Communication Efficient Distributed Learning for Kernelized Contextual Bandits

Chuanhao Li[1], Huazheng Wang[2], Mengdi Wang[3], and Hongning Wang[1]

1. University of Virginia, 2. Oregon State University, 3. Princeton University

NEURAL INFORMATION PROCESSING SYSTEMS

Abstract

We tackle the communication efficiency challenge of learning kernelized contextual bandits in a distributed setting. Despite the recent advances in communication-efficient distributed bandit learning, existing solutions are restricted to simple models like multi-armed bandits and linear bandits, which hamper their practical utility. In this paper, instead of assuming the existence of a linear reward mapping from the features to the expected rewards, we consider non-linear reward mappings, by letting agents collaboratively search in a reproducing kernel Hilbert space (RKHS). This introduces significant challenges in communication efficiency as distributed kernel learning requires the transfer of raw data, leading to a communication cost that grows linearly w.r.t. time horizon T. We address this issue by equipping all agents to communicate via a common Nystrom embedding that gets updated adaptively as more data points are collected. We rigorously proved that our algorithm can attain sub-linear rate in both regret and communication cost.

Distributed Bandit Learning

For each round l= $1,2,\dots,T$ $\boxed{\text{index } t := N(l-1)+i}$

For client $i=1,2,\dots,N$

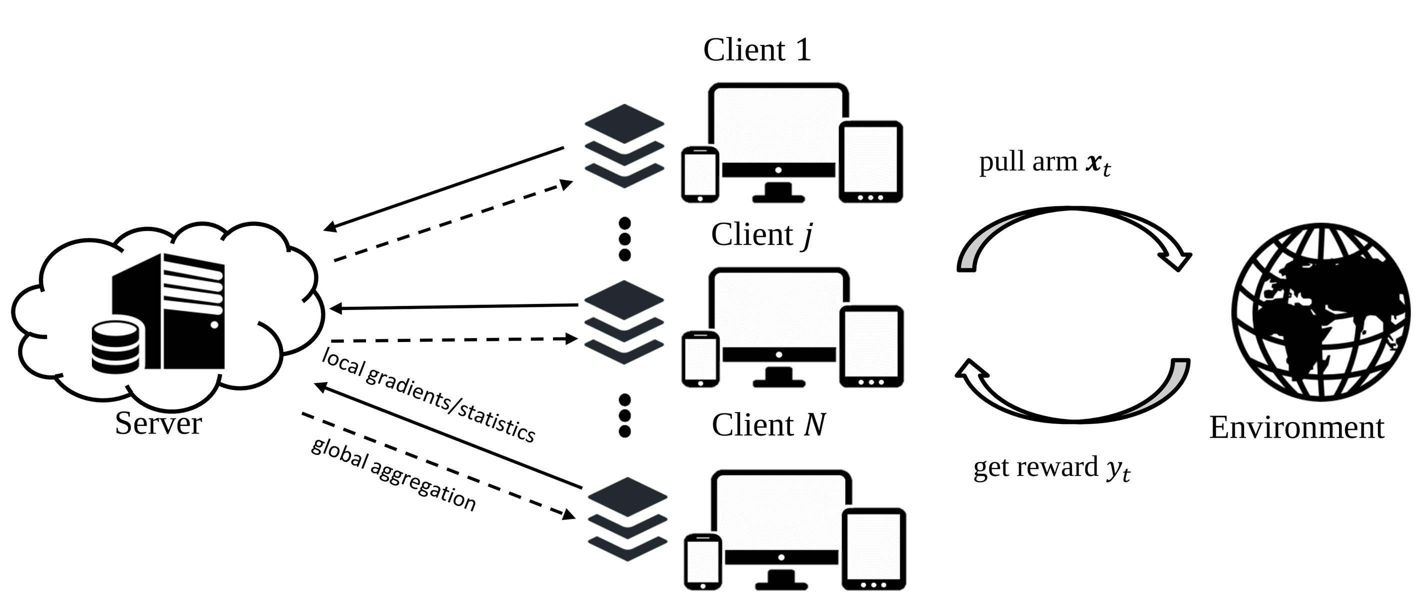
- Client i_t picks arm x_t from set $\mathcal{A}_t$ and observes reward $y_t = f(x_t) + \eta_t$
- Communication between the server and clients

Regret & Communication

- $R_T = \sum_{t=1}^{NT} r_t$, where $r_t = \max_{x \in \mathcal{A}_t} f(x) - f(x_t)$
- 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 $R_T = \tilde{O}(\sqrt{NT})$



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 Kernelized Contextual Bandits

Prior works in linear bandits & challenges in extension to kernelized bandits

Distributed 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]	$O(d\sqrt{NT}\log NT)$	$\tilde{O}(d^3N^2)$

Distributed kernelized contextual bandits:

