# Follow-ups Also Matter: Improving Contextual Bandits via Post-serving Contexts

Chaoqi Wang[1], Ziyu Ye[1], Zhe Feng[2],
Ashwinkumar Badanidiyuru[3], Haifeng Xu[1]

The University of Chicago[1]
Google Research[2]
Google[3]

NeurIPS 2023

# Background

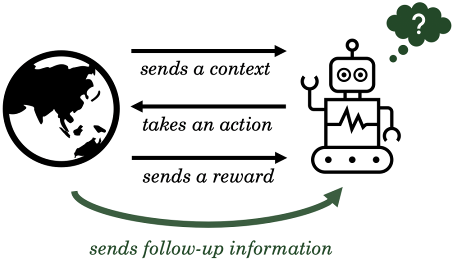


Figure 1: Illustration of learning with post-serving contexts.

- **Motivation**: Post-serving contexts are prevalent in recommender systems.
- **Challenges**: Classical bandit algorithms often fall short in such scenarios.
- **Research question**: How to effectively utilize **post-serving information** in **linear contextual bandits**?

# Problem Setup and Notations

- **Problem Setup**: Each time $t = 1, 2, \cdots, T$:
  - The learner observes the context $\boldsymbol{x}_t$.
  - The learner selects an arm $a_t \in [K]$.
  - The learner observes the reward $r_{t,a_t}$.
  - **The learner observes the post-serving context $\boldsymbol{z}_t$.**
- **Notations**:
  - Actions space: $\mathcal{A} = [K]$.
  - Pre-serving context: $\boldsymbol{x} \in \mathbb{R}^{d_{\boldsymbol{x}}}$; post-serving context: $\boldsymbol{z} \in \mathbb{R}^{d_{\boldsymbol{z}}}$.
    - $\boldsymbol{z} = \phi^\star(\boldsymbol{x}_t) + \boldsymbol{\epsilon}_t$, and $\phi^\star(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{z} \mid \boldsymbol{x}]$
  - Reward function:
    - $r_a(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^\top \boldsymbol{\theta}_a^\star + \boldsymbol{z}^\top \boldsymbol{\beta}_a^\star + \eta$, where $\eta$ is $R_\eta$-sub-Gaussian.
  - Matrix representation:
    - $\boldsymbol{X}_t = \sum_{s=1}^t \boldsymbol{x}_s \boldsymbol{x}_s^\top + \lambda \boldsymbol{I}$ and $\boldsymbol{Z}_t = \sum_{s=1}^t \boldsymbol{z}_s \boldsymbol{z}_s^\top + \lambda \boldsymbol{I}$.
  - Norm restrictions:
    - $\forall a \in \mathcal{A}, \|\boldsymbol{\theta}_a^\star\|_2 \le 1, \|\boldsymbol{\beta}_a^\star\|_2 \le 1$; $\|\boldsymbol{x}\|_2 \le L_x, \|\boldsymbol{z}\|_2 \le L_z$.

# Our Contributions

- **New framework**:
  - Proposed a novel family of contextual bandits with post-serving contexts.
- **Enhanced lemma**:
  - Introduced the Generalized Elliptical Potential Lemma (EPL).
- **Algorithm and theory**:
  - Designed `poLinUCB` with a regret bound of $\widetilde{\mathcal{O}}(T^{1-\alpha} d_u^{\alpha} + d_u\sqrt{TK})$.
- **Empirical validation**:
  - Achieved SOTA performance on synthetic and real-world datasets.

# Assumption: Generalized Learnability of $\phi^*(\cdot)$

## Learnability Assumption

There exists an algorithm that, given $t$ pairs of examples $\{(\boldsymbol{x}_s, \boldsymbol{z}_s)\}_{s=1}^t$ with arbitrarily chosen $\boldsymbol{x}_s$'s, outputs an estimated function of $\phi^\star : \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ such that for any $\boldsymbol{x} \in \mathbb{R}^{d_x}$, the following holds with probability at least $1 - \delta$,

$$e_t^\delta := \left\| \widehat{\phi}_t(\boldsymbol{x}) - \phi^\star(\boldsymbol{x}) \right\|_2 \leq C_0 \cdot \left( \|\boldsymbol{x}\|^2_{\boldsymbol{X}_t^{-1}} \right)^{\alpha} \cdot \log(t/\delta),$$

where $\alpha \in (0, 1/2]$ and $C_0$ is some universal constant.

