Reward-agnostic Fine-tuning: Provable Statistical Benefits of Hybrid Reinforcement Learning

Gen Li* Wenhao Zhan* Jason D. Lee
Yuejie Chi Yuxin Chen

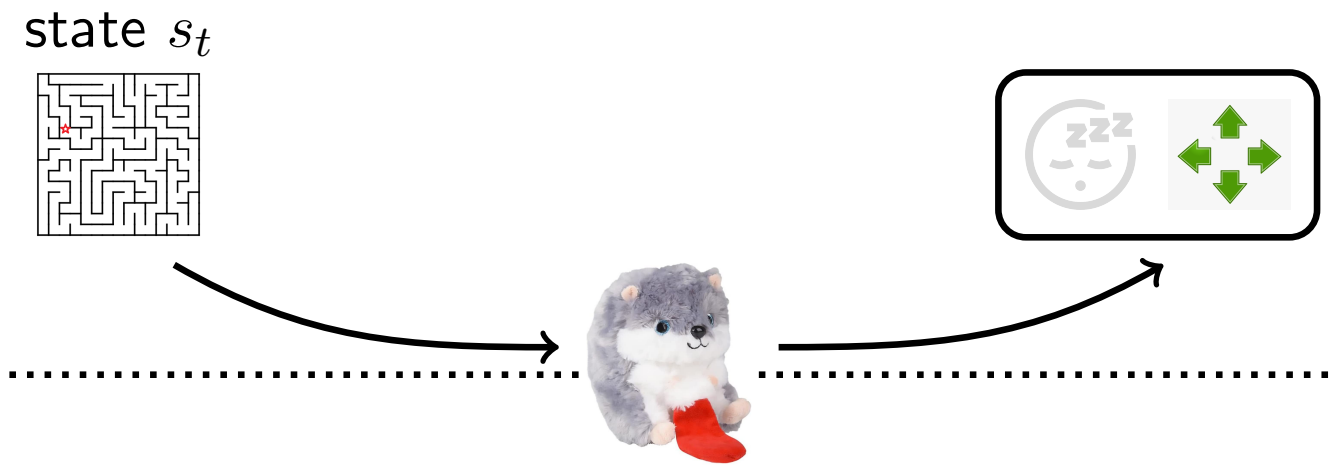
October 9, 2023

Figures borrowed from Yuxin Chen and Shicong Cen.

Recent successes in RL

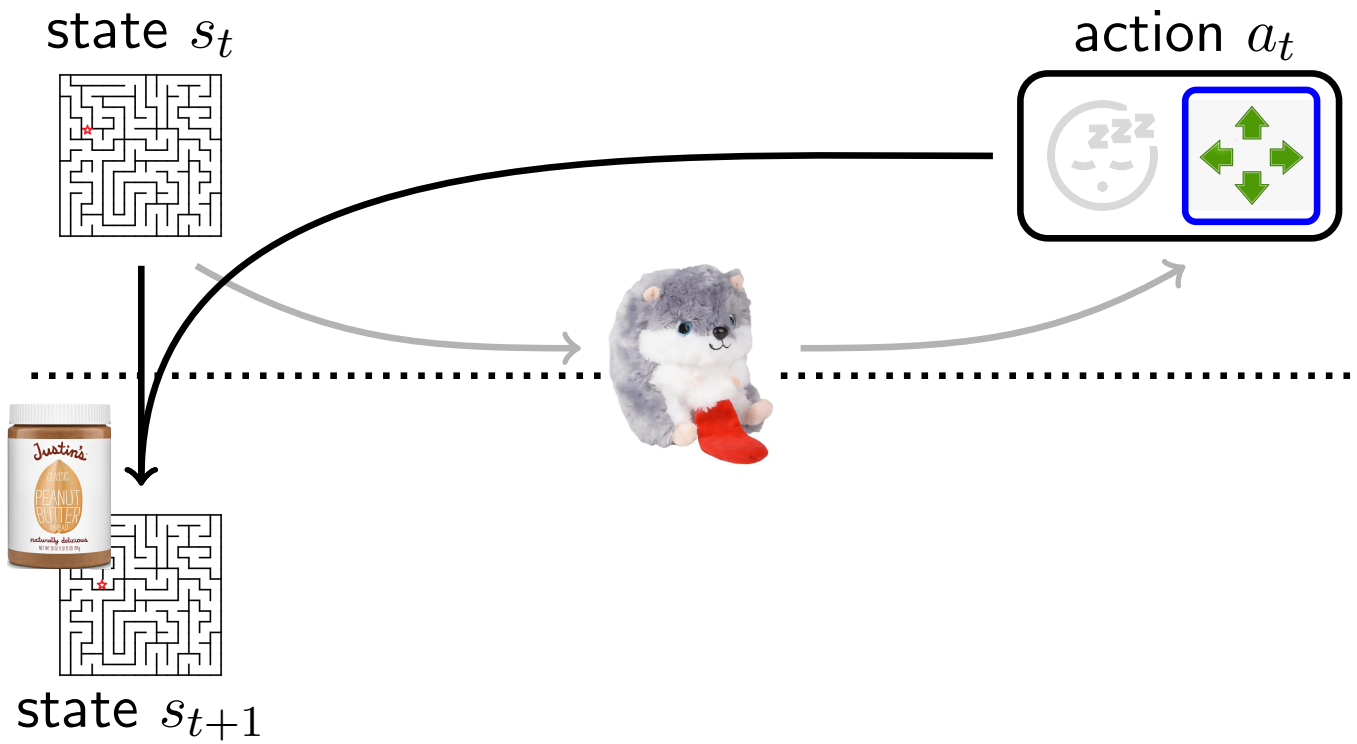
Markov decision process (MDP)

- A collection of MABs indexed by state $s \in \mathcal{S}$.
- At time step t, an agent observes the state s_t, selects an action $a_t \sim \pi(\cdot|s_t)$, and then receives a reward $r(s_t, a_t)$.
- The environment transitions to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$.



