Infrequent Exploration in Linear Bandits

Harin Lee (University of Washington)
Min-hwan Oh (Seoul National University)

Linear Bandits

Linear bandits

- Arm set $\mathcal{X} \subset \mathbb{R}^d$
- True parameter $\theta^* \in \mathbb{R}^d$
- Interaction : for $t = 1, \ldots, T$

  choose $X_t \in \mathcal{X}$, receive $Y_t = \underbrace{X_t^\top \theta^*}_{\text{expected reward}} + \underbrace{\eta_t}_{\text{noise}}$

Performance measurement: Minimize cumulative regret

$$\mathcal{R}_{\mathsf{Alg}}(T) := \sum_{t=1}^{T} \underbrace{x^{*\top} \theta^*}_{\text{optimal reward}} - \underbrace{X_t{}^\top \theta^*}_{\text{agent's reward}}$$

(Optimal arm : $x^* = \operatorname{argmax}_{x \in \mathcal{X}} x^\top \theta^*$)

Minimum gap $\Delta := x^{*\top} \theta^* - \max_{x \in \mathcal{X} \setminus \{x^*\}} x^\top \theta^*$

Exploration-Exploitation Trade-off

Exploitation : Choosing the best arm based on current information
Exploration : Choosing another arm to gain information

- Adaptive balancing
  - Explores adaptively *every time step*.
  - ex.) `LinUCB` [1], `LinTS` [2]

- Pure exploitation (greedy policy)
  - Exploration may be costly, risky, ethically problematic in healthcare or safety-critical setting.
  - Works only under contextual setting with certain distributional assumption on the context [4, 7, 3, 6, 5].

Can we bridge these two extremes?

Algorithmic Framework : INFEX

Algorithm INFEX(Alg, $\mathcal{T}_e$): INFrequent EXploration

Input : Base algorithm Alg, exploration schedule $\mathcal{T}_e \subset \mathbb{N}$
Initialize $V_0 = I_d$
for $t = 1, 2, ...,$ **do**
  if $t \in \mathcal{T}_e$ **then** ▷ *Exploratory selection*
    Choose X_t according to Alg and observe Y_t
  else ▷ *Greedy selection*
    Compute ridge estimator $\hat{\theta}_{t-1} = V_{t-1}^{-1} \sum_{i=1}^{t-1} X_i Y_i$
    Choose $X_t = \text{argmax}_{x \in \mathcal{X}} x^\top \hat{\theta}_{t-1}$ and observe Y_t
  end if
  Update $V_t = V_{t-1} + X_t X_t^\top$
end for

- Performs Alg infrequently based on $\mathcal{T}_e$.
- Can take any base algorithm Alg (e.g. LinUCB, LinTS) and any exploration schedule $\mathcal{T}_e$.

Main Result

Theorem : Regret bound of INFEX

If the base algorithm Alg achieves polylogarithmic regret and the number of exploratory time steps $f(t) := |\mathcal{T}_e \cap \{1, \dots, t\}|$ satisfies $f(t) = \omega(\log t)$, then we have

$$\mathcal{R}_{\texttt{INFEX}(\mathsf{Alg}, \mathcal{T}_e)}(T) \leq \mathcal{R}_{\mathsf{Alg}}(f(T)) + G_{\mathrm{const}}(\mathsf{Alg}, \mathcal{T}_e) + G(T)\,,$$

where

$$G(T) = \mathcal{O}\left(\frac{1}{\Delta}\left(\log T + d \log\left(\frac{d}{\Delta}\log T\right)\right)^2\right)$$

- $G_{\mathrm{const}}(\mathsf{Alg}, \mathcal{T}_e)$ does **not** depend on T.
  - Becomes larger as the regret of Alg becomes larger or the exploration schedule $\mathcal{T}_e$ becomes sparser.
- $G(T)$ is on the same scale with the instance-dependent regret bound of LinUCB [1].

