Optimal Exploration is no harder than Thompson Sampling

Zhaoqi Li, Kevin Jamieson, Lalit Jain

University of Washington

Motivation

$\theta_\star$: unknown population preferences

Goal: find the best action to maximize profit!

However, large companies have millions of users and millions of items are listed

⇒ **Can we find the best action quickly on large scale?**

Problem Statement

Input: $\mathcal{X} \subset \mathbb{R}^d$

for $t = 1,2,\ldots$

1. Learner chooses $x_t \in \mathcal{X}$
2. Nature reveals $y_t = \langle x_t, \theta_\star \rangle + \epsilon_t$, $\epsilon_t \sim N(0,1)$

Goal: identify $x_\star := \arg\max_{x\in\mathcal{X}} \langle x, \theta_\star \rangle$ as quickly as possible

Key Advantages of Our Algorithm

- Achieves **optimal** rate ⇒ find best action quickly
- Algorithm is **easy** to implement on **large scale**

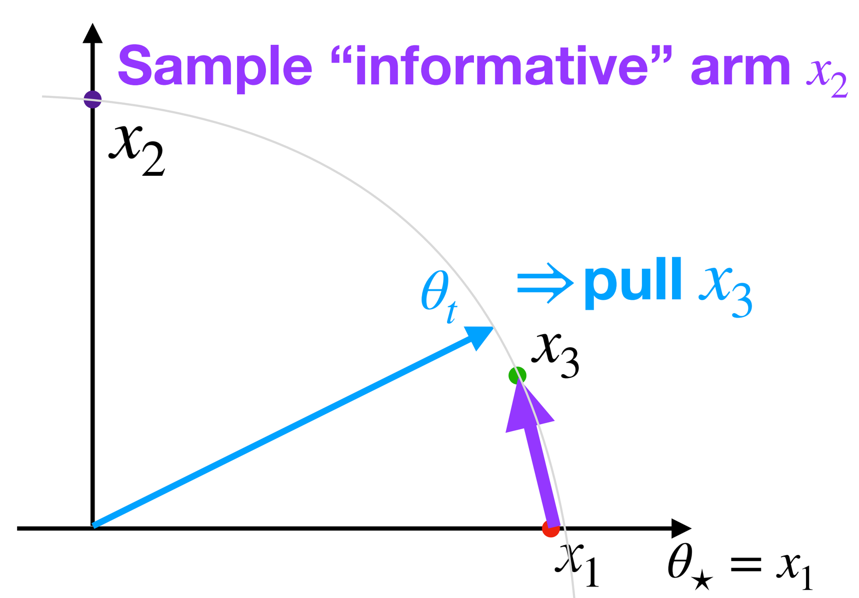
Related Work

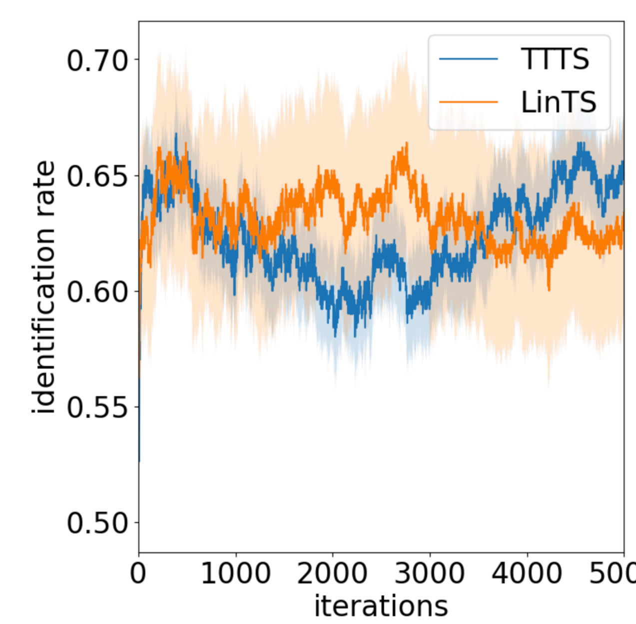
Existing Optimal Approaches: [Xu et al. 2018], [Fiez et al. 2019], [Degenne et al. 2020]

- Require **enumeration of $\mathcal{X}$ or complicated projections**

Thompson Sampling:

- Regret minimizing algorithm and **computationally easy**
- However **suboptimal** for BAI on certain instances





Is there an algorithm as easy as TS yet still optimal?