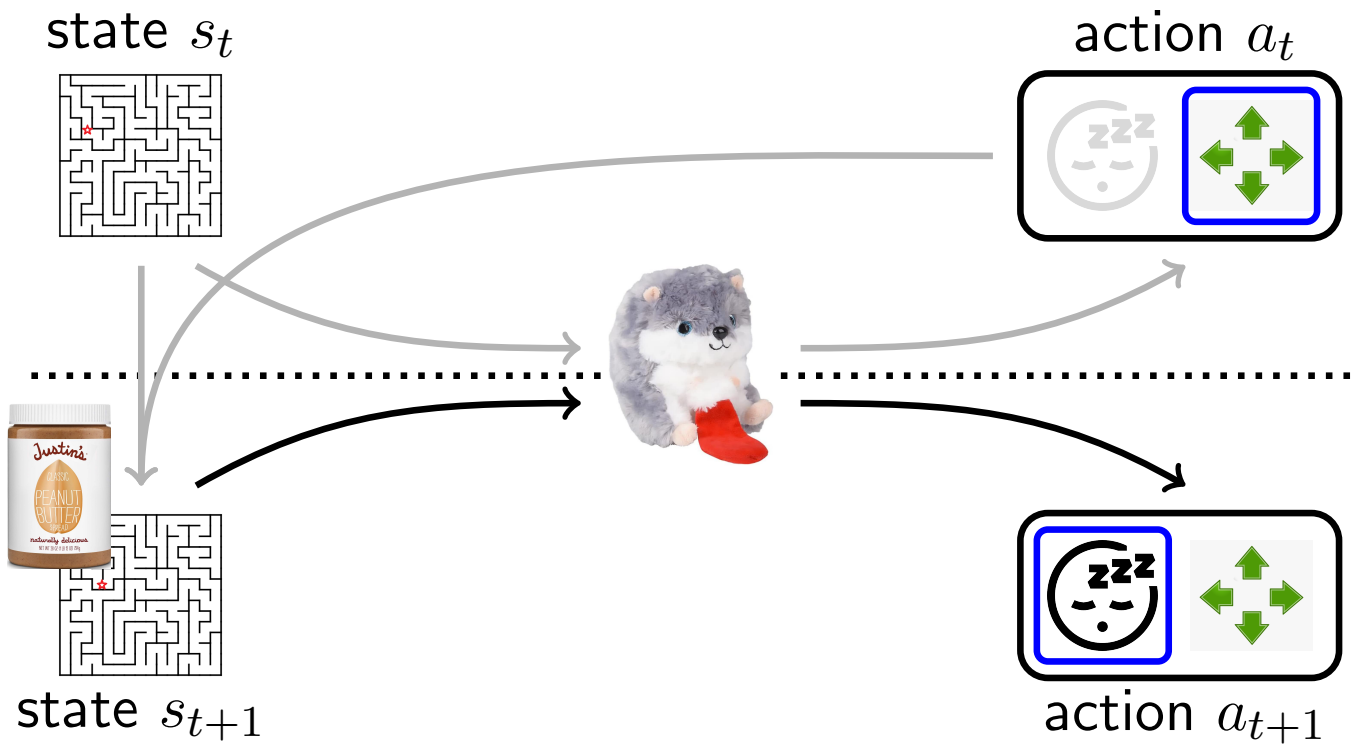
Markov decision process (MDP)

- A collection of MABs indexed by state $s \in \mathcal{S}$.
- At time step t, an agent observes the state s_t, selects an action $a_t \sim \pi(\cdot|s_t)$, and then receives a reward $r(s_t, a_t)$.
- The environment transitions to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$.



