Frozen Policy Iteration: Computationally Efficient RL under Linear Q^π Realizability for Deterministic Dynamics

Yijing Ke[1] * Zihan Zhang[2] Ruosong Wang[1]

[1]Peking University [2]HKUST

The Fourteenth International Conference on Learning Representations

*Presenting Author

The Setting: RL with Function Approximation

We study computationally and statistically efficient reinforcement learning with function approximation.

Two Structural Assumptions:

1. **Linear Q^π Realizability:**
   For any policy π, its action-value function is linear in a known feature map $\phi(s, a) \in \mathbb{R}^d$.

$$Q^\pi(s, a) = \langle \phi(s, a), \theta^\pi \rangle$$

2. **Deterministic Transitions:**
   Taking action a in state s leads to a unique next state s'.
   (Note: Initial states and rewards can still be highly stochastic.)

The Challenge: Distribution Shifts without a Simulator

Why is this hard?

- In standard policy iteration, updating the policy changes the state distribution.
- Historical data becomes **off-policy**. Evaluating a new policy using off-policy data under linear Q^π realizability introduces severe bias.

Limitations of Prior Works

Even under deterministic dynamics, existing statistically efficient algorithms fail in the online setting.

- **Computationally Intractable:** Methods that work online often rely on computationally intractable optimization problems or oracles.
- **Simulator Dependency:** Efficient algorithms require a **simulator** to resample states. In standard online RL, you cannot *revisit* a specific state at will if the initial state is stochastic.

Our Solution: Frozen Policy Iteration (FPI)

How do we stay *on-policy* without a simulator?

1. Strategic Data Collection

- When sampling a trajectory, we only trust the **high-confidence** segment.
- We find the *last* state on the trajectory that is not well-covered by our dataset, record its reward-to-go, and **discard all preceding data**.

2. Freezing the Policy

- Once a state is *well-explored* (all actions covered), we **freeze** its policy update for the rest of the learning process.
- **The Benefit:** This ensures that the downstream policy never changes, meaning historical trajectories remain effectively on-policy and their reward-to-go labels stay perfectly accurate!

Scaling to Regret Minimization

The FPI strategy easily yields a PAC bound, but bounding cumulative Regret requires continuous exploration.

FPI-Regret: Multi-Level Accuracy Framework

- Instead of one fixed threshold, we maintain **multiple accuracy levels** $l \in [1, \bar{L}]$, corresponding to precisions $\varepsilon = 2^{-l}$.
- **Dynamic Level Adaptation:** Start each episode at the highest accuracy, dropping to lower-accuracy levels to guide exploration.
- **Constrained Exploration:** Exploration is restricted to actions that are *near-optimal* according to the next-lower accuracy level, ensuring the suboptimality incurred during exploration is strictly controlled.

Main Theoretical Results

Theorem (Main Regret Bound)

With probability $1-\delta$, the regret of FPI-Regret is bounded by:

$$\mathrm{Reg}(T) = \widetilde{\mathcal{O}}\left(\sqrt{d^2H^6T} + \sqrt{d}H^2T\kappa\right)$$

Significance:

- **First** algorithm under linear Q^π realizability and deterministic transitions to achieve polynomial sample and computational complexity in standard Online RL.
- **Optimal for Bandits:** When $H=1$, our bound is $\widetilde{\mathcal{O}}(d\sqrt{T})$, matching the optimal lower bound for linear contextual bandits.