Experiments

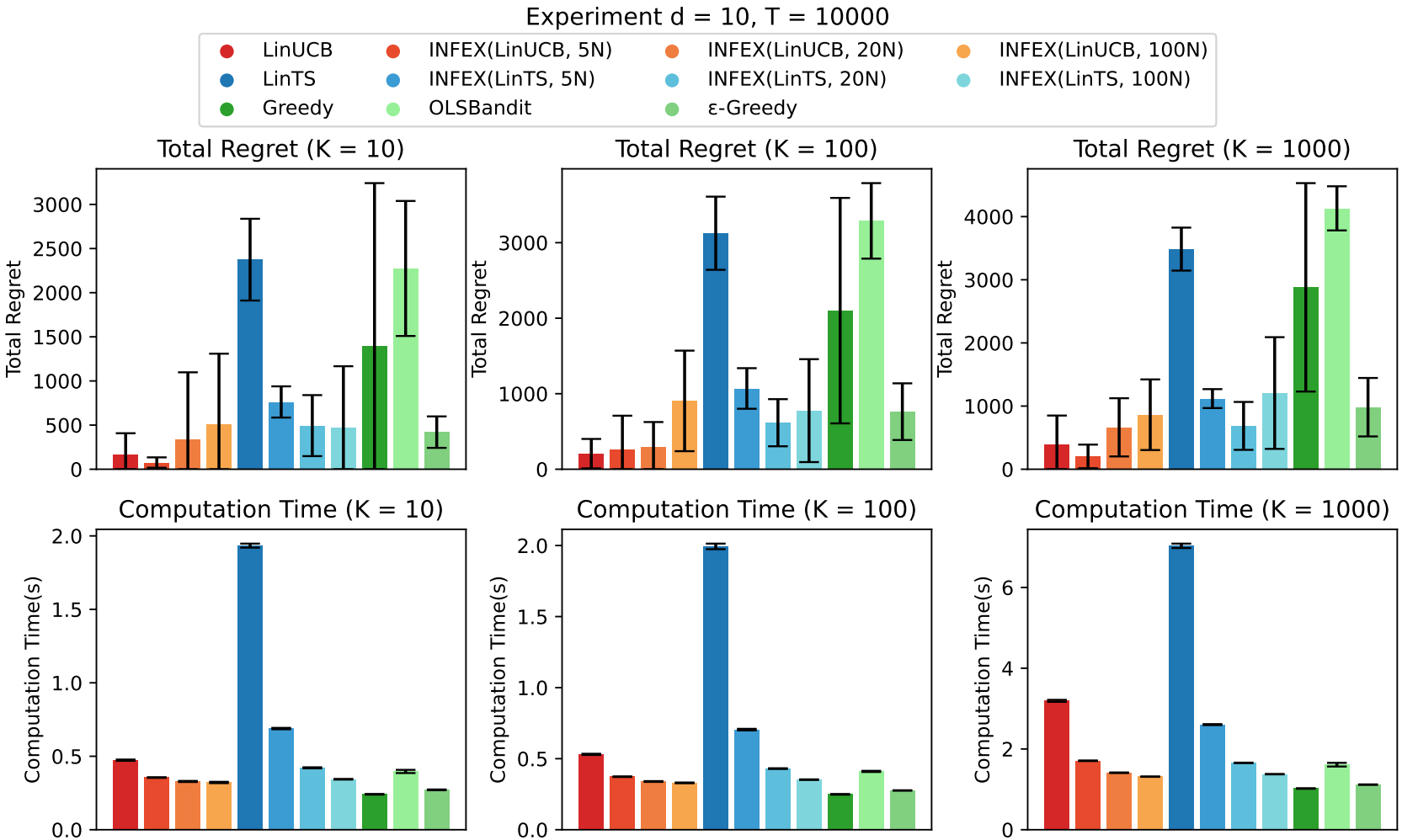

Figure: Comparison of total regret (top) and computation time (bottom) when $d = 10$, $T = 10000$, and $K = 10$ (left), $K = 100$ (middle), and $K = 1000$ (right).

Summary

- We propose an algorithmic framework `INFEX` that smoothly bridges any fully adaptive linear bandit algorithms and purely greedy policy.
- We show that as long as the number of exploratory selections is in $\omega(\log t)$, `INFEX` achieves $\mathcal{O}(\mathrm{poly}\log T)$ regret independently of the exploration schedule.
- Empirical results demonstrate that `INFEX` outperforms the baselines in cumulative regret and computational efficiency.

References I

[1] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. Advances in Neural Information Processing Systems, 24:2312–2320.

[2] Abeille, M. and Lazaric, A. (2017). Linear Thompson Sampling Revisited. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54, pages 176–184. PMLR, PMLR.

[3] Bastani, H., Bayati, M., and Khosravi, K. (2021). Mostly exploration-free algorithms for contextual bandits. Management Science, 67(3):1329–1349.

[4] Kannan, S., Morgenstern, J. H., Roth, A., Waggoner, B., and Wu, Z. S. (2018). A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. Advances in neural information processing systems, 31.

[5] Kim, S.-J. and Oh, M.-h. (2024). Local anti-concentration class: Logarithmic regret for greedy linear contextual bandit. Advances in Neural Information Processing Systems, 37:77525–77592.

[6] Raghavan, M., Slivkins, A., Vaughan, J. W., and Wu, Z. S. (2023). Greedy algorithm almost dominates in smoothed contextual bandits. SIAM Journal on Computing, 52(2):487–524.

[7] Sivakumar, V., Wu, S., and Banerjee, A. (2020). Structured linear contextual bandits: A sharp and geometric smoothed analysis. In International Conference on Machine Learning, pages 9026–9035. PMLR.