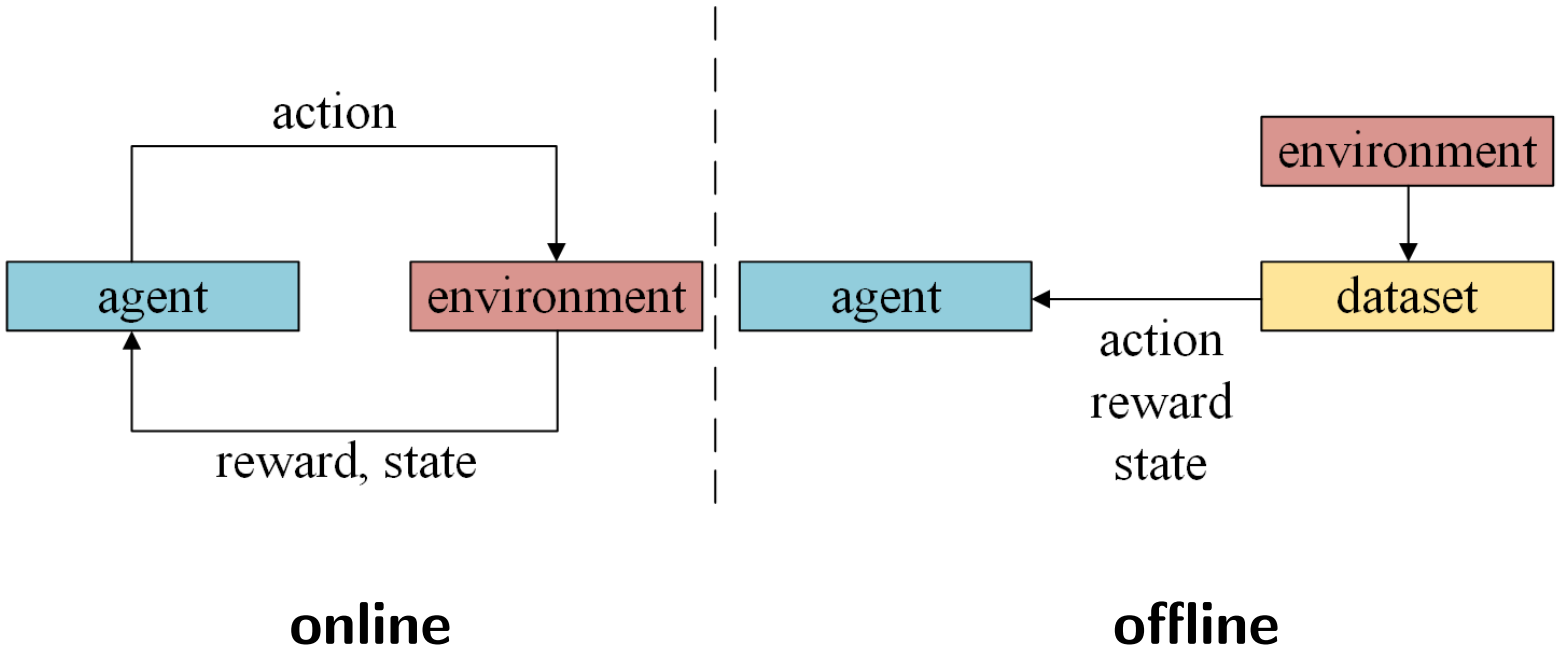
Markov decision process (MDP)

- A collection of MABs indexed by state $s \in \mathcal{S}$.
- At time step t, an agent observes the state s_t, selects an action $a_t \sim \pi(\cdot|s_t)$, and then receives a reward $r(s_t, a_t)$.
- The environment transitions to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$.



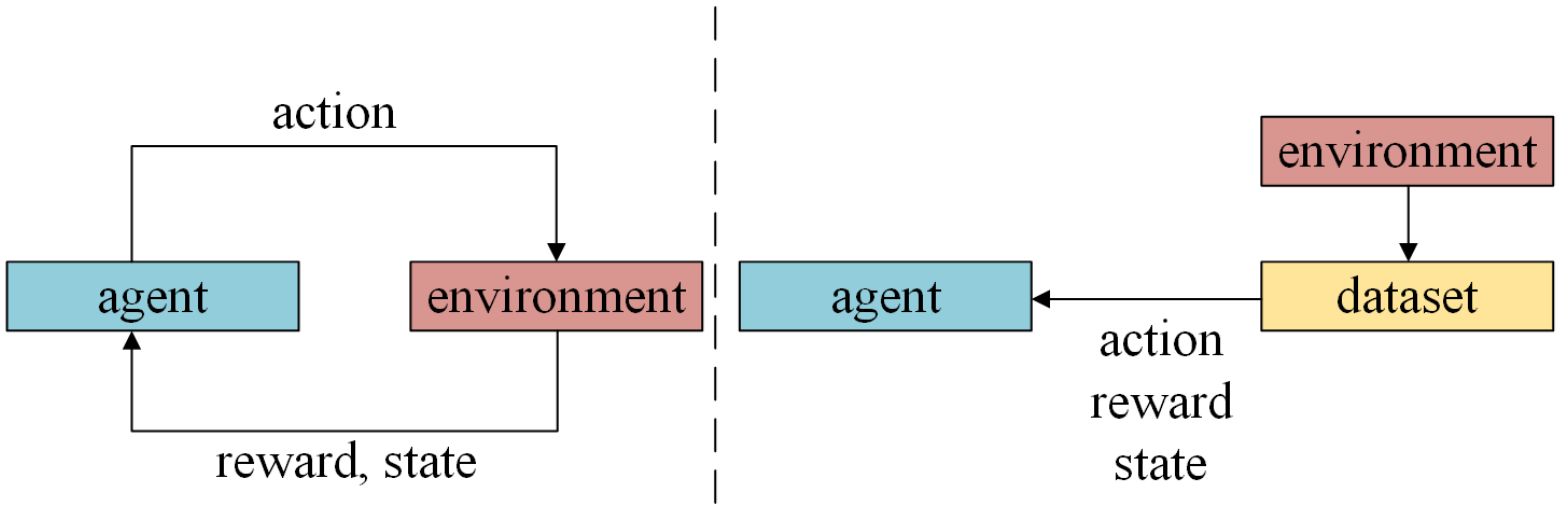
Reinforcement learning (RL)

Reinforcement Learning: **online vs offline**



Reinforcement learning (RL)

Reinforcement Learning: **online vs offline**



Both have limitations!

Online RL vs Offline RL

Limitations:

- Pure online RL: **overlooks all the information in the past data** and might be overly restrictive.
- Pure offline RL: the **concentrability requirement** might be too stringent and thus fragile.

