On the complexity of learning good policies with and without rewards

Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent and Michal Valko

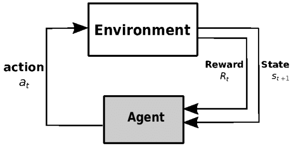




Mathematics of Online Decision Making, Simons Institute
October 28th, 2020

Many RL problems

RL setup: an agent interacts with an environement (MDP)

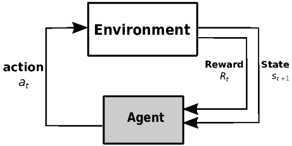


Several Performance measures:

1. the agent should *adopt* a good behavior
   - ➜ maximize the total rewards (*regret minimization*)
   - ➜ use as much as possible an ε-optimal policy (*PAC-MDP*)
2. the agent should *learn* a good behavior
   - ➜ learn an optimal policy for a given reward function
   - ➜ learn the dynamics so that to be robust to find the optimal policy for *any* reward function

Many RL problems

RL setup: an agent interacts with an environement (MDP)



Several Performance measures:

1. the agent should *adopt* a good behavior
   - ➜ maximize the total rewards (*regret minimization*)
   - ➜ use as much as possible an ε-optimal policy (*PAC-MDP*)
2. the agent should *learn* a good behavior
   - ➜ learn an optimal policy for a given reward function
   - ➜ learn the dynamics so that to be robust to find the optimal policy for *any* reward function

two Pure Exploration problems

Setting: Episodic Markov Decision Process

Episodic MDP: horizon H and MDP $(\mathcal{S}, \mathcal{A}, P, r)$ for

- a state space $\mathcal{S}$ of size $S < \infty$
- an action space $\mathcal{A}$ of size $A < \infty$
- a transition kernel $P = (p_h(s'|s,a))_{\substack{(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S} \\ h \in [H]}}$
- a reward function $r = (r_h(s,a))_{\substack{(s,a) \in \mathcal{S}\times\mathcal{A} \\ h \in [H]}}$

Value of a policy $\pi = (\pi_h)_{h=1}^H$, $\pi_h : \mathcal{S} \to \mathcal{A}$:

$$V_h^\pi(s; r) \triangleq \mathbb{E}^\pi\left[\sum_{\ell=h}^H r_\ell(s_\ell, \pi_\ell(s_\ell)) \middle| \substack{s_h = s \\ s_{\ell+1} \sim p_\ell(\cdot|s_\ell, \pi_\ell(s_\ell))}\right]$$

Optimal policy: $\pi_r^\star$ such that $V_h^{\pi_r^\star}(s; r) \geq V_h^\pi(s; r)$ for all π, s, h.

Setting: Episodic Markov Decision Process

Episodic MDP: horizon H and MDP $(\mathcal{S}, \mathcal{A}, P, r)$ for

- a state space $\mathcal{S}$ of size $S < \infty$
- an action space $\mathcal{A}$ of size $A < \infty$
- a transition kernel $P = (p_h(s'|s,a))_{\substack{(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S} \\ h\in[H]}}$
- a reward function $r = (r_h(s,a))_{\substack{(s,a)\in\mathcal{S}\times\mathcal{A} \\ h\in[H]}}$

Value of a policy $\pi = (\pi_h)_{h=1}^H$, $\pi_h : \mathcal{S} \to \mathcal{A}$:

$$V_h^\pi(s; r) \triangleq \mathbb{E}^\pi\left[\sum_{\ell=h}^{H} r_\ell(s_\ell, \pi_\ell(s_\ell)) \middle| \substack{s_h = s \\ s_{\ell+1}\sim p_\ell(\cdot|s_\ell,\pi_\ell(s_\ell))}\right]$$

Optimal policy: $\pi_r^\star$ such that $V_h^{\pi_r^\star}(s; r) \geq V_h^\pi(s; r)$ for all π, s, h.

