Federated Linear Bandits with Finite Adversarial Actions

Li Fan [1], Ruida Zhou[2], Chao Tian[2], Cong Shen[1]

[1]Electrical & Computer Engineering, University of Virginia

[2]Electrical & Computer Engineering, Texas A&M University

November 14, 2023

Agenda

- Problem Formulation
- Preliminaries
- FedSupLinUCB Framework
- Algorithms
- Experiments
- Conclusion

Problem Formulation

- Star-shape system: M clients communicate with the central server.
- At round t, some clients $I_t \in [M]$ activate and receive K arms associated with contexts $\{x_{t,a}^i\}_{a\in[K]}$.
- Client $i \in I_t$ pulls arm a_t^i and receives reward $r_{t,a_t^i} = \theta^\top x_{t,a_t^i}^i + \epsilon_t$, where $\theta \in \mathbb{R}^d$ is unknown parameter, and ϵ_t is sub-Gaussian noise.
- Two arrival patterns: 1) Synchronous: all M clients are active at each round. 2) Asynchronous: one client is active at each round.
- Total regret: $R_T = \sum_{i=1}^M R_T^i = \sum_{i=1}^M \mathbb{E}\left[\sum_{t\in P_T^i} r_{t,a_t^{i,*}}^i - r_{t,a_t^i}^i\right]$, where P_T^i is the set of time indices at which client i is active.
- Communication cost: total number of communication rounds between clients and the server.

Preliminaries

Algorithm 1 Sync(s, server, client 1, … client n)

for $i = 1, 2, \ldots, n$ **do** ▷ Client-side local information upload
  Client i sends the local new layer s information $(\Delta A_s^i, \Delta b_s^i)$ to the server
end for
Update server's layer s information: ▷ Server-side information aggregation and distribution

$$A_s^{ser} \leftarrow A_s^{ser} + \sum\nolimits_{i=1}^{n} \Delta A_s^i, \quad b_s^{ser} \leftarrow b_s^{ser} + \sum\nolimits_{i=1}^{n} \Delta b_s^i$$

Send server information A_s^{ser}, b_s^{ser} back to all clients
for $i = 1, 2, \ldots, n$ **do**
  $A_s^i \leftarrow A_s^{ser}, b_s^i \leftarrow b_s^{ser}, \Delta A_s^i \leftarrow 0, \Delta b_s^i \leftarrow 0$ ▷ Client i updates the local information
end for

- Information encoding: matrix $A_n = \sum_{t=1}^{n} x_t x_t^\top$ and vector $b_n = \sum_{t=1}^{n} r_t x_t$, encoding function: $A_n, b_n \leftarrow Encode(\{x_t, r_t\}_{t=1}^{n})$.
- Communication criterion: 'doubling trick' introduced in [Abbasi-Yadkori et al., 2011] and the protocol by [Wang et al., 2019, He et al., 2022].
- Synchronization procedure: clients upload the local information, and the server aggregates and distributes updated information.

FedSupLinUCB Framework

Algorithm 2 S-LUCB

Initialization: $S = \lceil \log d \rceil$, $\overline{w}_0 = d^{1.5}/\sqrt{T}$, $\overline{w}_s \leftarrow 2^{-s}\overline{w}_0, \forall s \in [1:S]$.
$\alpha_0 = 1 + \sqrt{d \ln(2M^2T/\delta)}$, $\alpha_s \leftarrow 1 + \sqrt{2\ln(2KMT \ln d/\delta)}, \forall s \in [1:S]$
Input: Client i (with local information $A^i, b^i, \Delta A^i, \Delta b^i$), contexts set $\{x^i_{t,1}, \ldots, x^i_{t,K}\}$
$A^i_{t,s} \leftarrow A^i_s + \Delta A^i_s, b^i_{t,s} \leftarrow b^i_s + \Delta b^i_s$ or $A^i_{t,s} \leftarrow A^i_s, b^i_{t,s} \leftarrow b^i_s$ for `lazy update`
$\hat{\theta}_s \leftarrow (A^i_{t,s})^{-1} b^i_{t,s}$, $\hat{r}^i_{t,s,a} = \hat{\theta}_s^\top x^i_{t,a}$, $w^i_{t,s,a} \leftarrow \alpha_s \|x^i_{t,a}\|_{(A^i_{t,s})^{-1}}, \forall s \in [0:S], \forall a \in [K]$
$s \leftarrow 0$; $\mathcal{A}_0 \leftarrow \{a \in [K] \mid \hat{r}^i_{t,0,a} + w^i_{t,0,a} \geq \max_{a \in [K]}(\hat{r}^i_{t,0,a} - w^i_{t,0,a})\}$ ▷ Initial screening
repeat ▷ Layered successive screening
  if $s = S$ **then**
    Choose action a^i_t arbitrarily from $\mathcal{A}_S$
  else if $w^i_{t,s,a} \leq \overline{w}_s$ for all $a \in \mathcal{A}_s$ **then**
    $\mathcal{A}_{s+1} \leftarrow \{a \in \mathcal{A}_s \mid \hat{r}^i_{t,s,a} \geq \max_{a' \in \mathcal{A}_s}(\hat{r}^i_{t,s,a'}) - 2\overline{w}_s\}$; $s \leftarrow s + 1$
  else
    $a^i_t \leftarrow \arg\max_{\{a \in \mathcal{A}_s, w^i_{t,s,a} > \overline{w}_s\}} w^i_{t,s,a}$
  end if
until action a^i_t is found
Take action a^i_t and and receive reward r^i_{t,a^i_t}
$\Delta A^i_s \leftarrow \Delta A^i_s + x^i_{t,a^i_t} x^{i\top}_{t,a^i_t}$, $\Delta b^i_s \leftarrow \Delta b^i_s + r^i_{t,a^i_t} x^i_{t,a^i_t}$ ▷ Update local information
Return layer index s

- Combination the principles of SupLinUCB [Chu et al., 2011] and OFUL [Abbasi-Yadkori et al., 2011, Ruan et al., 2021].
- Maintain $S = \lceil \log d \rceil$ information layers.
- Accuracy strengthens as the layer index increases.

