any $\delta > 0$, with probability at least $1-\delta$, the estimation error of bandit parameters in LDP-CoLin is bounded by,*

$$\|\hat{\boldsymbol{\vartheta}}_t^p - \boldsymbol{\vartheta}^*\|_{\mathbf{A}_t} \leq \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'}j}^2}{\lambda dN}\Big) - 2\log(\delta)} + \sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \frac{1}{\epsilon}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\log t_i(\Delta_i\log T_i)^2}$$

The proof detail is shown in Appendix. Similarly, the right-hand side of the inequality gives us α_t which is used in line 5 of Algorithm 2. Based on it, we have the following theorem about the upper regret bound of LDP-CoLin.

Theorem 4 (Regret of LDP-CoLin). *With probability at least $1-\delta$, the cumulative regret of LDP-CoLin algorithm (Algorithm 2)*

satisfies,

$$\boldsymbol{R}(T) \leq 2\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}\Bigg(\sqrt{\lambda}\|\boldsymbol{\vartheta}^*\| + \sqrt{dN\log\Big(1+\frac{\sum_{t'=1}^{t}\sum_{j=1}^{N} w_{u_{t'} j}^2}{\lambda dN}\Big)-2\log(\delta)} + \frac{1}{\epsilon}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\|\mathbf{W}_i\|^2\log^3 T_i}\Bigg) \quad (4)$$

Specifically, the added regret of LDP-CoLin comparing to the non-private CoLin is the last term.

Due to space limit, we omit the details of this proof. Note that Theorem 4 is in a general form in which we do not make any assumption about the users' arriving frequency or order. To better illustrate the added regret, we discuss a special case where the frequency of each user interacting with the system is the same, i.e., $T_i = \frac{T}{N}$. The added regret can thus be simplified as,

$$\frac{2}{\epsilon}\log^{1.5}\frac{T}{N}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\|\mathbf{W}_i\|^2}\sqrt{2dNT\log\Big(1+\frac{\sum_{t=1}^{T}\sum_{j=1}^{N} w_{u_t j}^2}{\lambda dN}\Big)}.$$

Consider the best case scenario where $\mathbf{W}$ is a uniform matrix, e.g., maximum collaboration, the added regret in LDP-CoLin is in the order of $O\Big(\frac{\sqrt{N}}{\epsilon}\log^2\frac{T}{N^2}\sqrt{T}\log\frac{1}{\delta}\Big)$, while DP-CoLin only has the added regret of $O\Big(\frac{1}{\epsilon}\log^{1.5}T\sqrt{\log\frac{T}{N^2}}\sqrt{T}\log\frac{1}{\delta}\Big)$. In fact, in both cases of the illustrative dependency structure, e.g., no collaboration and uniform collaboration, the added regret of LDP-CoLin is roughly $\sqrt{N}$-times larger compared with DP-CoLin's, and increases when the number of users grows. This is the inevitable cost to protect privacy in the local (user) level. We verified this relationship between the number of users and regret in our empirical evaluations later as well.

5 GENERAL FRAMEWORK AND APPLICATION TO GOBLIN

5.1 A General Framework for Differentially Private Collaborative Bandits

Although having different ways of realizing user dependency, the estimation of user preference $\boldsymbol{\vartheta}$ in a collaborative bandit algorithm can be unified by following: $\hat{\boldsymbol{\vartheta}}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$, with $\mathbf{A}_t = \lambda\mathbf{I} + \sum_{t'=1}^{t-1}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}^{\mathsf{T}}$ and $\mathbf{b}_t = \sum_{t'=1}^{t-1}\tilde{\mathbf{x}}_{a_{t'},u_{t'}}r_{a_{t'},u_{t'}}$. While the matrix $\mathbf{A}_t$ and vector $\mathbf{b}_t$ take the same form as that in CoLin, the projected feature vector $\tilde{\mathbf{x}}_{a_t,u_t} \in \mathbb{R}^{dN}$ takes different forms in different collaborative bandit algorithms, because of their unique ways of user dependency modeling. As discussed in previous section, CoLin [37] enables additive weighted reward sharing through projected feature vector $\tilde{\mathbf{x}}_{a_t,u_t} = \mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t}\mathbf{W}^{\mathsf{T}})$. As another example, GOBLin [6] encodes collaboration through graph regularization in the projected feature vector $\tilde{\mathbf{x}}_{a_t,u_t} = \mathbf{G}_{\otimes}^{-1/2}\mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t})$, where $\mathbf{G} = \mathbf{I}_d + \mathbf{L}$, $\mathbf{L}$ is the graph Laplacian of the network and $G_{\otimes} = G \otimes \mathbf{I}$ is the Kronecker product between two matrices G and $\mathbf{I}_d$.