Setting: Episodic Markov Decision Process

Episodic MDP: horizon H and MDP $(\mathcal{S}, \mathcal{A}, P, r)$ for

- a state space $\mathcal{S}$ of size $S < \infty$
- an action space $\mathcal{A}$ of size $A < \infty$
- a transition kernel $P = (p_h(s'|s,a))_{\substack{(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S} \\ h \in [H]}}$
- a reward function $r = (r_h(s,a))_{\substack{(s,a) \in \mathcal{S}\times\mathcal{A} \\ h \in [H]}}$

Q-value of a policy $\pi = (\pi_h)_{h=1}^H$, $\pi_h : \mathcal{S} \to \mathcal{A}$:

$$Q_h^\pi(s,a;r) \triangleq \mathbb{E}^\pi\left[r_h(s,a) + \sum_{\ell=h+1}^{H} r_\ell(s_\ell, \pi_\ell(s_\ell)) \middle| \substack{s_h = s, a_h = a \\ s_{\ell+1} \sim p_\ell(\cdot|s_\ell, \pi_\ell(s_\ell))} \right]$$

Optimal policy: $\pi_r^\star$ such that $V_h^{\pi_r^\star}(s;r) \geq V_h^\pi(s;r)$ for all π, s, h.

Setting: Episodic Markov Decision Process

Episodic MDP: horizon H and MDP $(\mathcal{S}, \mathcal{A}, P, r)$ for

- a state space $\mathcal{S}$ of size $S < \infty$
- an action space $\mathcal{A}$ of size $A < \infty$
- a transition kernel $P = (p_{\boldsymbol{h}}(s'|s,a))_{\substack{(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S} \\ h \in [H]}}$
- a reward function $r = (r_{\boldsymbol{h}}(s,a))_{\substack{(s,a) \in \mathcal{S}\times\mathcal{A} \\ h \in [H]}}$ **(step-dependent)**

Q-value of a policy $\pi = (\pi_h)_{h=1}^H$, $\pi_h : \mathcal{S} \to \mathcal{A}$:

$$Q_h^\pi(s,a;r) \triangleq \mathbb{E}^\pi\left[r_h(s,a) + \sum_{\ell=h+1}^H r_\ell(s_\ell, \pi_\ell(s_\ell)) \,\middle|\, \substack{s_h = s, a_h = a \\ s_{\ell+1} \sim p_\ell(\cdot|s_\ell, \pi_\ell(s_\ell))} \right]$$

Optimal policy: $\pi_r^\star$ such that $V_h^{\pi_r^\star}(s;r) \geq V_h^\pi(s;r)$ for all π, s, h.

Outline

Online episodic algorithm

Collect data from the MDP by generating trajectories (episodes)
$\neq$ generative model

In each episode $t = 1, 2, \ldots,$ the agent

- selects an exploration policy π^t
- generates an episode under this policy

$$(s_1^t, a_1^t, s_2^t, a_2^t, \ldots, s_H^t, a_H^t)$$

  where $s_1^t \sim \rho$, $a_h^t = \pi_h^t(s_h^t)$ and $s_{h+1}^t \sim p_h(\cdot|s_h^t, a_h^t)$
- can decide to stop exploration
- if decides to stop, outputs a prediction

➜ three data-dependent components

Online episodic algorithm

Collect data from the MDP by generating trajectories (episodes)
$\neq$ generative model

In each episode $t = 1, 2, \dots,$ the agent

- selects an exploration policy π^t
- generates an episode under this policy

$$(s_1^t, a_1^t, s_2^t, a_2^t, \dots, s_H^t, a_H^t)$$

  where $\boldsymbol{s_1^t} = \boldsymbol{s}_1$, $a_h^t = \pi_h^t(s_h^t)$ and $s_{h+1}^t \sim p_h(\cdot|s_h^t, a_h^t)$
- can decide to stop exploration
- if decides to stop, outputs a prediction

➜ three data-dependent components

Best Policy Identification (BPI)

➜ Learn the optimal policy for a known reward function r

[Fiechter, 1994]

