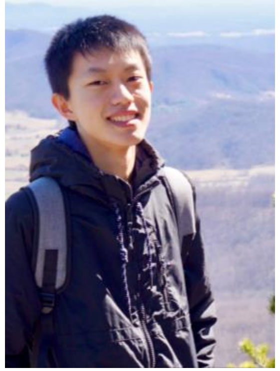
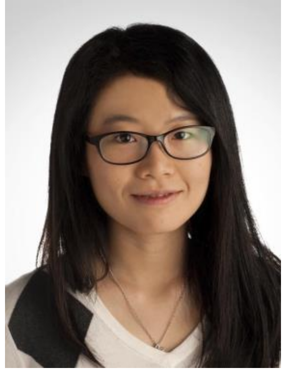
Learning Kernelized Contextual Bandits in a Distributed and Asynchronous Environment

Chuanhao Li, Huazheng Wang, Mengdi Wang, Hongning Wang

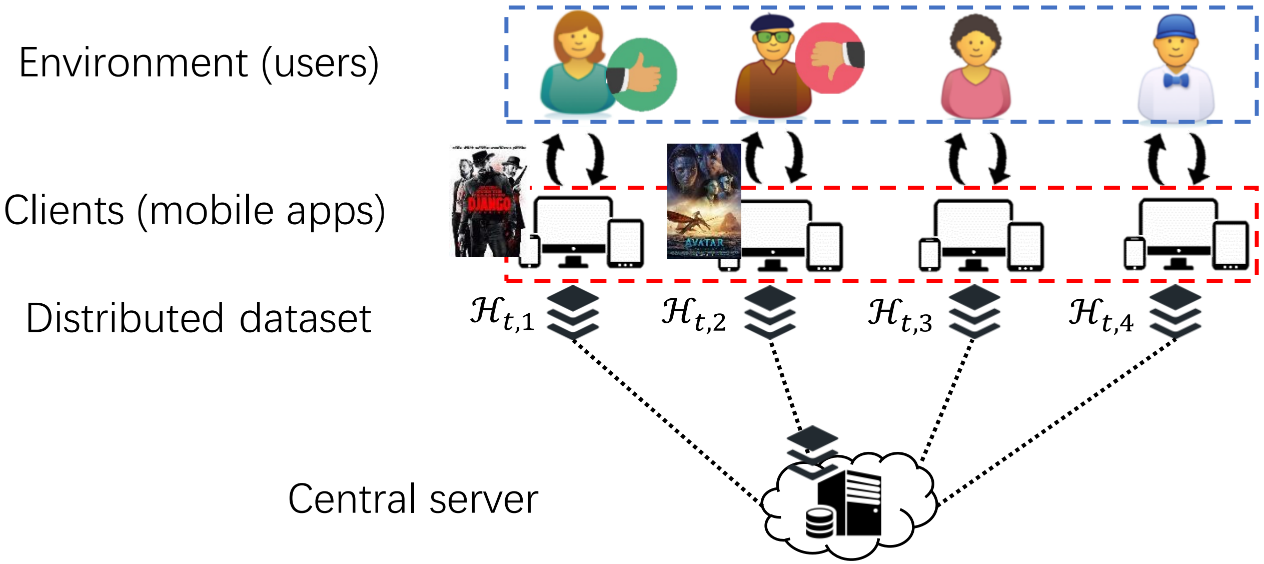




University of Virginia
Oregon State University
Princeton University

Distributed Bandit Learning

- For time step $t = 1,2,\dots,T$
  - An arbitrary client $i_t \in [N]$ becomes active
  - Client i_t picks arm x_t from a time-varying set $\mathcal{A}_t \subseteq \mathbb{R}^d$
  - Client i_t observes reward $y_t = f(x_t) + \eta_t$ from the environment
  - Communication between the server and clients

Example: movie recommendation



cumulative regret vs communication cost

- $R_T = \sum_{t=1}^{T} r_t$, where $r_t = \max_{x \in \mathcal{A}_t} f(x) - f(x_t)$
- C_T: total number of scalars transferred in the learning system

Prior Works

	Modeling Assumption	Communication Type	Regret R_T	Communication C_T
[Wang et al., ICLR'20]	linear	synchronous	$O(d\sqrt{T}\log T)$	$\tilde{O}(d^3 N^{1.5})$
[Li et al., NeurIPS'22]	**RKHS**	synchronous	$O(\sqrt{T}\gamma_T)$	$\tilde{O}(\gamma_T^3 N^2)$
[Li and Wang, AISTATS'22, He et al., NeurIPS'22]	linear	**asynchronous**	$O(d\sqrt{T}\log T)$	$\tilde{O}(d^3 N^2)$

more expressive model

more robust against stragglers

γ_T denotes the maximum information gain

Contribution of this work

- Propose the first asynchronous algorithm for distributed kernelized contextual bandit
- Still attain the same regret and communication guarantee as synchronous solution