- Assume f in RKHS: $f(x) = \phi(x)^\top\theta_\star$
  - $\phi: \mathbb{R}^d \to \mathbb{R}^p$ is a known feature map $\boxed{p \text{ is possibly infinite}}$
  - $\theta_\star \in \mathbb{R}^p$ is the unknown parameter
- Search for unknown reward function f in RKHS, which is a powerful tool for optimizing black box functions

Challenge: joint kernel estimation is communication expensive

- Empirical mean and variance

$$\hat{\mu}_{t,i}(\mathbf{x}) = \mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})^\top\left(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} + \lambda I\right)^{-1}\mathbf{y}_{\mathcal{D}_t(i)}$$

$$\hat{\sigma}_{t,i}(\mathbf{x}) = \lambda^{-1/2}\sqrt{k(\mathbf{x},\mathbf{x}) - \mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})^\top\left(\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} + \lambda I\right)^{-1}\mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x})}$$

where

$$\mathbf{K}_{\mathcal{D}_t(i)}(\mathbf{x}) = \mathbf{\Phi}_{\mathcal{D}_t(i)}\phi(\mathbf{x}) = [k(\mathbf{x}_s,\mathbf{x})]_{s\in\mathcal{D}_t(i)}^\top \in \mathbb{R}^{\boxed{|\mathcal{D}_t(i)|}}$$

grows linearly w.r.t. T

$$\mathbf{K}_{\mathcal{D}_t(i),\mathcal{D}_t(i)} = \mathbf{\Phi}_{\mathcal{D}_t(i)}^\top\mathbf{\Phi}_{\mathcal{D}_t(i)} = [k(\mathbf{x}_s,\mathbf{x}_{s'})]_{s,s'\in\mathcal{D}_t(i)} \in \mathbb{R}^{|\mathcal{D}_t(i)|\times|\mathcal{D}_t(i)|}$$

Proposed Solution

Nystrom Approximation

- Approximated mean and variance

$$\tilde{\mu}_{t,i}(\mathbf{x}) = z(\mathbf{x};\mathcal{S})^\top\left(\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} + \lambda\mathbf{I}\right)^{-1}\boxed{\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top\mathbf{y}_{\mathcal{D}_t(i)}}$$

$$\tilde{\sigma}_{t,i}(\mathbf{x}) = \lambda^{-1/2}\sqrt{k(\mathbf{x},\mathbf{x}) - z(\mathbf{x};\mathcal{S})^\top\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}[\boxed{\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}^\top\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}}} + \lambda\mathbf{I}]^{-1}z(\mathbf{x}|\mathcal{S})}$$

size $O(\gamma_{NT}^2)$, much more efficient to communicate

where

$\mathbf{Z}_{\mathcal{D}_t(i);\mathcal{S}} \in \mathbb{R}^{|\mathcal{D}_t(i)|\times|\mathcal{S}|}$ is obtained by applying embedding function $z(\cdot)$ to $\mathbf{\Phi}_{\mathcal{D}_t(i)}$

embedding function $z(\cdot)$ is shared by all N clients

Event-triggered communication

- if $\sum_{s\in\mathcal{D}_t(i)\setminus\mathcal{D}_{t_{\text{last}}}(i)} \tilde{\sigma}_{t_{\text{last}},i}^2(\mathbf{x}_s) > D$ for any client i

  update the shared embedding function $z(\cdot)$

  synchronize embedded statistics of all clients

Algorithm 2 Approximated Distributed Kernel UCB (Approx-DisKernelUCB)