BPI algorithm

- exploration policy π^t: may dependent on past data $\mathcal{D}_{t-1}$ and r

$$\mathcal{D}_t = \mathcal{D}_{t-1} \cup \left\{ \left(s_1^t, a_1^t, s_2^t, a_2^t, \dots, s_H^t, a_H^t \right) \right\}$$

- stopping rule τ : stopping time w.r.t. $(\mathcal{D}_t)_{t\in\mathbb{N}}$ (can depend on r)
- prediction $\hat{\pi}$: a **policy** that may depend on $\mathcal{D}_\tau$ and r

(ε, δ)-PAC algorithm for Best Policy Identification

$$\mathbb{P}\left(V_1^\star(s_1; r) - V_1^{\hat{\pi}}(s_1; r) \le \varepsilon \right) \ge 1 - \delta$$

Wanted: (ε, δ)-PAC algorithm with a small sample complexity τ

Reward-Free Exploration (RFE)

➜ Learn the optimal policy for **any** reward function r

[Jin et al., 2020]

RFE algorithm

- exploration policy π^t: may dependent on past data $\mathcal{D}_{t-1}$

$$\mathcal{D}_t = \mathcal{D}_{t-1} \cup \left\{(s_1^t, a_1^t, s_2^t, a_2^t, \dots, s_H^t, a_H^t)\right\}$$

- stopping rule τ : stopping time w.r.t. $(\mathcal{D}_t)_{t\in\mathbb{N}}$
- prediction $\hat{P} = (\hat{p}_h(\cdot|s,a))_{h,s,a}$: a **transition kernel** that may depend on $\mathcal{D}_\tau$

$\hat{\pi}_r^\star$: optimal policy in the MDP $(\hat{P}, r)$

(ε, δ)-PAC algorithm for Reward-Free Exploration

$$\mathbb{P}\left(\text{for all reward function } r, V_1^\star(s_1; r) - V_1^{\hat{\pi}_r^\star}(s_1; r) \le \varepsilon\right) \ge 1 - \delta$$

Wanted: (ε, δ)-PAC algorithm with a small sample complexity τ

Outline

A model-based algorithm

Based on the available data $\mathcal{D}_t$, builds estimates of the transition probabilities $p_h(s,a)$

➜ estimates of the Q-values $Q_h^{\pi}(s,a;r)$

Number of visits:

$$n_h^t(s,a) = \sum_{k=1}^{t} \mathbb{1}_{\{(s_h^k,a_h^k)=(s,a)\}} \quad n_h^t(s,a,s') = \sum_{k=1}^{t} \mathbb{1}_{\{(s_h^k,a_h^k,s_{h+1}^k)=(s,a,s')\}}$$

Empirical transitions: $\hat{P}^t = (\hat{p}_h^t(s'|s,a))_{h,s,a,s'}$

$$\hat{p}_h^t(s'|s,a) = \begin{cases} \frac{n_h^t(s,a,s')}{n_h^t(s,a)} & \text{if } n_h^t(s,a) > 0 \\ \frac{1}{S} & \text{else} \end{cases}$$

Empirical values:

- $\hat{V}_h^{t,\pi}(s;r)$ values in the empirical MDP $(\mathcal{S},\mathcal{A},\hat{P}^t,r)$
- $\hat{Q}_h^{t,\pi}(s;r)$ Q-values in the empirical MDP $(\mathcal{S},\mathcal{A},\hat{P}^t,r)$

Reward-Free UCRL

Central observation

A sufficient condition to be (ε, δ)-PAC is to have accurate estimates of the value function for all π and r:

$$\mathbb{P}\left(\forall\pi, \forall r,\ |\hat{V}_1^{\tau,\pi}(s_1; r) - V_1^{\pi}(s_1; r)| \leq \varepsilon/2\right) \geq 1 - \delta.$$

RF-UCRL:

- builds upper bounds on the errors

$$\hat{e}_h^{t,\pi}(s, a; r) := |\hat{Q}_h^{t,\pi}(s, a; r) - Q_h^{\pi}(s, a; r)|$$

... that are independent of π and r!