- The larger the value of $\alpha$, the faster the learning rate for $\phi^\star(\cdot)$.
- The regret of our algorithm is tied to $O(T^{1-\alpha})$.
- For linear functions, $\alpha = 1/2$ is optimal, yielding a regret of $O(\sqrt{T})$.

# Why Natural Attempts May be Inadequate?

- Similar to [Wang et al., 2016][1], a natural idea is to fit $\widehat{\phi}(\cdot)$, and obtain the parameter estimate by solving:

$$\ell_t(\boldsymbol{\theta}_a, \boldsymbol{\beta}_a) = \sum_{s \in [t]: a_s = a} \left( r_{s,a} - \boldsymbol{x}_t^\top \boldsymbol{\theta}_a - \widehat{\phi}_s(\boldsymbol{x}_s)^\top \boldsymbol{\beta}_a \right)^2 + \lambda \left( \|\boldsymbol{\theta}_a\|_2^2 + \|\boldsymbol{\beta}_a\|_2^2 \right).$$

  - The regret can be $\widetilde{\mathcal{O}}(T^{3/4})$ when initialized away from the global optimum.

- We propose to get the parameter estimate by solving:

$$\ell_t(\boldsymbol{\theta}_a, \boldsymbol{\beta}_a) = \sum_{s \in [t]: a_s = a} \left( r_{s,a} - \boldsymbol{x}_s^\top \boldsymbol{\theta}_a - \boldsymbol{z}_s^\top \boldsymbol{\beta}_a \right)^2 + \lambda \left( \|\boldsymbol{\theta}_a\|_2^2 + \|\boldsymbol{\beta}_a\|_2^2 \right).$$

  - This requires modification over the original Elliptical Potential Lemma (EPL) to accommodate noise in contexts during learning.

[1] Huazheng Wang, Qingyun Wu, and Hongning Wang. "Learning Hidden Features for Contextual Bandits". In: *CIKM*. 2016, pp. 1633–1642.

# The Proposed Lemma: Generalized EPL

## Generalized Elliptical Potential Lemma[2]

Suppose (1) $\boldsymbol{X}_0 \in \mathbb{R}^{d\times d}$ is any positive definite matrix; (2) $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \in \mathbb{R}^d$ and $\max_t \|\boldsymbol{x}_t\| \le L_x$; (3) $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T \in \mathbb{R}^d$ are independent bounded zero-mean noises satisfying $\max_t \|\boldsymbol{\epsilon}_t\| \le L_\epsilon$ and $\mathbb{E}[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t^\top] \succcurlyeq \sigma_\epsilon^2 \boldsymbol{I}$; and (4) $\widetilde{\boldsymbol{X}}_t$ is defined as:

$$\widetilde{\boldsymbol{X}}_t = \boldsymbol{X}_0 + \sum_{s=1}^{t} (\boldsymbol{x}_s + \boldsymbol{\epsilon}_s)(\boldsymbol{x}_s + \boldsymbol{\epsilon}_s)^\top \in \mathbb{R}^{d\times d}.$$

For any $p \in [0,1]$, the following inequality holds with probability at least $1-\delta$,

$$\sum_{t=1}^{T} \left(1 \wedge \|\boldsymbol{x}_t\|^2_{\widetilde{\boldsymbol{X}}_{t-1}^{-1}}\right)^p \le 2^p T^{1-p} \log^p \left(\frac{\det \boldsymbol{X}_T}{\det \boldsymbol{X}_0}\right) + \frac{8L_\epsilon^2 (L_\epsilon + L_x)^2}{\sigma_\epsilon^4} \log\left(\frac{32 d L_\epsilon^2 (L_\epsilon + L_x)^2}{\delta \sigma_\epsilon^4}\right).$$

[2] The original EPL corresponds to the specific case of $p = 1$.

