Communication Efficient Federated Learning for Generalized Linear Bandits

Chuanhao Li and Hongning Wang
Department of Computer Science, University of Virginia

NEURAL INFORMATION PROCESSING SYSTEMS

Abstract

Contextual bandit algorithms have been recently studied under the federated learning setting to satisfy the demand of keeping data decentralized and pushing the learning of bandit models to the client side. But limited by the required communication efficiency, existing solutions are restricted to linear models to exploit their closed-form solutions for parameter estimation. Such a restricted model choice greatly hampers these algorithms' practical utility.

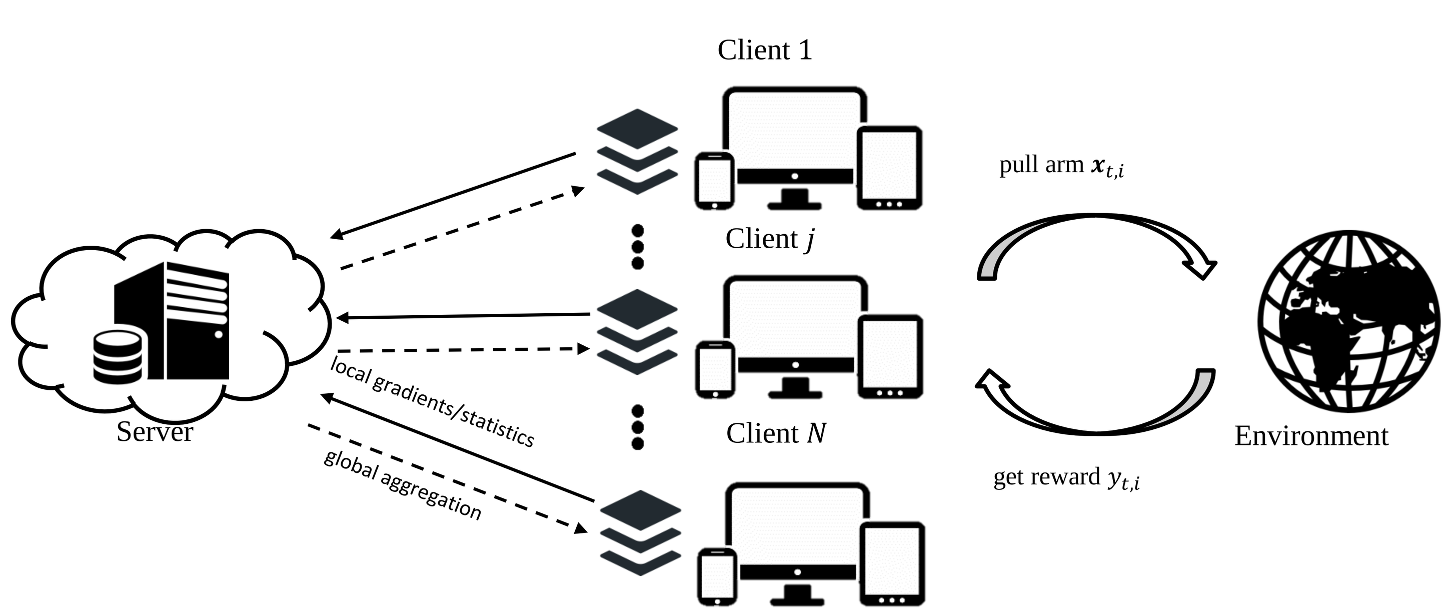
In this paper, we take the first step to addressing this challenge by studying generalized linear bandit models under the federated learning setting. We propose a communication-efficient solution framework that employs online regression for local update and offline regression for global update. We rigorously proved, though the setting is more general and challenging, our algorithm can attain sub-linear rate in both regret and communication cost, which is also validated by our extensive empirical evaluations.

Federated Bandit Learning

For time $t = 1,2,\dots,T$
For client $i = 1,2,\dots,N$
- Client i picks arm $x_{t,i}$ from set $\mathcal{A}_{t,i}$ and observes reward
$$y_{t,i} = f(x_{t,i}) + \eta_{t,i}$$
- Communication between the server and clients

Regret & Communication
- $R_T = \sum_{t=1}^{T}\sum_{i=1}^{N} r_{t,i}$, where $r_{t,i} = \max_{x\in\mathcal{A}_{t,i}} f(x) - f(x_{t,i})$
- C_T: total number of real numbers transferred in the system

Goal
- Incur sub-linear C_T w.r.t. T, while having near-optimal $R_T = O(d\sqrt{NT}\log NT)$



A network with N clients sequentially taking actions and receiving feedback from the environment, and a server that coordinates the communication among the clients.