Online RL vs Offline RL

Limitations:

- Pure online RL: **overlooks all the information in the past data** and might be overly restrictive.
- Pure offline RL: the **concentrability requirement** might be too stringent and thus fragile.

What if the agent has access to **an offline dataset**, while **(limited) online data collection** is also permitted? $\rightarrow$ **Hybrid RL**!

Online RL vs Offline RL

Limitations:

- Pure online RL: **overlooks all the information in the past data** and might be overly restrictive.
- Pure offline RL: the **concentrability requirement** might be too stringent and thus fragile.

What if the agent has access to **an offline dataset**, while **(limited) online data collection** is also permitted? $\rightarrow$ **Hybrid RL**!

Does hybrid RL allow for improved sample complexity compared to pure online or offline RL?

Setting

Episodic finite-horizon MDPs:

- MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, H, P = \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H, \rho\}$.
- $P_{h,s,a} := P_h(\cdot|s,a)$ is the transition probability.
- $r_h : \mathcal{S} \times \mathcal{A} \to [0,1]$ is the reward function.
- $\rho \in \Delta(\mathcal{S})$ is the initial distribution.

Setting

Policy and value function:

- $\pi = \{\pi_h\}_{h=1}^H$ where $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$ is the Markovian policy.
- value function $V_h^\pi(s) := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'}(s,a) \mid s_h = s\right]$
  $(V_h^\pi(\mu) := \mathbb{E}_{s\sim\mu}[V_h^\pi(s)])$.
- Q function $Q_h^\pi(s,a) := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_{h'}(s,a) \mid s_h = s, a_h = a\right]$.
- there exists an optimal deterministic policy $\pi^\star$ such that

$$V_h^\star(s) := \max_\pi V_h^\pi(s) = V_h^{\pi^\star}(s),$$

$$Q_h^\star(s,a) := \max_\pi Q_h^\pi(s,a) = Q_h^{\pi^\star}(s,a).$$

- occupancy distribution:
  $d_h^\pi(s,a) := \mathbb{P}\left(s_h = s, a_h = a \mid \pi\right), d_h^\pi(s) := \mathbb{P}\left(s_h = s \mid \pi\right)$

Sampling Mechanism

Hybrid RL: assumes access to a historical dataset as well as the ability to further explore the environment.

- **offline data**: $\mathcal{D}^{\mathsf{off}} = \{\tau^{k,\mathsf{off}}\}_{1 \le k \le K^{\mathsf{off}}}$ where each trajectory $\tau^{k,\mathsf{off}} = \left(s_h^{k,\mathsf{off}}, a_h^{k,\mathsf{off}}\right)_{h=1}^{H}$ is i.i.d. sampled from an **unknown** mixture of policies $\pi^{\mathsf{off}} = \mathbb{E}_{\pi \sim \mu^{\mathsf{off}}}[\pi]$
- we use $d_h^{\mathsf{off}}(s,a)$ to denote $\mathbb{P}\left(s_h^{k,\mathsf{off}} = s, a^{k,\mathsf{off}} = a\right)$.
- **online exploration**: the learner is able to sample K^{on} trajectories sequentially where the initial state is generated independently from ρ.
- total sample complexity: $K^{\mathsf{off}} + K^{\mathsf{on}}$.

Single-Policy Partial Concentrability

Single-policy concentrability requires the offline dataset to cover **all** the state-action pairs visited by the optimal policy $\rightarrow$ **too strong**!

Definition: Single-Policy Partial Concentrability

For any $\sigma \in [0,1]$, the single-policy partial concentrability coefficient $C^\star(\sigma)$ of the offline dataset $\mathcal{D}^{\mathsf{off}}$ is defined as

$$C^\star(\sigma) := \min \left\{ \max_{1 \le h \le H} \max_{(s,a) \in \mathcal{G}_h} \frac{d_h^{\pi^\star}(s,a)}{d_h^{\mathsf{off}}(s,a)} \;\middle|\; \{\mathcal{G}_h\}_{1 \le h \le H} \subseteq \mathcal{G}(\sigma) \right\},$$

where

$$\mathcal{G}(\sigma) := \left\{ \{\mathcal{G}_h\}_{1 \le h \le H} \subseteq \mathcal{S} \times \mathcal{A} \;\middle|\; \frac{1}{H} \sum_{h=1}^{H} \sum_{(s,a) \notin \mathcal{G}_h} d_h^{\pi^\star}(s,a) \le \sigma \right\}.$$

Single-Policy Partial Concentrability

- $C^\star(\sigma)$ allows a fraction of the state-action space reachable by $\pi^\star$ to be **insufficiently covered** ($\mathcal{G}(\sigma)$).
- $\mathcal{G}_h$ corresponds to a set of state-action pairs that undergo reasonable distribution shift while **the total occupancy density of the uncovered state-actions remain under control** ($\leq \sigma$).
- $C^\star(\sigma)$ is **non-increasing** in σ. When $\sigma = 0$, $C^\star(0)$ reduces to single-policy concentrability.

Single-Policy Partial Concentrability

- $C^\star(\sigma)$ allows a fraction of the state-action space reachable by $\pi^\star$ to be **insufficiently covered** ($\mathcal{G}(\sigma)$).
- $\mathcal{G}_h$ corresponds to a set of state-action pairs that undergo reasonable distribution shift while **the total occupancy density of the uncovered state-actions remain under control** ($\leq \sigma$).
- $C^\star(\sigma)$ is **non-increasing** in σ. When $\sigma = 0$, $C^\star(0)$ reduces to single-policy concentrability.