Key Ideas behind Our Algorithm

The optimal allocation is implied by the lower bound $\tau_\star$

⇒ solve the saddle point problem using online learning!

$$\tau_\star := \max_{\lambda\in\Delta_{\mathcal{X}}} \min_{\theta\in\Theta^c_{x_\star}} \|\theta_\star - \theta\|^2_{A(\lambda)} = \max_{\lambda\in\Delta_{\mathcal{X}}} \min_{p\in\Delta(\Theta^c_{x_\star})} \mathbb{E}_{\theta\sim p}[\|\theta_\star - \theta\|^2_{A(\lambda)}]$$

Maintain a distribution over $\Theta^c_{x_\star} := \{\theta \in \Theta : x_\star \neq \arg\max_{x\in\mathcal{X}} x^\top\theta\}$

Idea: Two player zero-sum game

1. Max-player: Exponential Weights on $\lambda \in \Delta_{\mathcal{X}}$
2. Min-player: Posterior Updates on $p \in \Delta(\Theta^c_{x_\star})$

$$p_{t+1,\theta} \propto \exp\Big(-(\theta - \hat\theta_{t+1})^\top \Big(\textstyle\sum_{s=1}^t x_s x_s^\top\Big)(\theta - \hat\theta_{t+1})\Big)$$

$$\approx \exp\Big(-\textstyle\sum_{s=1}^t (x_s^\top(\theta - \hat\theta_s))^2\Big) \propto p_{t,\theta} \exp\left(-(x_t^\top(\theta - \hat\theta_t))^2\right)$$

⇒ **exponential weights update!**

Also, **sampling from p is easy** since p_{t+1} is Gaussian posterior

By OCO argument, $(\lambda_t, p_t) \to \max_{\lambda\in\Delta_{\mathcal{X}}} \min_{p\in\Delta(\Theta^c_{x_\star})} \mathbb{E}_{\theta\sim p}[\|\theta_\star - \theta\|^2_{A(\lambda)}] =: \tau_\star$

Our Algorithm

PEPS: Pure Exploration with Projection-Free Sampling

Input: $\mathcal{X}$, T, η

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

1. compute the *leader* $\hat x_t = \arg\max_{x\in\mathcal{X}} x^\top \hat\theta_t$
2. sample a *challenger* $\theta_t \sim p_t$ whose best arm is not $\hat x_t$
3. sample $x_t \sim \lambda_t$, observe y_t
4. Update $\hat\theta_{t+1}$, $\lambda_{t+1,x} \leftarrow \lambda_{t,x} e^{\eta(x^\top(\theta_t - \hat\theta_t))^2}$
5. Update $p_{t+1} = N(\hat\theta_{t+1}, (\sum_{s=1}^t x_s x_s^\top)^{-1})$ ← **Posterior Update**

pull arm x maximizing $|x^\top(\theta_t - \hat\theta_t)|$

Return $\hat x = \arg\max_{x\in\mathcal{X}} x^\top\theta, \theta \sim p_T$

Performance Guarantee

Theorem (Li, Jamieson, Jain) **Set $\eta = O(1/\sqrt{T})$, and assume Θ is bounded. Then with probability 1**

$$\lim_{T\to\infty} -\frac{1}{T} \log \mathbb{P}_{\theta\sim p_T}(\hat x_T \neq x_\star) = \tau_\star$$

optimal rate!

