Deployment Efficient Reward-Free Exploration with Linear Function Approximation

Zihan Zhang, Lin F. Yang, Yuxin Chen, Jason D. Lee, Simon S. Du, Ruosong Wang

Nov 4, 2025

Motivation

- Limitations in deploying a new policy
  - High deployment cost (e.g., clinical trial)
  - Restricted updates (e.g., recommendation system)
- Agnostic/reward-free learning
  - Adaptivity to unstable environment

Problem Settings: RL with linear function approximation

- An Markov Decesion Process (MDP) $< \mathcal{S}, \mathcal{A}, R, P, H, \mu >$
  - $\mathcal{S} \times \mathcal{A}$: state-action space with feature $\{\phi_h(s, a)\}$
  - R : reward function with mean r
  - P : transition kernel
  - H : planning horizon
  - μ : initial distribution

Problem Settings: RL with linear function approximation

- Linear MDP: unknown kernels $\{\theta_h\}_{h\in[H]}$ and $\{\mu_h\}_{h\in[H]}$
  - $r_h(s,a) = \langle \phi_h(s,a), \theta_h \rangle$
  - $P_h(\cdot \mid s,a) = \langle \phi_h(s,a), \mu_h(\cdot) \rangle$

Learning with L deployments

- Sampling phase:
    - For $\ell = 1,2,\ldots,L$
        - Compute and execute π^{ℓ} for a certain number of episodes
        - Collect the trajectories and update the whole dataset $\mathscr{D}$

- Planning Phase:
    - Receive reward kernel $\{\theta_h\}_{h\in[H]}$
    - Given $\{\theta_h\}_{h\in[H]}$ and $\mathscr{D}$, return a policy π such that $\mathbb{E}_{s_1\sim\mu}\left[V_1^{\pi}(s_1)\right] \geq \mathbb{E}_{s_1\sim\mu}\left[V_1^{*}(s_1)\right] - \epsilon$

$$V_h^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s\right], \quad V_h^{*}(s) = \max_{\pi} V_h^{\pi}(s)$$

Main Result

- **Theorem.** For reward-free exploration in linear MDPs, there is an algorithm (Algorithm 1) with deployment complexity H and sample complexity $\mathrm{poly}(d, H, 1/\epsilon, \log(1/\delta))$, such that with probability $1 - \delta$, for *all* linear reward functions, the algorithm returns a policy with suboptimality at most ϵ.

Pros:

Depolyment complexity H, where as the lower bound is $\tilde{\Omega}(H)$

Computational efficient algorithm

Does not require coverage assumption

Cons:

Sample complexity of $\tilde{O}\left(\frac{d^{15}H^{15}}{\epsilon^5}\right)$: bad dependencies on d, H and $\frac{1}{\epsilon}$

$\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ hides the log factors

High-level Intuitions

- Layer-by-layer approach
    - Construct the dataset from layer 1 to layer H
- Dealing with the infrequent directions with truncation
    - Rescale the infrequent feature ϕ such that $\mathbb{E}[\phi^\top \Lambda^{-1} \phi]$ is well bounded
- Independent copies to decouple the statistics

Λ: the information matrix

Future Directions

- Improving the polynomial dependencies
- Extending the results to general function approximation