Extension to Generalized Linear Bandits (GLB)

Prior works in linear bandits & challenges in extension to GLB

Federated linear bandits
- Joint model estimation $\hat{\boldsymbol{\theta}} = \mathbf{A}^{-1}\mathbf{b}$
- Communicate local updates of $\mathbf{A} = \lambda\mathbf{I} + \mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{d\times d}$, $\mathbf{b} = \mathbf{X}^\top\mathbf{y} \in \mathbb{R}^d$

	Regret R_T	Communication C_T
[Wang et al., ICLR' 20]	$O(d\sqrt{NT}\log NT)$	$\tilde{O}(d^3N^{1.5})$
[Li and Wang, AISTATS' 22, He et al., NeurIPS' 22]	$O(d\sqrt{NT}\log NT)$	$\tilde{O}(d^3N^2)$

Federated GLB:
- Assume reward function $f(x) = \mu(x^\top\theta_\star)$ where μ is the (inverse) link function, $\theta_\star \in \mathbb{R}^d$ is the unknown parameter
- More flexible modeling choices, e.g., linear, Poisson, logistic regression.
- No closed-form solution, need gradient-based optimization

Challenges
- Joint model estimation requires many rounds of communications to collect and aggregate local models/gradients
- Updating using local gradients pushes the model away from global model, i.e., forget knowledge gained in previous communications

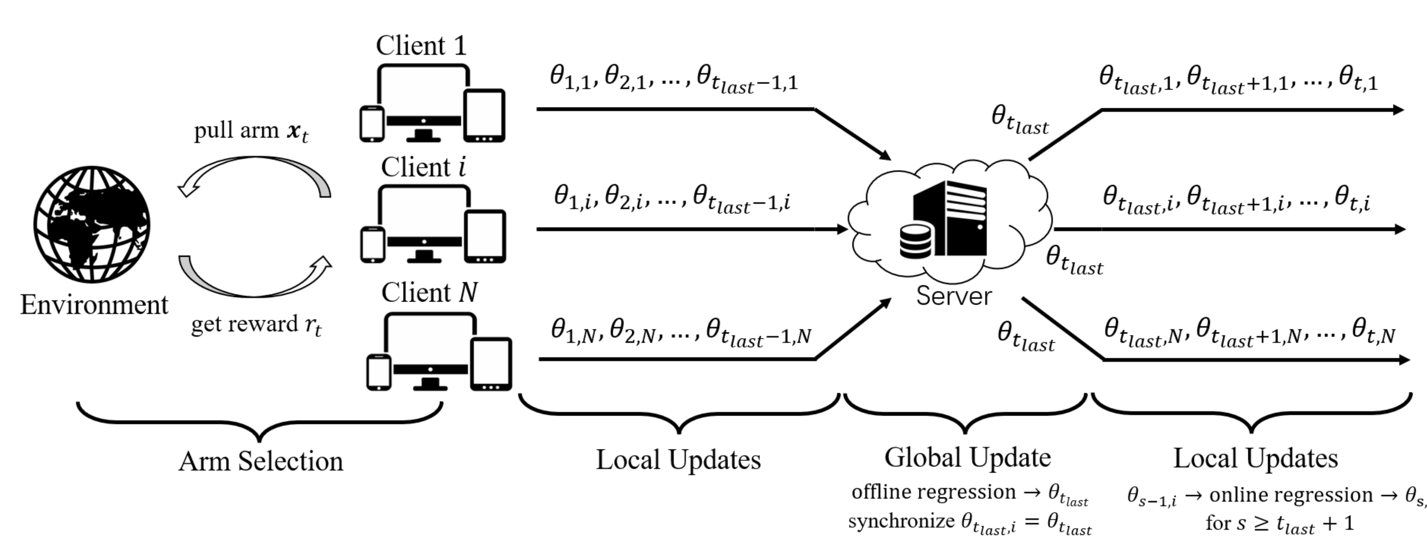
Main question: is sublinear communication cost still possible?

Proposed Solution

Combination of online and offline regression

if $(t - t_{\text{last}})\log\frac{\det(A_{t,i})}{\det(A_{t,i} - \Delta A_{t,i})} < D$ [Local update]
- use $(x_{t,i}, y_{t,i})$ to update client i's model via online Newton step

else [Global update]
- use $\{(x_{s,j}, y_{s,j})\}_{s\in[t],j\in[N]}$ to compute a global model via distributed accelerated gradient descent



Our proposed algorithm adopts a combination of online and offline regression subroutine, with online regression locally adjusting each client's model using its newly collected data, and offline (distributed) regression occasionally soliciting local gradients from all clients for joint model estimation when sufficient amount of new data has been accumulated

Construction of confidence ellipsoid for arm selection (UCB)
- The global and local model updates for each client i
$$\theta_{t_{last}}, \theta_{t_{last}+1,i}, \theta_{t_{last}+2,i}, \cdots, \theta_{t-1,i}$$
global update   local updates
- Define loss difference
$$\underbrace{\sum_{s=1}^{t_{\text{last}}}\sum_{i=1}^{N}\left[l(\mathbf{x}_{s,i}^\top\theta_{t_{\text{last}}}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\right]}_{\text{convergence of distributed AGD}} + \underbrace{\sum_{s=t_{\text{last}}+1}^{t}\left[l(\mathbf{x}_{s,i}^\top\theta_{s-1,i}, y_{s,i}) - l(\mathbf{x}_{s,i}^\top\theta_\star, y_{s,i})\right]}_{\text{online regret of ONS}}$$
- Confidence ellipsoid
$$\|\hat{\theta}_{t,i} - \theta_\star\|_{A_{t,i}}^2 \leq \beta_{t,i} + \frac{\lambda}{c_\mu}S^2 - \|\mathbf{z}_{t,i}\|_2^2 + \hat{\theta}_{t,i}^\top b_{t,i} := \alpha_{t,i}^2$$
where $\beta_{t,i} = O\left(\frac{d\log NT}{c_\mu^2}[k_\mu^2 + R_{\max}^2]\right)$