- greedily reduces the upper bounds

Reward-Free UCRL

$$\hat{e}_h^{t,\pi}(s,a;r) := |\hat{Q}_h^{t,\pi}(s,a;r) - Q_h^{\pi}(s,a;r)|$$

We define inductively $\bar{E}_{H+1}^t(s,a) = 0$ and

$$\bar{E}_h^t(s,a) = \min\left[(H-h);\ (H-h)\sqrt{\frac{2\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)}} + \sum_{s'} \hat{p}_h^t(s'|s,a) \max_b \bar{E}_{h+1}^t(s',b)\right]$$

for some treshold function $\beta(n,\delta)$.

➜ like in UCRL [Jaksch et al. 10], this construction relies on confidence regions on the transitions probabilities

Upper Bound Property

On the event

$$\mathcal{E} = \left\{\forall t \in \mathbb{N}, \forall h \in [H], \forall (s,a), \mathrm{KL}\left(\hat{p}_h^t(\cdot|s,a), p_h(\cdot|s,a)\right) \le \frac{\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)}\right\},$$

for all π and r, for all h, s, a, $\hat{e}_h^{t,\pi}(s,a;r) \le \bar{E}_h^t(s,a)$.

Proof

$$\hat{e}_h^{t,\pi}(s,a;r) := \left|\hat{Q}_h^{t,\pi}(s,a;r) - Q_h^{\pi}(s,a;r)\right|$$

A simple consequence of Bellman equations:

$$\begin{aligned}
\hat{Q}_h^{t,\pi}(s,a;r) &= r_h(s,a) + \sum_{s'} \hat{p}_h^t(s'|s,a)\hat{Q}_{h+1}^{t,\pi}(s',\pi(s');r) \\
\text{and} \quad Q_h^{\pi}(s,a;r) &= r_h(s,a) + \sum_{s'} p_h(s'|s,a)Q_{h+1}^{\pi}(s',\pi(s');r)\,.
\end{aligned}$$

Error decomposition:

$$\begin{aligned}
\hat{e}_h^{t,\pi}(s,a;r) &\le \sum_{s'} \left|\hat{p}_h^t(s'|s,a) - p_h(s'|s,a)\right| Q_{h+1}^{\pi}(s',\pi(s');r) \\
&\quad + \sum_{s'} \hat{p}_h^t(s'|s,a)\left|\hat{Q}_{h+1}^{t,\pi}(s',\pi(s');r) - Q_{h+1}^{\pi}(s',\pi(s');r)\right| \\
&\le (H-h)\underbrace{\left\|\hat{p}_h^t(\cdot|s,a) - p_h(\cdot|s,a)\right\|_1}_{\le\sqrt{\frac{2\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)}}\ (\text{Pinsker}+\mathcal{E})} + \sum_{s'} \hat{p}_h^t(s'|s,a)\underbrace{\hat{e}_{h+1}^{t,\pi}(s',\pi(s');r)}_{\substack{\le \max_b \overline{E}_{h+1}^t(s',b) \\ \text{(induction)}}}\,.
\end{aligned}$$

The algorithm

$$\bar{E}_h^t(s,a) = \min\left[(H-h);\ (H-h)\sqrt{\frac{2\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)}} + \sum_{s'} \hat{p}_h^t(s'|s,a)\max_b \bar{E}_{h+1}^t(s',b)\right]$$

Reward-Free UCRL

- **exploration policy**: π^{t+1} is the greedy policy wrt $\bar{E}^t(s,a)$:

$$\forall s \in \mathcal{S}, \forall h \in [h], \quad \pi_h^{t+1}(s) = \underset{a\in\mathcal{A}}{\arg\max}\ \bar{E}_h^t(s,a).$$

- **stopping rule**: $\tau = \inf\left\{t \in \mathbb{N} : \bar{E}_1^t(s_1, \pi_1^{t+1}(s_1)) \leq \varepsilon/2\right\}$
- **prediction**: transition kernel $\hat{P}^\tau$

➜ very close to an old algorithm by [Fiechter, 1994]
... originally proposed for Best Policy Identification!