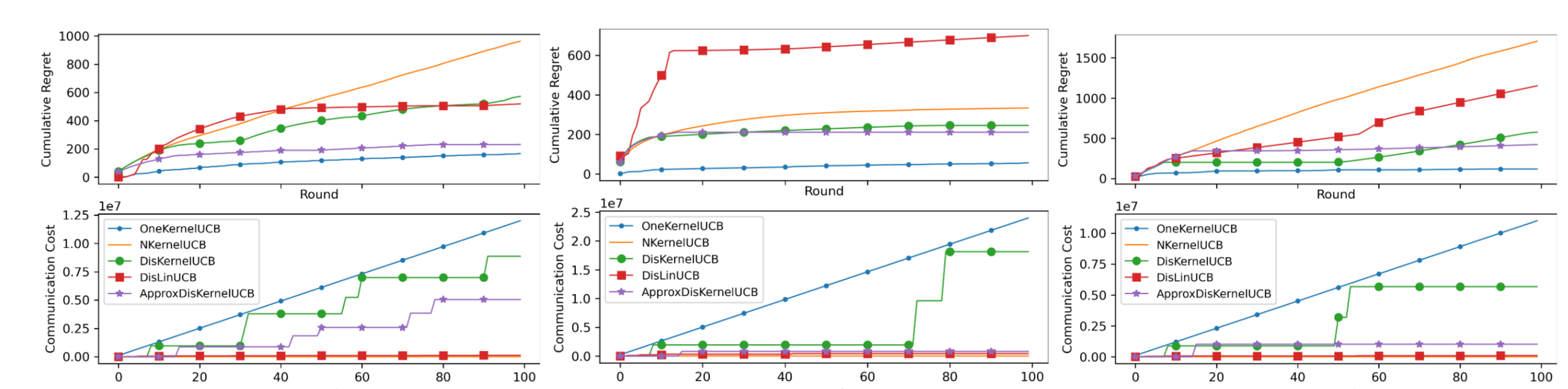
```
Input: threshold D > 0, regularization parameter λ > 0, δ ∈ (0,1) and kernel function k(·,·).
Initialize μ̃_{0,i}(x) = 0, σ̃_{0,i}(x) = sqrt(k(x,x)), N_0(i) = D_0(i) = ∅, ∀i ∈ [N]; S_0 = ∅, t_last = 0
for round l = 1, 2, ..., T do
  for client i = 1, 2, ..., N do
    [Client i] selects arm x_t ∈ A_t according to Eq (3) and observes reward y_t, where t := N(l − 1) + i
    [Client i] updates Z^T_{D_t(i);S_{t_last}} Z_{D_t(i);S_{t_last}} and Z^T_{D_t(i);S_{t_last}} y_{D_t(i)} using (z(x_t; S_{t_last}), y_t); sets N_t(i) = N_{t−1}(i) ∪ {t}, and D_t(i) = D_{t−1}(i) ∪ {t}
    // Global Synchronization
    if the event U_t(D) defined in Eq (4) is true then
      [Clients ∀i] sample S_{t,i} = RLS(N_t(i), q̄, σ̃²_{t_last,i}), and send {(x_s, y_s)}_{s∈S_{t,i}} to server
      [Server] aggregates and sends {(x_s, y_s)}_{s∈S_t} back to all clients, where S_t = ∪_{i∈[N]} S_{t,i}
      [Clients ∀i] compute and send {Z^T_{N_t(i);S_t} Z_{N_t(i);S_t}, Z^T_{N_t(i);S_t} y_{N_t(i)}} to server
      [Server] aggregates Σ_{i=1}^N Z^T_{N_t(i);S_t} Z_{N_t(i);S_t}, Σ_{i=1}^N Z^T_{N_t(i);S_t} y_{N_t(i)} and sends it back
      [Clients ∀i] updates Z^T_{D_t(i);S_t} Z_{D_t(i);S_t} and Z^T_{D_t(i);S_t} y_{D_t(i)}; sets D_t(i) = ∪_{i=1}^N N_t(i) = [t] and t_last = t
```

Theoretical Results

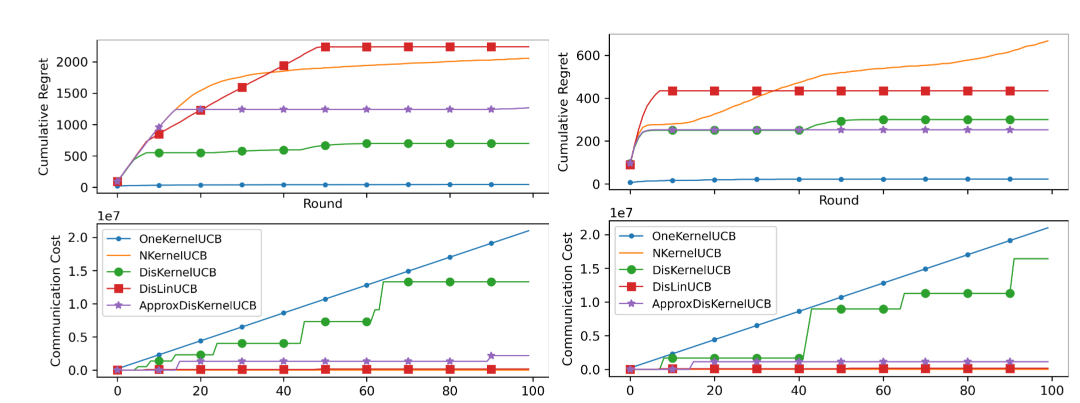
To attain near-optimal regret $R_T = O(\sqrt{NT}(\|\theta_\star\|\sqrt{\gamma_{NT}} + \gamma_{NT}))$, our proposed solution requires $C_T = O(\gamma_{NT}^3 N^2)$ communication, where

- γ_{NT} is the maximum information gain, $\gamma_{NT} = d\log NT$ for linear kernel, $\gamma_{NT} = \log^{d+1} NT$ for Gaussian kernel
- under linear setting, it matches C_T of dedicated distributed linear bandit algorithms [Li and Wang, AISTATS' 22, He et al., NeurIPS' 22] up to $O(\log^2 NT)$

Experiment Results



(a) MagicTelescope (b) Mushroom (c) Shuttle



(a) MovieLens (b) Yelp

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