Single-policy partial concentrability is a reasonable generalization of single-policy concentrability!

Sample complexity of Pure Online/Offline RL

Minimax optimal sample complexity to learn an ε-optimal policy under single-policy partial concentrability:

- pure online RL: $\widetilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right)$.
- pure offline RL: $\widetilde{O}\left(\frac{H^3SC^\star(0)}{\varepsilon^2}\right)$
- hybrid RL: ?

Sample complexity of Pure Online/Offline RL

Minimax optimal sample complexity to learn an ε-optimal policy under single-policy partial concentrability:

- pure online RL: $\widetilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right)$.
- pure offline RL: $\widetilde{O}\left(\frac{H^3SC^\star(0)}{\varepsilon^2}\right)$
- hybrid RL: ?

We propose a **three-staged** algorithm to balance between offline data and online exploration.

Algorithm: Step 1

We divide the offline dataset $\mathcal{D}^{\mathsf{off}}$ into two halves $\mathcal{D}^{\mathsf{off},1}, \mathcal{D}^{\mathsf{off},2}$ and our online exploration consists of three parts, each collecting $K^{\mathsf{on}}_{\mathsf{prepare}} = K^{\mathsf{on}}_{\mathsf{imitate}} = K^{\mathsf{on}}_{\mathsf{explore}} = K^{\mathsf{on}}/3$ trajectories.

Step 1: estimation of the occupancy distributions.

- estimating d^{π} for any policy π: we invoke the estimation scheme developed in Li et al. (2023)*, which collects **a set of N sample trajectories for each step h** in order to facilitate estimation of the occupancy distributions.
- required samples: $K^{\mathsf{on}}_{\mathsf{prepare}} = NH$.

*Li, G., Yan, Y., Chen, Y., and Fan, J. (2023). Minimax-optimal reward-agnostic exploration in reinforcement learning.

Algorithm: Step 1

Step 1: estimation of the occupancy distributions.

- estimating d^{off}: we invoke the empirical estimate using the $K^{\mathsf{off}}/2$ sample trajectories from $\mathcal{D}^{\mathsf{off},1}$:

$$\widehat{d}_h^{\mathsf{off}}(s,a) = \frac{2N_h^{\mathsf{off}}(s,a)}{K^{\mathsf{off}}} \mathbb{1}\left(\frac{N_h^{\mathsf{off}}(s,a)}{K^{\mathsf{off}}} \geq c_{\mathsf{off}} \left\{ \frac{\log \frac{HSA}{\delta}}{K^{\mathsf{off}}} + \frac{H^4 S^4 A^4 \log \frac{HSA}{\delta}}{N} + \frac{SA}{K^{\mathsf{on}}} \right\} \right),$$

  where $N_h^{\mathsf{off}}(s,a) = \sum_{k=1}^{K^{\mathsf{off}}/2} \mathbb{1}\left(s_h^{k,\mathsf{off}} = s, a_h^{k,\mathsf{off}} = a\right)$,

- cutoff threshold: avoid the state-action pairs that are poorly covered $\rightarrow$ **Pessimism!**

Algorithm: Step 2

Step 2: online exploration.

Preliminary fact: if we have K independent trajectories sampled from d^{b}, then Li et al. (2023) is able to compute a policy $\widehat{\pi}$ satisfying $V^{\star}(\rho) - V^{\widehat{\pi}}(\rho) \lesssim H \left[\sum_h \sum_{s,a} \frac{d_h^{\pi^\star}(s,a)}{1/H + K d_h^{\mathsf{b}}(s,a)}\right]^{\frac{1}{2}}$.

- **imitiating the offline dataset**: if the offline dataset is collected by experts, it has rich information that we can utilize $\rightarrow$ we want to compute a mixture of policies that can cover d^{off} !

- we want to solve the following optimization problem:

$$\mu^{\mathsf{imitate}} \approx \arg \min_{\mu \in \Delta(\Pi)} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \max_{a \in \mathcal{A}} \frac{\widehat{d}_h^{\mathsf{off}}(s,a)}{\frac{1}{K^{\mathsf{on}} H} + \mathbb{E}_{\pi' \sim \mu}\left[\widehat{d}_h^{\pi'}(s,a)\right]}.$$

Algorithm: Step 2

Step 2: online exploration - imitating the offline dataset

- the above optimization problem is equivalent to

$$\mu^{\mathsf{imitate}} \approx \arg \min_{\mu \in \Delta(\Pi)} \max_{\pi : \mathcal{S} \times [H] \to \Delta(\mathcal{A})} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \mathbb{E}_{a \sim \pi_h(\cdot|s)} \left[\frac{\widehat{d}_h^{\mathsf{off}}(s,a)}{\frac{1}{K^{\mathsf{on}} H} + \mathbb{E}_{\pi' \sim \mu} \left[\widehat{d}_h^{\pi'}(s,a) \right]} \right].$$

- use Follow-The-Regularized-Leader (FTRL) to solve this optimization problem:

$$\pi_h^{t+1}(\cdot \,|\, s) \propto \exp\left(\eta \sum_{k=1}^{t} \frac{\widehat{d}_h^{\mathsf{off}}(s,\cdot)}{\frac{1}{K^{\mathsf{on}} H} + \mathbb{E}_{\pi' \sim \mu^k} \left[\widehat{d}_h^{\pi'}(s,\cdot) \right]} \right),$$

$$\mu^{t+1} \approx \arg \min_{\mu \in \Delta(\Pi)} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \mathbb{E}_{a \sim \pi_h^{t+1}(\cdot|s)} \left[\frac{\widehat{d}_h^{\mathsf{off}}(s,a)}{\frac{1}{K^{\mathsf{on}} H} + \mathbb{E}_{\pi' \sim \mu} \left[\widehat{d}_h^{\pi'}(s,a) \right]} \right],$$

for $t \in [T_{\max}]$. η is the learning rate.