Asynchronous FedSupLinUCB

Algorithm 3 Async-FedSupLinUCB

Initialization: $T, C, S = \lceil \log d \rceil$

$\{A_s^{ser} \leftarrow I_d, b_s^{ser} \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization

$\{A_s^i \leftarrow I_d, \Delta A_s^i, b_s^i, \Delta b_s^i \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization

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

  Client $i_t = i$ is active, and observes K contexts $\{x_{t,1}^i, x_{t,2}^i, \cdots, x_{t,K}^i\}$

  $s \leftarrow$ S-LUCB $\left(\text{client } i, \{x_{t,1}^i, x_{t,2}^i, \cdots, x_{t,K}^i\}\right)$ with lazy update

  if $\frac{\det\left(A_s^i + \Delta A_s^i\right)}{\det(A_s^i)} > (1 + C)$ **then**

    Sync(s, server, clients i) for each $s \in [0:S]$

  end if

end for

- Only one client is active in each round.
- Global synchronization and coordination are not required

Synchronous FedSupLinUCB

Algorithm 4 `Sync-FedSupLinUCB`

Initialization: T_c, D, $S = \lceil \log d \rceil$, $t^s_{last} \leftarrow 0, \forall s \in [0:S]$, CommLayers $\leftarrow \emptyset$.
$\{A^{ser}_s \leftarrow I_d, b^{ser}_s \leftarrow 0 \mid s \in [0:S]\}$ ▷ Server initialization
$\{A^i_s \leftarrow I_d, \Delta A^i_s, b^i_s, \Delta b^i_s \leftarrow 0 \mid s \in [0:S], i \in [M]\}$ ▷ Clients initialization
for $t = 1, 2, \cdots, T_c$ **do**
  for $i = 1, 2, \cdots, M$ **do**
    Client $i_t = i$ is active, and observes K contexts $\{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}$
    $s \leftarrow$`S-LUCB` $\left(\text{client } i, \{x^i_{t,1}, x^i_{t,2}, \cdots, x^i_{t,K}\}\right)$
    if $(t - t^s_{last}) \log \frac{\det\left(A^i_s + \Delta A^i_s\right)}{\det(A^i_s)} > D$ **then**
      Add s to CommLayers
    end if
  end for
end for
for $s \in$ CommLayers **do**
  Sync(s, server, clients $[M]$); $t^s_{last} \leftarrow t$, CommLayers $\leftarrow \emptyset$
end for

- All clients are active and make decisions at each round.
- Refined communication criterion to enable time-independent communication cost

Performance

Theorem

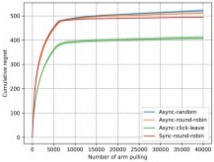
For any $0 < \delta < 1$, if we run Async with $C = 1/M^2$, then with probability at least $1 - \delta$, the regret of the algorithm is bounded as $R_T \leq \tilde{O}\left(\sqrt{d \sum_{i=1}^{M} T_i}\right) = \tilde{O}\left(\sqrt{dT}\right)$. Moreover, the corresponding communication cost is bounded by $O(dM^2 \log d \log T)$.

Theorem

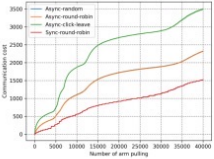
For any $0 < \delta < 1$, if we run Sync with $D = \frac{T_c \log T_c}{d^2 M}$, with probability at least $1 - \delta$, the regret of the algorithm is bounded as $R_T \leq \tilde{O}(\sqrt{dMT_c})$ where T_c is the total per-client arm pulls. Moreover, the corresponding communication cost is bounded by $O(\sqrt{d^3 M^3} \log d)$.

- Both achieve minimax optimal regret with sublinear communication cost.

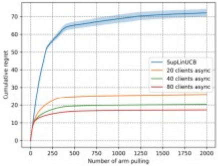
Experiments



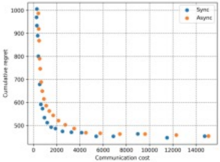
(a) Regret: arrival patterns.



(b) Communication: arrival patterns.



(c) Regret: client numbers.



(d) Regret vs communications.

- Compare with different arrival patterns and amount of clients.
- The trade-off between regrets and communications.

Thank you!

References I

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011).
Improved algorithms for linear stochastic bandits.
Advances in neural information processing systems, 24.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011).
Contextual bandits with linear payoff functions.
In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings.

He, J., Wang, T., Min, Y., and Gu, Q. (2022).
A simple and provably efficient algorithm for asynchronous federated contextual linear bandits.
arXiv preprint arXiv:2207.03106.

Ruan, Y., Yang, J., and Zhou, Y. (2021).
Linear bandits with limited adaptivity and learning distributional optimal design.
In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–87.

Wang, Y., Hu, J., Chen, X., and Wang, L. (2019).
Distributed bandit learning: Near-optimal regret with efficient communication.
In *International Conference on Learning Representations*.