We now describe our general framework for equipping collaborative bandit algorithms with differential privacy. The *first key step* is to add noise η_t to the sufficient statistics $\mathbf{b}_t$ at each time t. Since $\mathbf{b}_t$ can be treated as a sum statistic, we sample the noise η_t from a tree-based mechanism [7, 12] based on Laplace noise to avoid adding unnecessary noise. However, it is non-trivial to decide the scale (variance) of η_t, where the scale is proportional to the sensitivity of the private statistics $\mathbf{b}_t$. In a collaborative learning framework, the challenge comes from the information sharing in collaborative learning, i.e., the change in one user's reward feedback promptly leads to changes in all dependent users' observed recommendation sequences. The trivial sensitivity analysis over $\mathbf{b}_t$ in private linear bandits results in a bound of NL and is unnecessarily large. To reduce the amount of noise, our key insight is to conduct tight sensitivity analysis over $\mathbf{b}_t$ with respect to the *structure of collaboration*. Similar to Lemma 1 of DP-CoLin, the tight sensitivity analysis in a private collaborative bandit algorithm makes it possible to inject *less noise* to guarantee same privacy level comparing to private LinUCB. The significance of this framework lies in that collaboration helps to achieve *stronger privacy* with the same amount of injected noise. The *second key step* is to derive the corresponding confidence bound for exploration considering the existence of the Laplace noise in the model (similar to Lemma 2). After the two steps, we obtain a private collaborative bandit algorithm and can analyze its privacy-utility trade-off (similar to Theorem 2).

5.2 Global Differential Privacy for GOBLin

We now show that our solution framework can be easily extended to another state-of-the-art collaborative bandit solution GOBLin [6]. Due to space limit and similarity between GOBLin and CoLin, we do not list the complete algorithm of DP-GOBLin.

In a nutshell, GOBLin models dependency among users by requiring connected users in a network to have similar bandit parameters via a graph Laplacian based model regularization. Specifically, the projected feature vectors $\tilde{\mathbf{x}}_{a_t,u_t} = \mathbf{G}_{\otimes}^{-1/2}\mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,u_t})$ and $\mathbf{G} = \mathbf{I}_d + \mathbf{L}$. $\mathbf{L}$ is the graph Laplacian and $\mathbf{I}_d$ is a $d \times d$ identity matrix. The parameter $\hat{\boldsymbol{\vartheta}}$, matrix $\mathbf{A}_t$ and vector $\mathbf{b}_t$ take the same form as that in CoLin. To realize the first step of our framework, we add noise η_t to $\mathbf{b}_t = \sum_{t'=1}^{t-1}\tilde{\mathbf{x}}_{a_{t'}}r_{t'}$ to achieve global differential privacy. The noise η_t is sampled from a tree-based mechanism and the scale depends on the privacy budget ϵ and sensitivity Δ of GOBLin. And to realize the second step of our framework, we can derive the confidence bound α_t of DP-CoLin as

$$\alpha_t = \sqrt{\log\frac{|\mathbf{A}_T|}{\delta}} + L(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_N) + \frac{\max_i L\|\mathbf{G}_i^{-1/2}\|_2}{\epsilon}\log^{1.5}T\log\frac{1}{\delta} \quad (5)$$

where $L(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_N) = \sum_{i=1}^{N}\|\boldsymbol{\theta}_i\|_2 + \sum_{(i,j)\in E}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2$ and $|\mathbf{A}_T|$ is the determinate of matrix $\mathbf{A}_T$.

We now show that our analysis for DP-CoLin can be seamlessly generalized to DP-GOBLin.