Algorithm: Step 2

Step 2: online exploration - imitating the offline dataset

- output: $\pi^{\text{imitate}} = \mathbb{E}_{\pi \sim \mu^{\text{imitate}}}[\pi]$ with $\mu^{\text{imitate}} = \frac{1}{T_{\max}} \sum_{t=1}^{T_{\max}} \mu^t$.
- from the preliminary fact and the performance guarantee of FTRL, we know π^{imitate} can cover the offline dataset.
- how to compute μ^{t+1} in each iteration: we design a **computationally-efficient Frank-Wolfe-type algorithm** to solve it with **iteration complexity** $O\left(\frac{(K^{\text{on}}H)^4}{S^2}\right)$.

Algorithm: Step 2

Step 2: online exploration

- **exploring the unknown environment**: we also attempt to explore the environment in a way that complements the offline data.
- invoke the reward-agnostic online exploration scheme proposed in Li et al. (2023), which returns $\pi^{\text{explore}} = \mathbb{E}_{\pi \sim \mu^{\text{explore}}}[\pi]$.
- With the above two exploration policies π^{imitate} and π^{explore}, we execute the MDP to obtain sample trajectories as follows:
  1) Execute the MDP $K^{\text{on}}_{\text{imitate}}$ times with π^{imitate} to obtain a dataset containing $K^{\text{on}}_{\text{imitate}} = K^{\text{on}}/3$ independent sample trajectories, denoted by $\mathcal{D}^{\text{on}}_{\text{imitate}}$;
  2) Execute the MDP $K^{\text{on}}_{\text{explore}}$ times with policy π^{explore} to obtain a dataset containing $K^{\text{on}}_{\text{explore}} = K^{\text{on}}/3$ independent sample trajectories, denoted by $\mathcal{D}^{\text{on}}_{\text{explore}}$.

Algorithm: Step 3

Step 3: policy learning via offline RL

Once the above online exploration process is completed, we need to compute a near-optimal policy on the basis of the data in hand.

- Let us look at the following dataset

$$\mathcal{D} = \mathcal{D}^{\mathsf{off},2} \cup \mathcal{D}^{\mathsf{on}}_{\mathsf{imitate}} \cup \mathcal{D}^{\mathsf{on}}_{\mathsf{explore}}.$$

  To circumvent the complicated statistical dependency between $\mathcal{D}^{\mathsf{off},1}$ and $\mathcal{D}^{\mathsf{on}}_{\mathsf{imitate}} \cup \mathcal{D}^{\mathsf{on}}_{\mathsf{explore}}$, we only include the second half $\mathcal{D}^{\mathsf{off},2}$ of the offline dataset $\mathcal{D}^{\mathsf{off}}$ due to the fact that $\mathcal{D}^{\mathsf{off},2}$ is statistically independent from $\mathcal{D}^{\mathsf{on}}_{\mathsf{imitate}} \cup \mathcal{D}^{\mathsf{on}}_{\mathsf{explore}}$.

- We invoke the pessimistic model-based offline RL algorithm proposed in Li et al. (2023) to compute the final policy estimate $\widehat{\pi}$.

Full Algorithm

Algorithm 1: The proposed hybrid RL algorithm.

Input: offline dataset $\mathcal{D}^{\mathsf{off}}$ (containing K^{off} trajectories), parameters $N, K^{\mathsf{on}}, T_{\mathsf{max}}$, learning rate η.

Initialize: $\pi_h^1(a \,|\, s) = 1/A$ for any (s, a, h); $K = K^{\mathsf{off}} + K^{\mathsf{on}}$; split $\mathcal{D}^{\mathsf{off}}$ into two halves $\mathcal{D}^{\mathsf{off},1}$ and $\mathcal{D}^{\mathsf{off},2}$.

/* Estimation of occupancy distributions for any policy π. */

Call Algorithm 3, which allows one to specify $\widehat{d}_h^{\pi}(s, a)$ for any deterministic policy π and any (s, a, h).

/* Estimation of occupancy distributions of the historical data. */

Use the dataset $\mathcal{D}^{\mathsf{off},1}$ to compute

$$\widehat{d}_h^{\mathsf{off}}(s,a) = \frac{2N_h^{\mathsf{off}}(s,a)}{K^{\mathsf{off}}} \mathbb{1}\left(\frac{N_h^{\mathsf{off}}(s,a)}{K^{\mathsf{off}}} \geq c_{\mathsf{off}}\left\{\frac{\log\frac{HSA}{\delta}}{K^{\mathsf{off}}} + \frac{H^4S^4A^4\log\frac{HSA}{\delta}}{N} + \frac{SA}{K^{\mathsf{on}}}\right\}\right)$$

for any (s, a, h), where $N_h^{\mathsf{off}}(s,a) = \sum_{k=1}^{K^{\mathsf{off}}/2} \mathbb{1}(s_h^k = s, a_h^k = a)$ and $c_{\mathsf{off}} > 0$ is some absolute constant.