# The Proposed Algorithm: poLinUCB

**Algorithm 1** `poLinUCB` (Linear UCB with post-serving contexts)

1: **for** $t = 0, 1, \ldots, T$ **do**
2: Receive the pre-serving context $\boldsymbol{x}_t$.
3: Compute the optimistic parameters by maximizing the UCB objective:

$$\left(a_t, \widetilde{\phi}_t(\boldsymbol{x}_t), \widetilde{\boldsymbol{w}}_t\right) = \underset{(a,\phi,\boldsymbol{w}_a)\in[K]\times\mathcal{C}_{t-1}\left(\widehat{\phi}_{t-1},\boldsymbol{x}_t\right)\times\mathcal{C}_{t-1}(\widehat{\boldsymbol{w}}_{t-1,a})}{\arg\max} \begin{bmatrix} \boldsymbol{x}_t \\ \phi(\boldsymbol{x}_t) \end{bmatrix}^\top \boldsymbol{w}_a.$$

4: Play the arm $a_t$ and receive the post-serving context $\boldsymbol{z}_t$ and the reward $r_{t,a_t}$.
5: Compute $\widehat{\boldsymbol{w}}_{t,a}$ for each $a \in \mathcal{A}$ using:

$$\ell_t\left(\boldsymbol{\theta}_a, \beta_a\right) = \sum_{s\in[t]:a_s=a} \left(r_{s,a} - \boldsymbol{x}_s^\top \boldsymbol{\theta}_a - \boldsymbol{z}_s{}^\top \beta_a\right)^2 + \lambda\left(\|\boldsymbol{\theta}_a\|_2^2 + \|\beta_a\|_2^2\right).$$

6: Compute the estimated post-serving context generating function $\widehat{\phi}_t(\cdot)$ using ERM.
7: Update confidence sets $\mathcal{C}_t(\widehat{\boldsymbol{w}}_{t,a})$ and $\mathcal{C}_t(\widehat{\phi}_t, \boldsymbol{x}_t)$ for each $a$.
8: **end for**

# Regret Analysis

| Settings | Ours |
|---|---|
| Ours | $\widetilde{\mathcal{O}}\left(T^{1-\alpha}d_u^{\alpha} + d_u\sqrt{TK}\right)$ |
| Action-dependent contexts | $\widetilde{\mathcal{O}}\left(T^{1-\alpha}d_u^{\alpha}\sqrt{K} + d_u\sqrt{TK}\right)$ |
| Same setting as in [Abbasi et al., 2011][3] | $\widetilde{\mathcal{O}}\left(T^{1-\alpha}d_u^{\alpha} + d_u\sqrt{T}\right)$ |

Table 1: Upper bound of regret of `poLinUCB`.

[3] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. "Improved Algorithms for Linear Stochastic Bandits". In: *Advances in neural information processing systems* 24 (2011).

# Experimental Results: The Synthetic Dataset

- Our proposed `poLinUCB` consistently outperforms other strategies.
  (Except for `LinUCB` ($x$ and $z$) which equips with post-serving contexts in arm selection.)

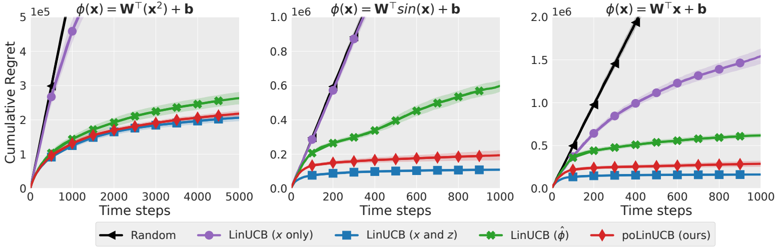


Figure 2: Algorithms' cumulative regrets in three synthetic environments. The shaded area denotes the standard error computed using 10 different random seeds.

# Experimental Results: The MovieLens Dataset[4]

- Our proposed `poLinUCB` consistently outperforms other strategies.
  (Except for `LinUCB` ($x$ and $z$) which equips with post-serving contexts in arm selection.)

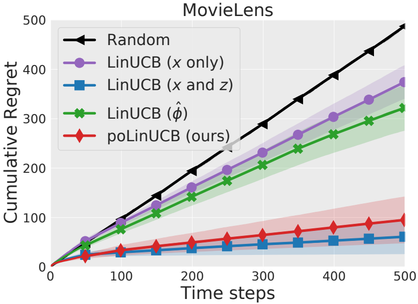


Figure 3: Algorithms' cumulative regrets in the MoiveLens Dataset. The shaded area denotes the standard error computed using 10 different random seeds.

[4] F Maxwell Harper and Joseph A Konstan. "The MovieLens Datasets: History and Context". In: *Acm Transactions on Interactive Intelligent Systems* 5.4 (2015), pp. 1–19.

# Thank you.

Please refer to our paper for more information:

https://arxiv.org/abs/2309.13896.