Lemma 5 (Sensitivity of $\mathbf{b}_t$ in GOBLin). *Sensitivity of $\mathbf{b}_t$ is $\Delta = \max_i L\|\mathbf{G}_i^{-1/2}\|_2$, where $\mathbf{G}_i^{-1/2}$ is the i-th row of the square root of user dependency $\mathbf{G}$'s graph Laplacian and L is the bound of context vector $\mathbf{x}$'s norm.*

Proof details of this lemma are provided in the appendix. Before we study the relationship between sensitivity Δ and the structure of $\mathbf{G}$, we first rewrite the sensitivity as $\Delta = \max_i L\|\mathbf{G}_i^{-1/2}\|_2 =$

$\max_i L\sqrt{\mathbf{G}_{ii}^{-1}}$, which is easier to perceive. We again use two extreme cases to explain the derived sensitivity. When $\mathbf{G}$ is an identity matrix, i.e., users are independent and disconnected, the graph Laplacian is a zero matrix, $\mathbf{G}_{ii}^{-1} = 1$, and the sensitivity Δ is L. When the graph is fully connected, we have $\mathbf{G}_{ii} = N$ and $\mathbf{G}_{ij} = -1$ for $\forall i, j \in [1..N], i \neq j$. Then its inverse is

$$\mathbf{G}^{-1} = \frac{1}{N+1}\begin{bmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 2 \end{bmatrix},$$

and the sensitivity Δ is $\frac{2L}{\sqrt{N+1}}$. Here again we observe that stronger user connectivity leads to smaller sensitivity of $\mathbf{b}_t$. Based on this analysis, we prove the following privacy guarantee of DP-GOBLin.

Theorem 5 (Privacy). *DP-GOBLin with global sensitivity Δ defined in Lemma 5 is ϵ-differentially private.*

The proof is similar as the privacy theorem in DP-CoLin using post-processing invariant property of differential privacy, and we omit the proof details. Correspondingly, we have the following theorem about the upper regret bound of the DP-GOBLin algorithm.

Theorem 6 (Regret of DP-GOBLin). *With probability at least $1-\delta$, the cumulative regret of DP-GOBLin algorithm satisfies,*

$$\begin{aligned}\boldsymbol{R}(T) \leq & 2\sqrt{T(1+L^2)\log|\mathbf{A}_T|}\Bigg(\sqrt{\log\frac{|\mathbf{A}_T|}{\delta}} + L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \\ & + \frac{\max_i L\|\mathbf{G}_i^{-1/2}\|_2}{\epsilon}\log^{1.5} T\log\frac{1}{\delta}\Bigg) \end{aligned} \tag{6}$$

Specifically, the added regret of DP-GOBLin comparing to the non-private GOBLin is the last term in Eq (6). Similarly as in DP-CoLin, the structure of collaboration, specified by $\mathbf{G}$, greatly affects the regret bound in terms of $\|\mathbf{G}_i^{-1/2}\|_2$ and $\mathbf{A}_T$. For example, larger regret reduction is expected when users are more closely related. Due to space limit, we omit the detailed results here.

5.3 Local Differential Privacy for GOBLin

Similar to LDP-CoLin, LDP-GOBLin works in the local setting where each user u maintains $\mathbf{b}_{u,t} = \sum_{t'=1}^{t_u-1} \tilde{\mathbf{x}}_{a_{t'},u} r_{a_{t'},u}$ locally and perturbs it by a tree-based mechanism with noise scales by per-user sensitivity Δ_u of GOBLin. Below we first show its local privacy guarantee, and then analyze the trade-off between privacy and regret.

Lemma 6 (Sensitivity of $\mathbf{b}_{u,t}$ in GOBLin). *Sensitivity of $\mathbf{b}_{u,t}$ for user u is $\Delta_u = L\|\mathbf{G}_u^{-1/2}\|_2$.*

Theorem 7 (Privacy of LDP-GOBLin.). *Randomized response $\mathbf{b}_{u,t}^p$ in LDP-GOBLin with sensitivity Δ_u defined in Lemma 6 is ϵ-locally differentially private.*

Theorem 8 (Regret of LDP-GOBLin). *With probability at least $1-\delta$, the cumulative regret of LDP-GOBLin algorithm satisfies,*

$$\begin{aligned}\boldsymbol{R}(T) \leq & 2\sqrt{T(1+L^2)\log|\mathbf{A}_T|}\Bigg(\sqrt{\log\frac{|\mathbf{A}_T|}{\delta}} + L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \\ & + \frac{1}{\epsilon}\log\frac{1}{\delta}\sqrt{\sum_{i=1}^{N}\|\mathbf{G}_i^{-1/2}\|^2\log^3 T_i}\Bigg) \end{aligned} \tag{7}$$

where $L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) = \sum_{i=1}^N \|\boldsymbol{\theta}_i\|_2 + \sum_{(i,j)\in E}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2$.