/* Compute a general sample-efficient online exploration scheme. */

Call Algorithm 5 with estimators $\widehat{d}^{\pi}$ to compute policy π^{explore} and the associated weight μ^{explore}.

/* Compute an online exploration scheme tailored to the offline dataset. */

for $t = 1, \cdots, T_{\mathsf{max}}$ **do**

  Compute μ^t using Algorithm 2.

  Update $\pi_h^{t+1}(a \,|\, s)$ for all $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ such that:

$$\pi_h^{t+1}(a \,|\, s) = \frac{\exp\left(\eta \sum_{k=1}^{t} \frac{\widehat{d}_h^{\mathsf{off}}(s,a)}{\frac{1}{K^{\mathsf{on}}H} + \mathbb{E}_{\pi' \sim \mu^k}\left[\widehat{d}_h^{\pi'}(s,a)\right]}\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\eta \sum_{k=1}^{t} \frac{\widehat{d}_h^{\mathsf{off}}(s,a')}{\frac{1}{K^{\mathsf{on}}H} + \mathbb{E}_{\pi' \sim \mu^k}\left[\widehat{d}_h^{\pi'}(s,a')\right]}\right)},$$

Set $\mu^{\mathsf{imitate}} = \frac{1}{T_{\mathsf{max}}} \sum_{t=1}^{T_{\mathsf{max}}} \mu^t$ and $\pi^{\mathsf{imitate}} = \mathbb{E}_{\pi \sim \mu^{\mathsf{imitate}}}[\pi]$.

/* Sampling using the above two exploration policies. */

Collect $K_{\mathsf{imitate}}^{\mathsf{on}}$ (resp. $K_{\mathsf{explore}}^{\mathsf{on}}$) sample trajectories using π^{imitate} (resp. π^{explore}) to form a dataset $\mathcal{D}_{\mathsf{imitate}}^{\mathsf{on}}$ (resp. $\mathcal{D}_{\mathsf{explore}}^{\mathsf{on}}$).

/* Run the model-based offline RL algorithm. */

Apply Algorithm 6 to the dataset $\mathcal{D} = \mathcal{D}^{\mathsf{off},2} \cup \mathcal{D}_{\mathsf{imitate}}^{\mathsf{on}} \cup \mathcal{D}_{\mathsf{explore}}^{\mathsf{on}}$ to compute a policy $\widehat{\pi}$.

Output: policy $\widehat{\pi}$.

Sample complexity

Theorem (Sample complexity of learning an ε-optimal policy)

Consider $\delta \in (0,1)$ and $\varepsilon \in (0, H]$. Choose the algorithmic parameters such that

$$\eta = \sqrt{\frac{\log A}{2T_{\mathsf{max}}(K^{\mathsf{on}}H)^2}} \qquad \text{and} \qquad T_{\mathsf{max}} \geq 2(K^{\mathsf{on}}H)^2 \log A.$$

Suppose that

$$K^{\mathsf{on}} + K^{\mathsf{off}} \geq c_1 \frac{H^3 S C^\star(\sigma)}{\varepsilon^2} \log^2 \frac{K}{\delta}$$
$$K^{\mathsf{on}} \geq c_1 \frac{H^3 S A \min\{H\sigma, 1\}}{\varepsilon^2} \log \frac{K}{\delta}$$

for some large enough constant $c_1 > 0$. Then with probability at least $1-\delta$, the policy $\widehat{\pi}$ returned by Algorithm 1 satisfies

$$V_1^\star(\rho) - V^{\widehat{\pi}}(\rho) \leq \varepsilon,$$

provided that K^{on} and K^{off} both exceed some polynomial $\mathsf{poly}(H, S, A, C^\star(\sigma), \log \frac{K}{\delta})$ *(independent of ε).*

Sample Complexity: Discussion

- In a nutshell, our algorithm yields ε-accuracy as long as

$$K^{\mathsf{on}} + K^{\mathsf{off}} \gtrsim \frac{H^3 SC^\star(\sigma)}{\varepsilon^2} \log^2 \frac{K}{\delta},$$
$$K^{\mathsf{on}} \gtrsim \frac{H^3 SA \min\{H\sigma, 1\}}{\varepsilon^2} \log \frac{K}{\delta},$$

- let $K^{\mathsf{off}} = K^{\mathsf{on}} = K/2$ and then the sample complexity bound simplifies to

$$\widetilde{O}\left(\min_{\sigma\in[0,1]} \left\{ \frac{H^3 SA \min\{H\sigma, 1\}}{\varepsilon^2} + \frac{H^3 SC^\star(\sigma)}{\varepsilon^2} \right\}\right) =: \widetilde{O}\left(\min_{\sigma\in[0,1]} f_{\mathsf{mixed}}(\sigma)\right) \tag{1}$$

Sample Complexity: Comparison with Pure Online RL

- now look at pure online RL, corresponding to the case where $K = K^{\mathsf{on}}$ (so that all sample episodes are collected via online exploration). In this case, the minimax-optimal sample complexity for computing an ε-optimal policy is known to be

$$\widetilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right) = \widetilde{O}\big(f_{\mathsf{mixed}}(1)\big) \tag{2}$$

- The sample complexity of pure online RL (2) is clearly **worse than** hyrid RL (1).
- For instance, if there exists some very small $\sigma \ll 1/H$ obeying $C^\star(\sigma) \lesssim 1$, then the ratio of (1) to (2) is at most $H\sigma + \frac{1}{A} \ll 1$.