Experimental Results

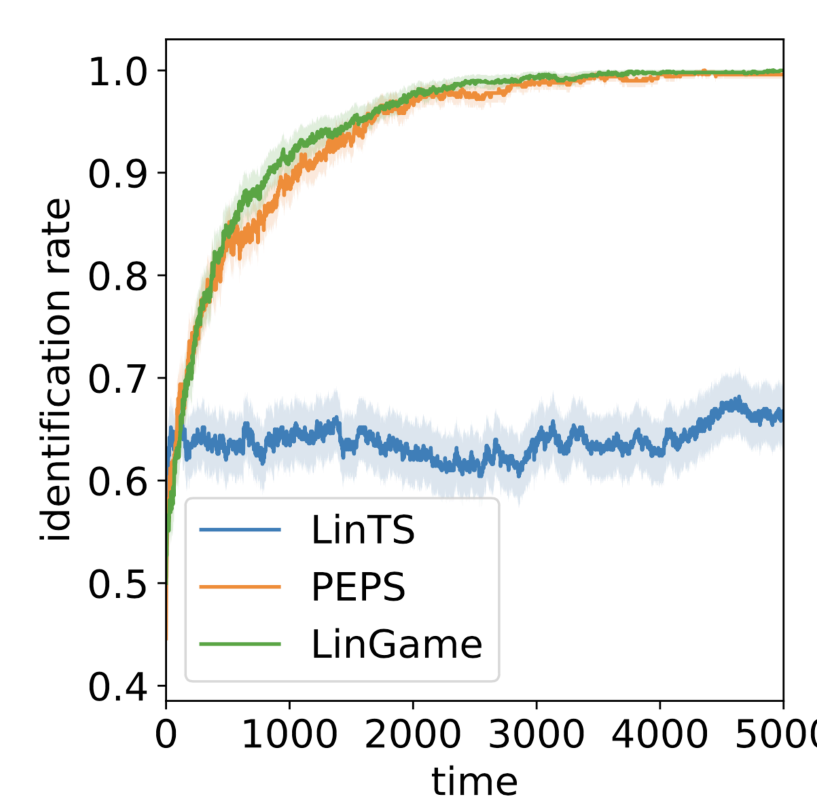


Figure: Hard Instance

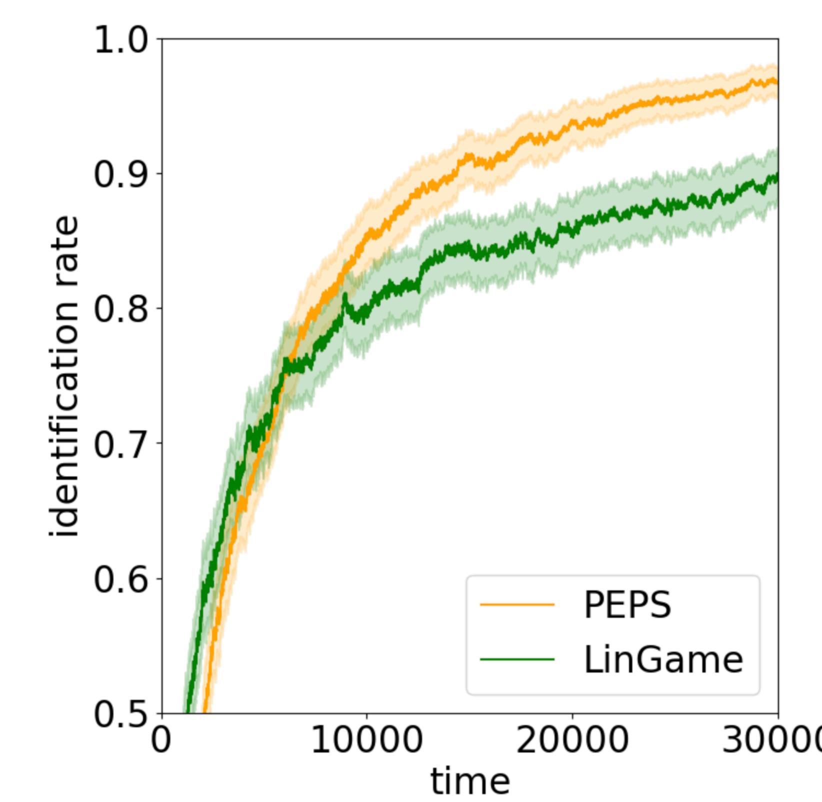


Figure: TopK Instance