Specifically, the added regret of LDP-GOBLin comparing to the non-private GOBLin is the last term. Similar to the discussion of LDP-CoLin, we can also compare DP-GOBLin and LDP-GOBLin in the scenario where each user interacts with the system at the same frequency. And the conclusion is also similar: in those two extreme cases, i.e., totally isolated or related users, the added regret of LDP-GOBLin is approximately $\sqrt{N}$*-times larger* than DP-GOBLin. This again illustrates the required cost for stronger privacy guarantee.

6 EXPERIMENT

We performed empirical evaluations of our developed private collaborative bandit algorithms against several baseline algorithms including the non-private collaborative bandit algorithms CoLin [37] and GOBLin [6], non-private LinUCB [23] and private LinUCB [29]. The datasets include a synthetic dataset from simulation, and two real-world datasets for music recommendation and bookmark recommendation.

6.1 Evaluation Datasets

• **Synthetic dataset.** To build a synthetic dataset, we follow the settings in [6, 37] to simulate a collaborative online recommendation environment. Specifically, we generate N users, each of which is associated with a d-dimensional parameter vector $\boldsymbol{\theta}^*$, i.e., $\boldsymbol{\Theta}^* = (\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_N^*)$. Each dimension of $\boldsymbol{\theta}_i^*$ is drawn from a uniform distribution $U(0,1)$ and normalized to $\|\boldsymbol{\theta}_i^*\|_2 = 1$. $\boldsymbol{\Theta}^*$ is treated as the ground-truth bandit parameters for reward generation, and they are withheld from bandit algorithms. We construct the golden relational stochastic matrix $\mathbf{W}$ for the graph of users by defining $w_{ij} \propto \langle \boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^* \rangle$. We delete the edges where w_{ij} is smaller than a predefined threshold, and get the final user graph G by normalizing each column of $\mathbf{W}$ by its L1 norm. Note that since w_{ij} is generated proportionally to the similarity between $\boldsymbol{\theta}_i^*$ and $\boldsymbol{\theta}_j^*$, the resulting graph naturally satisfies the collaborative assumption in GOBLin [6], i.e., connected users share similar $\boldsymbol{\theta}^*$. The resulting user graph G represented by the relational matrix $\mathbf{W}$ are disclosed to the bandit algorithms. In the end, we generate a size-K arm pool $\mathcal{A}$. Each arm a in $\mathcal{A}$ is associated with a d-dimensional feature vector $\mathbf{x}_a$, each dimension of which is also drawn from $U(0,1)$. We normalize $\mathbf{x}_a$ by its L2 norm.

To simulate the collaborative reward generation process among users, we compute the reward of arm a for user i at time t as $r_{a_{t,i}} = \mathrm{Vec}(\mathring{\mathbf{X}}_{a_t,i}\mathbf{W}^\mathsf{T})^\mathsf{T}\mathrm{Vec}(\boldsymbol{\Theta}^*) + \gamma_t$ following Eq (2), where $\gamma_t \sim N(0, \sigma^2)$. To increase the learning complexity, at each time t, our simulator only discloses a subset of arms in $\mathcal{A}$ to the learning algorithms, e.g., randomly select 10 arms from $\mathcal{A}$ without replacement. In simulation, based on the known bandit parameters $\boldsymbol{\Theta}^*$, the optimal arm $a_{t,i}^*$ and the corresponding reward $r_{a_{t,i}^*}$ for each user i at time t can be explicitly computed. In our experiment, we set the number of users $N = 10$ and size of arm pool $K = 1,000$. We run $T = 30,000$ iterations and interact with users evenly, which means we serve each user i in total $T_i = 3,000$ iterations.

• **LastFM and Delicious datasets** The LastFM dataset is extracted from the music streaming service Last.fm, and the Delicious dataset is extracted from the social bookmark sharing service Delicious.