Sample Complexity: Comparison with Pure Online RL

- now look at pure online RL, corresponding to the case where $K = K^{\text{on}}$ (so that all sample episodes are collected via online exploration). In this case, the minimax-optimal sample complexity for computing an ε-optimal policy is known to be

$$\widetilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right) = \widetilde{O}\big(f_{\text{mixed}}(1)\big) \tag{2}$$

- The sample complexity of pure online RL (2) is clearly **worse than** hyrid RL (1).
- For instance, if there exists some very small $\sigma \ll 1/H$ obeying $C^\star(\sigma) \lesssim 1$, then the ratio of (1) to (2) is at most $H\sigma + \frac{1}{A} \ll 1$.

Hybrid RL improves the sample complexity with respect to pure online RL!

Sample Complexity: Comparison with Pure Offline RL

- In the pure offline case where $K = K^{\text{off}}$, the minimax sample complexity is known to be

$$\widetilde{O}\left(\frac{H^3 S C^\star(0)}{\varepsilon^2}\right) = \widetilde{O}\big(f_{\text{mixed}}(0)\big) \tag{3}$$

  for any target accuracy level ε, which is apparently larger than (1) in general.

- In particular, recognizing that $C^\star(0) = \infty$ in the presence of incomplete coverage of the state-action space reachable by $\pi^\star$, we might harvest enormous sample size benefits.

Sample Complexity: Comparison with Pure Offline RL

- In the pure offline case where $K = K^{\text{off}}$, the minimax sample complexity is known to be

$$\widetilde{O}\left(\frac{H^3 S C^\star(0)}{\varepsilon^2}\right) = \widetilde{O}\big(f_{\text{mixed}}(0)\big) \tag{3}$$

  for any target accuracy level ε, which is apparently larger than (1) in general.

- In particular, recognizing that $C^\star(0) = \infty$ in the presence of incomplete coverage of the state-action space reachable by $\pi^\star$, we might harvest enormous sample size benefits.

Hybrid RL also improves the sample complexity with respect to pure offline RL!

Discussion about the algorithm

In addition to the sample complexity advantages, the proposed hybrid RL enjoys several attributes that could be practically appealing:

- **Adaptivity to unknown optimal σ :** our algorithm does not rely on any knowledge of σ and automatically identifies the optimal σ that minimizes the function $f_{\text{mixed}}(\sigma)$.

Discussion about the algorithm

In addition to the sample complexity advantages, the proposed hybrid RL enjoys several attributes that could be practically appealing:

- **Adaptivity to unknown optimal σ :** our algorithm does not rely on any knowledge of σ and automatically identifies the optimal σ that minimizes the function $f_{\text{mixed}}(\sigma)$.

Our algorithm can **automatically identify the optimal trade-offs** between distribution mismatch and inadequate coverage!

Discussion about the algorithm

- **Reward-agnostic data collection:** the online exploration procedure employed in our algorithm does not require any prior information about the reward function. The reward function is only queried at the last step to output the learned policy.

Discussion about the algorithm

- **Reward-agnostic data collection:** the online exploration procedure employed in our algorithm does not require any prior information about the reward function. The reward function is only queried at the last step to output the learned policy.

This enables us to perform hybrid RL in a **reward-agnostic** manner!

Discussion about the algorithm

- **Strengthening behavior cloning:** our algorithm does not rely on prior knowledge about π^{off} and is capable of finding a mixed exploration policy π^{imitate} that inherits the advantages of the unknown behavior policy.

Discussion about the algorithm

- **Strengthening behavior cloning:** our algorithm does not rely on prior knowledge about π^{off} and is capable of finding a mixed exploration policy π^{imitate} that inherits the advantages of the unknown behavior policy.
- In behavior cloning where the offline dataset $\mathcal{D}^{\text{off}}$ is generated by an expert policy, with $C^\star = C^\star(0) \approx 1$, the supplement of online data collection improves behavior cloning by **lowering the statistical error from** $\sqrt{\frac{H^3SC^\star}{K_{\text{off}}}}$ **to** $\sqrt{\frac{H^3SC^\star}{K_{\text{off}}+K_{\text{on}}}}$, together with an executable learned policy π^{imitate}.

Discussion about the algorithm

- **Strengthening behavior cloning:** our algorithm does not rely on prior knowledge about π^{off} and is capable of finding a mixed exploration policy π^{imitate} that inherits the advantages of the unknown behavior policy.
- In behavior cloning where the offline dataset $\mathcal{D}^{\text{off}}$ is generated by an expert policy, with $C^\star = C^\star(0) \approx 1$, the supplement of online data collection improves behavior cloning by **lowering the statistical error from** $\sqrt{\frac{H^3SC^\star}{K_{\text{off}}}}$ **to** $\sqrt{\frac{H^3SC^\star}{K_{\text{off}}+K_{\text{on}}}}$, together with an executable learned policy π^{imitate}.

Our algorithm provides a method to **strengthen behavior cloning**!

Conclusion

Takeaway: hyrid RL can indeed achieves better sample complexity compared against pure online and pure offline RL. The key is to balance between the offline data and online explorations.