Challenge with Joint Kernel Estimation

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

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

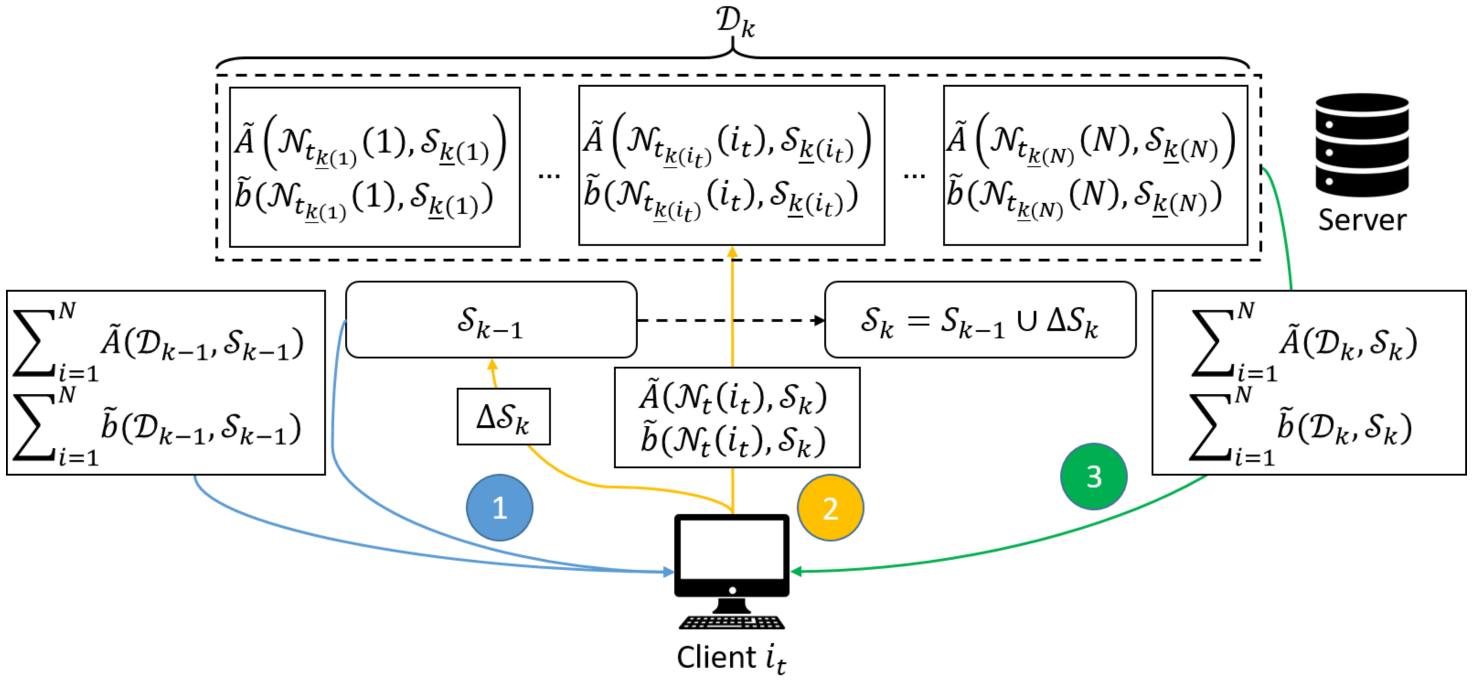
grows linearly wr.t. T

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

Solution Idea

- Communicate $O(\gamma_T)$ dim. embedded statistics to avoid C_T linear in T
  - Adopt Nystrom approximation as prior work [Li, et al. NeurIPS'22]
  - Propose a novel **async. update of the dictionary $\mathcal{S}$ and embedded statistics $\tilde{A}, \tilde{b}$**

If $\sum_{s \in \mathcal{N}_t(i_t) \setminus \mathcal{N}_{t_{\underline{k}(i_t)}}(i_t)} \tilde{\sigma}^2_{\underline{k}(i_t)}(\mathbf{x}_s) > D$ for the current client i_t:



Thank you!