The two datasets are created by the HetRec 2011 workshop with the goal of investigating the usage of heterogeneous information in recommender systems [1]. The LastFM dataset contains 1,892 users and 17,632 items (artists). The Delicious dataset contains 1,861 users and 69,226 items (URLs). To make these two datasets suitable for evaluating collaborative contextual bandit algorithms, necessary pre-processing is needed. We followed the same pre-processing steps and experiment settings in [6, 37]. To make this paper self-explanatory, we provide a brief description about the pre-processing steps on the two datasets. More details of the pre-processing can be found in [6, 37].

Reward: On LastFM dataset, information about "listened artists" of each user is used to create reward for the bandit algorithms: if a user listened to an artist at least once, the reward is 1; otherwise 0. On Delicious dataset, the reward for bookmarked URLs is set to 1; otherwise 0.

Context features and candidate arms: On both datasets, all tags associated with a particular item are used to create a TF-IDF feature vector, which uniquely represents the content of that item. PCA is used to reduce the dimensionality of the feature vectors to $d = 25$. For a particular user i, we generate the candidate arm pool with size $K = 25$ by first selecting one item from those non-zero reward items in user i based on the observations in the dataset, and then randomly selecting the other 24 from those zero-reward items for user i.

User relation information: Both datasets contain users' social network graph, which makes them a suitable testbed for collaborative bandit algorithms. User relation graph is directly extracted from the available social network in the datasets. In order to make the graph denser and the algorithms computationally feasible, we used graph-cut [9] to cluster users into 100 clusters. Users in the same cluster are assumed to have the same bandit model. After user clustering, a weighted graph can be generated: the nodes are the clusters from the original graph; and the edges between different clusters are weighted by the number of inter-cluster edges in the original graph. Then the relational matrix $\mathbf{W}$ in CoLin is obtained by setting $w_{ij} \propto c(i, j)$, where $c(i, j)$ is the number of edges between cluster i and j.

6.2 Experiment Results

• **Regret comparison.** On the synthetic dataset, cumulative regret is used to evaluate the performance of the compared algorithms. In the real-world datasets, since we do not have an oracle policy, we instead use each learning algorithm's cumulative reward for evaluation. The cumulative regret (the lower the better) on the synthetic dataset and cumulative reward (the higher the better) on real-world datasets are reported in Figure 1 (a) and Figure 2 respectively. We set the privacy budget $\epsilon = 2$ for all private algorithms in our experiments by default.

In both synthetic and real-world datasets, the non-private collaborative bandits performed better than their globally and locally private counterparts, which is surely expected. We also observe that compared with the globally differentially private collaborative bandit algorithms, i.e., DP-CoLin and DP-GOBLin, the locally differentially private algorithms have significantly worse regret (smaller cumulative reward). This is also expected as local differential privacy is a stronger privacy definition on the user side, and more model perturbation has to be introduced to achieve so. Specifically, as our analysis in Section 4.2 and Section 5.3 suggested, the added regret of LDP collaborative bandit algorithms are roughly $\sqrt{N}$-times larger than their DP counterparts.

We also notice that DP-CoLin and DP-GOBLin performed better than DP-LinUCB in both synthetic and real-world datasets. The improvement comes from two sources: 1) collaborative learning, which improves the convergence rates of model parameter estimation as discussed in [6, 37]; and 2) privacy mechanism under the collaborative environment, which adds less noise than DP-LinUCB when users are not all independent or disconnected. Accordingly to Figure 1 (a), it is obvious that comparing to the regret difference between LinUCB and GOBLin or CoLin, the regret difference between DP-LinUCB and DP-GOBLin or DP-CoLin is much larger. This confirms that the main reason of regret reduction is the calibrated privacy mechanisms developed in this paper.

• **Parameter estimation quality.** To better illustrate the performance of different bandit algorithms, we also studied their parameter estimation quality, which directly measures the algorithms' online learning convergence. Specifically, we reported the L2 difference between the estimated bandit parameter $\hat{\boldsymbol{\vartheta}}_t$ and the ground-truth parameter $\boldsymbol{\vartheta}^*$ in Figure 1 (b). We observe that private collaborative bandit algorithms have a slower model convergence than their non-private counterparts. Moreover, local differential privacy clearly imposes a much larger estimation error comparing to their counterparts with global differential privacy (note that the y-axis is on a log-scale), which further confirms the required cost to guarantee privacy in the local setting.