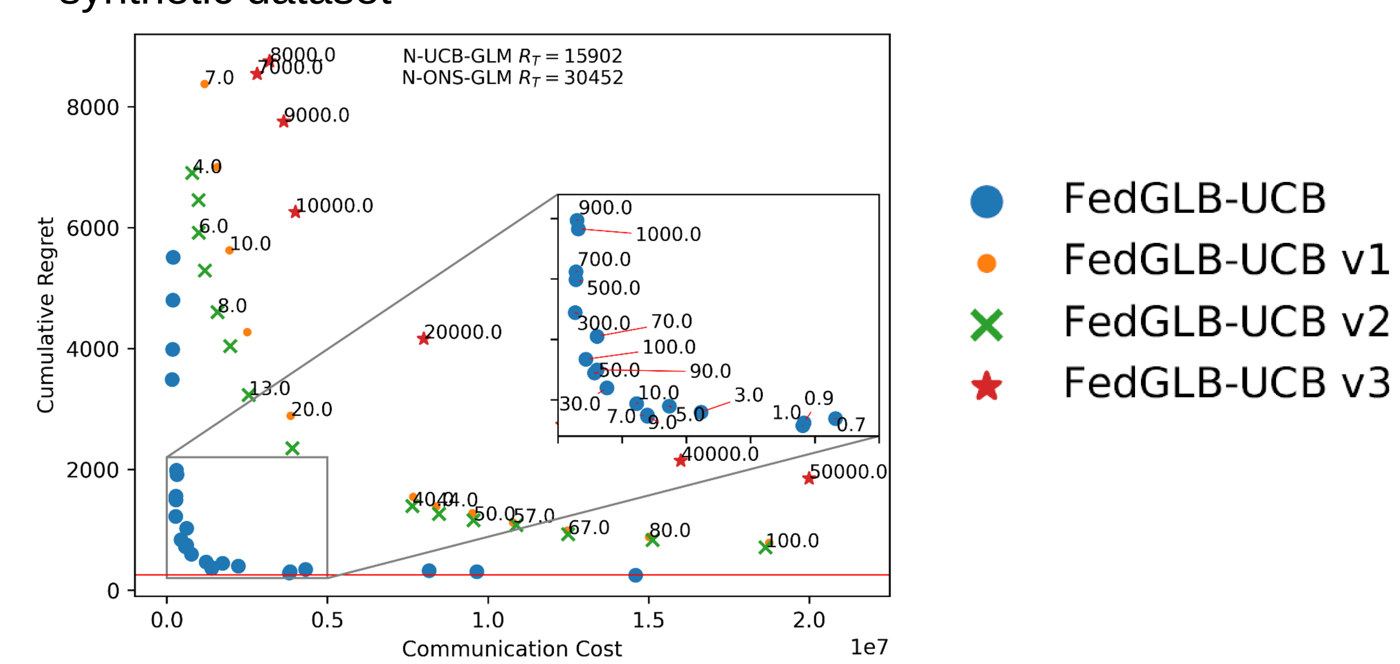
Theoretical Results

To attain near-optimal regret $R_T = O(k_\mu(k_\mu + R_{\max})/c_\mu d\sqrt{NT}\log NT)$, our proposed solution requires $C_T = O(d^3N^{1.5}\log NT + d^2N^2T^{0.5}\log^2 NT)$

During each of the $O(dN^{0.5}\log NT)$ global updates
- 1st term: synchronizing covariance matrix requires $O(N\cdot d^2)$
Match C_T of [Wang et al., ICLR' 20]
- 2nd term: iterative gradient optimization requires $O(N\cdot d\sqrt{NT}\log NT)$
Match the lower bound of distributed convex optimization [Arjevani and Shamir, NeurIPS' 15]

Experiment Results

synthetic dataset



References

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. (2019). "Distributed bandit learning: Near-optimal regret with efficient communication." In: International Conference on Learning Representations,

Chuanhao Li and Hongning Wang (2022). "Asynchronous upper confidence bound algorithms for federated linear bandits.". In: International Conference on Artificial Intelligence and Statistics.

Jiafan He, Tianhao Wang, Yifei Min, Quanquan Gu (2022). "A Simple and Provably Efficient Algorithm for Asynchronous Federated Contextual Linear Bandits". In: Advances in Neural Information Processing Systems 36

Yossi Arjevani and Ohad Shamir. (2015). "Communication complexity of distributed convex learning and optimization." In: Advances in Neural Information Processing Systems 28