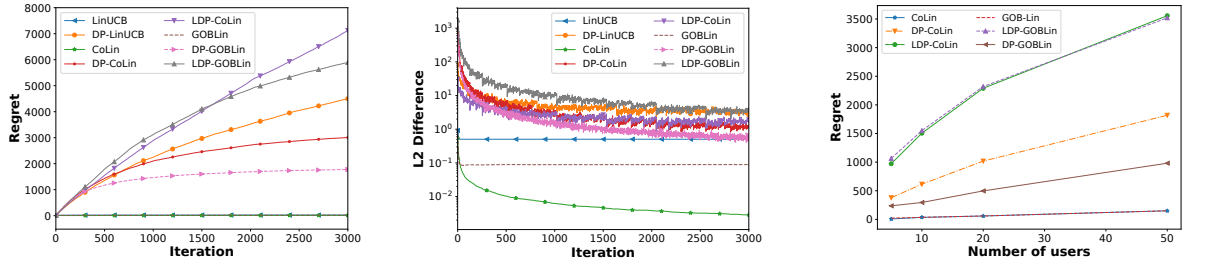
6.3 Detailed Algorithm-level Analysis

To better understand the trade-off between privacy and utility in collaborative bandit learning, we varied the privacy parameter ϵ and number of users in our evaluation.

• **Effect of privacy budget ϵ.** In Table 1, we reported the cumulative regret of the collaborative bandit algorithms with global and local differential privacy under different privacy parameter ϵ. We vary ϵ from 0.5 to 10. We run each experiment for $T = 10,000$ iterations and report the average regret of 5 repeated runs. From the results, we notice a clear trade-off between the required privacy level ϵ and the resulting regret. Stronger privacy requirement (i.e., a smaller ϵ) requires the privacy mechanism to introduce more noise, which directly inflates regret. This result also supports our theoretical analysis that the added regret of the private collaborative bandit algorithms is in the order of $O(\frac{1}{\epsilon})$.

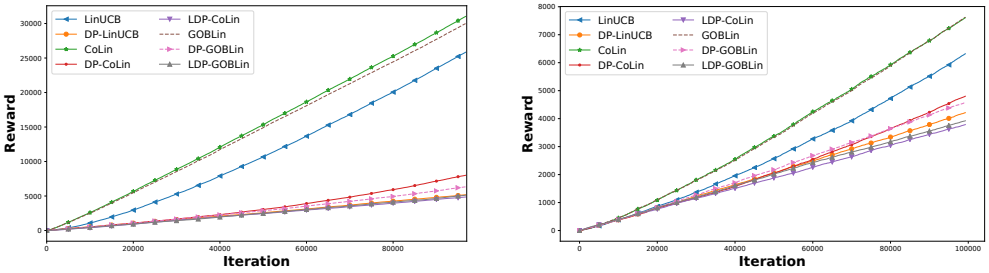
• **Effect of number of users N.** In Figure 1 (c), we show the cumulative regret of the collaborative bandit algorithms with global and local differential privacy under different number of users N. We run $T = 10,000$ iterations and all users are evenly served for $\frac{T}{N}$ times. We vary N from 5 to 50. From the result we observe that the regret increases with the number of users. By looking at the difference between the regret of non-private algorithms and their private versions, we can notice that the added regret increases with number of users N. This also validates our theoretical analysis that

[1] Datasets and their full description is available at http://grouplens.org/datasets/hetrec-2011



(a) Cumulative regret (b) Parameter estimation error (c) Regret under different number of users

Figure 1: Experimental results on synthetic dataset.



(a) Cumulative reward on LastFM dataset. (b) Cumulative reward on Delicious dataset.

Figure 2: Experimental results on real-world datasets.

Table 1: Cumulative regret across different bandit algorithm under different privacy level ϵ.

ϵ	0.5	1	2	5	10
DP-LinUCB	3082.90±82.69	2683.69±89.74	1504.14±30.40	910.97±20.83	496.11±14.72
DP-CoLin	2619.56±29.44	2450.70±50.51	1327.70±23.79	884.19±23.12	297.18±6.80
DP-GOBLin	2672.56±29.13	2550.22±19.67	964.63±13.61	685.65±6.47	246.92±9.70
LDP-CoLin	3310.53±51.85	3095.10±48.97	2389.05±61.24	1795.40±31.21	938.76±26.16
LDP-GOBLin	3268.70±65.61	3004.80±75.08	2334.62±63.78	1743.57±36.40	1060.53±28.99

the added regret for LDP collaborative bandit algorithms is roughly $\sqrt{N}$ times larger than their DP versions, which is the inevitable cost to protect privacy at the local level.

7 CONCLUSION

We studied the problem of protecting global and local differential privacy for collaborative bandits. Our solution framework allows the privacy mechanism to calibrate the noise scale with respect to the user dependency graph. Our theoretical analysis proves the desired privacy guarantee under both settings in two well-studied collaborative bandits. We also rigorously proved the corresponding upper regret bound of the derived private algorithms. Most importantly, we showed the added regret caused by differential privacy mechanism is still sublinear and benefits from the collaboration structure. Extensive experiments on both synthetic and real-world public datasets verified the effectiveness of the private collaborative bandit algorithms, especially the improved trade-off between utility and privacy requirement.

As the first private collaborative bandits research, we explored a specific type of collaborative bandits with explicit knowledge of user dependency structure. In future, we plan to study privacy for other types of collaborative bandit algorithms, such as online clustering-based bandits [17, 24] and matrix factorization based bandits [20, 35, 36]. We also note that the lower regret bound of a DP collaborative bandit algorithm is yet unknown, and it is important to investigate this lower bound to show the optimality of the upper bound of a private collaborative bandit algorithm.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This work was supported in part by Bloomberg Data Science Ph.D. Fellowship and National Science Foundation Award IIS-1553568 and IIS-1618948.

REFERENCES

[1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *NIPS*. 2312–2320.

[2] Jacob Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. 2017. Online learning via differential privacy. *arXiv preprint arXiv:1711.10019* (2017).

[3] Naman Agarwal and Karan Singh. 2017. The price of differential privacy for online learning. In *ICML 2017*. JMLR. org, 32–40.

[4] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Tradeoffs. *Journal of Machine Learning Research* 3 (2002), 397–422.

[5] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. 2011. " You might also like:" Privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy*. IEEE, 231–246.

[6] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. 2013. A gang of bandits. In *Advances in Neural Information Processing Systems*. 737–745.

[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] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, Mar (2011), 1069–1109.

[9] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. 2007. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence* 29, 11 (2007), 1944–1957.

[10] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 429–438.

[11] 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.

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

[13] 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.

[14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *SIGSAC 2014*. ACM, 1054–1067.

[15] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. 2010. Parametric bandits: The generalized linear case. In *NIPS*. 586–594.

[16] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *ICML 2017*. JMLR. org, 1253–1262.

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

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

[19] Prateek Jain, Om Dipakbhai Thakkar, and Abhradeep Thakurta. 2018. Differentially Private Matrix Completion Revisited. In *ICML*. 2215–2224.

[20] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *NIPS*. 1297–1305.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[22] Aleksandra Korolova. 2010. Privacy violations using microtargeted ads: A case study. In *ICDM 2010*. IEEE, 474–482.

[23] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of 19th WWW*. ACM, 661–670.

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

[25] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 1 (2003), 76–80.

[26] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. 2015. Fast differentially private matrix factorization. In *RecSys 2015*. ACM, 171–178.

[27] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD*. ACM, 627–636.

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

[29] Seth Neel and Aaron Roth. 2018. Mitigating bias in adaptive data gathering via differential privacy. *arXiv preprint arXiv:1806.02329* (2018).

[30] Roshan Shariff and Or Sheffet. 2018. Differentially Private Contextual Linear Bandits. In *Advances in Neural Information Processing Systems*. 4296–4306.

[31] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[32] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. 2017. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753* (2017).

[33] Abhradeep Guha Thakurta and Adam Smith. 2013. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*. 2733–2741.

[34] Aristide Charles Yedia Tossou and Christos Dimitrakakis. 2017. Achieving privacy in the adversarial multi-armed bandit. In *AAAI 2017*.

[35] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1633–1642.

[36] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization bandits for interactive recommendation. In *AAAI 2017*.

[37] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. 2016. Contextual bandits in a collaborative environment. In *SIGIR 2016*. ACM, 529–538.

[38] Tianqing Zhu, Gang Li, Yongli Ren, Wanlei Zhou, and Ping Xiong. 2013. Differential privacy for neighborhood-based collaborative filtering. In *ASONAM 2013*. ACM, 752–759.