Theoretical guarantees

Theorem [Kaufmann et al. 2020]

With $\beta(n,\delta) \simeq \log\left(\frac{1}{\delta}\right) + S\log(n)$, RF-UCRL is (ε,δ)-PAC for Reward-Free Exploration and satisfies, w.p. $1-\delta$,

$$\tau^{\text{RF-UCRL}} = \tilde{\mathcal{O}}\left(\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]\right)$$

➜ improves over the state-of-the art bound of [Jin et al. 20]

$$\tau^{\text{RF-RL-Explore}} = \tilde{\mathcal{O}}\left(\frac{S^2 A H^5}{\varepsilon^2}\log\left(\frac{1}{\delta}\right) + \frac{S^4 A H^7}{\varepsilon^2}\log^3\left(\frac{1}{\delta}\right)\right)$$

➜ RF-UCRL is a natural *adaptive* approach to RFE
... with a simple sample complexity analysis

Outline

Beyond RF-UCRL

First observation: RF-UCRL is also (ε, δ)-PAC for Best Policy Identification with the updated

- **prediction rule**: $\hat{\pi}$, the optimal policy if the MDP $(\mathcal{S}, \mathcal{A}, \hat{P}^{\tau}, r)$

$$\tau^{\text{RF-UCRL}} = \tilde{\mathcal{O}}\left(\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]\right) \quad \text{w.h.p.}$$

Lower bound for BPI [Domingues et al. 2020]

For every (ε, δ)-PAC BPI algorithm, there exists an MDP (with stage-dependent transitions) such that

$$\mathbb{E}[\tau] \geq c_1 \frac{H^3 SA}{\varepsilon^2} \log\left(\frac{1}{\delta}\right),$$

where c_1 is an absolute constant.

➜ some room for improvement...

Building on Regret Minimizing algorithm

The UCB-VI algorithm of [Azar et al. 17] satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(V_1^{\star}(s_1;r) - V_1^{\pi^t}(s_1;r)\right)\right] \leq C\left(\sqrt{H^3SAT}\right)$$

(minimax optimal cumulative regret)

From UCB-VI to a BPI algorithm [Jin et al. 18]

- **exploration policy:** that of the UCB-VI algorithm
- **stopping rule:** $T = \frac{C^2SAH^3}{\varepsilon^2\delta^2}$
  ($\tau = T$ is fixed in advance)
- **prediction rule:** $\hat{\pi}$ is one of the policies used by UCB-VI, chosen uniformly at random: $\hat{\pi} = \hat{\pi}^{\boldsymbol{n}}$ $\boldsymbol{n} \sim \mathcal{U}(\{1,\ldots,T\})$

Building on Regret Minimizing algorithm

The UCB-VI algorithm of [Azar et al. 17] satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(V_1^{\star}(s_1; r) - V_1^{\pi^t}(s_1; r)\right)\right] \leq C\left(\sqrt{H^3 SAT}\right)$$

(minimax optimal cumulative regret)

From UCB-VI to a BPI algorithm [Jin et al. 18]

- **exploration policy:** that of the UCB-VI algorithm
- **stopping rule:** $T = \frac{C^2 SAH^3}{\varepsilon^2 \delta^2}$
  ($\tau = T$ is fixed in advance)
- **prediction rule:** $\hat{\pi}$ is one of the policies used by UCB-VI, chosen uniformly at random: $\hat{\pi} = \hat{\pi}^{\boldsymbol{n}} \quad \boldsymbol{n} \sim \mathcal{U}(\{1, \ldots, T\})$

➜ optimal dependency in ε, sub-optimal dependency in δ

A more *adaptive* conversion from a regret minimizer:

→ associate a data-dependent stopping rule to a UCRL algorithm

BPI-UCRL

- **exploration policy**: $\pi^{t+1}(s) = \arg\max_{a\in\mathcal{A}} \overline{Q}_h^t(s,a;r)$
- **stopping rule**: $\tau = \inf\left\{t \in \mathbb{N} : \overline{V}_1^t(s_1;r) - \underline{V}_1^t(s_1;r) \leq \epsilon\right\}$
- **prediction rule**: $\hat{\pi}_h(s) = \arg\max_{a\in\mathcal{A}} \underline{Q}_h^\tau(s,a;r)$

where we have built upper and lower confidence bounds

$$\begin{aligned} \underline{Q}_h^t(s,a;r) &\leq Q_h^\star(s,a;r) &\leq \overline{Q}_h^t(s,a;r) \\ \underline{V}_h^t(s;r) &\leq V_h^\star(s;r) &\leq \overline{V}_h^t(s;r). \end{aligned}$$

An alternative: BPI-UCRL

A more *adaptive* conversion from a regret minimizer:

➜ associate a data-dependent stopping rule to a UCRL algorithm

BPI-UCRL

- **exploration policy**: $\pi^{t+1}(s) = \arg\max_{a \in \mathcal{A}} \overline{Q}_h^t(s, a; r)$
- **stopping rule**: $\tau = \inf\left\{t \in \mathbb{N} : \overline{V}_1^t(s_1; r) - \underline{V}_1^t(s_1; r) \leq \epsilon\right\}$
- **prediction rule**: $\hat{\pi}_h(s) = \arg\max_{a \in \mathcal{A}} \underline{Q}_h^\tau(s, a; r)$

where we have built upper and lower confidence bounds

$$\begin{aligned} \underline{Q}_h^t(s, a; r) &\leq Q_h^\star(s, a; r) &\leq \overline{Q}_h^t(s, a; r) \\ \underline{V}_h^t(s; r) &\leq V_h^\star(s; r) &\leq \overline{V}_h^t(s; r). \end{aligned}$$

Sample complexity of BPI-UCRL

Theorem [Kaufmann et al. 2020]

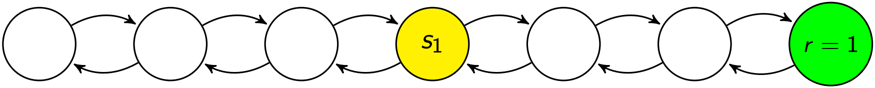
With $\beta(n, \delta) \simeq \log\left(\frac{1}{\delta}\right) + S\log(n)$, BPI-UCRL is (ε, δ)-PAC for Best Policy Identification and satisfies, w.p. $\geq 1 - \delta$,

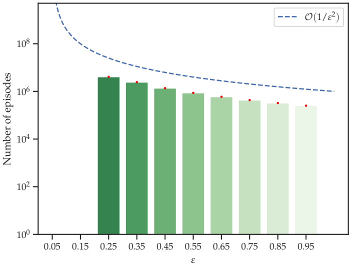
$$\tau^{\text{BPI-UCRL}} = \tilde{\mathcal{O}}\left(\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]\right)$$

- ➜ similar sample complexity bound as RF-UCRL (obtained with a similar proof)
- ➜ yet the practical story is different...

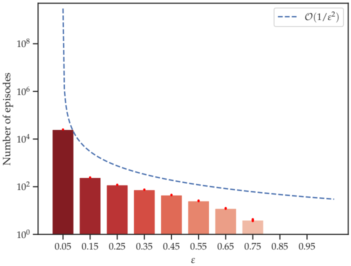
RF-UCRL versus BPI-UCRL

Double Chain MDP with $L = 31, H = 20$:





$\mathbb{E}[\tau|\tau < 10^8]$ for RF-UCRL



$\mathbb{E}[\tau|\tau < 10^6]$ for BPI-UCRL

➜ BPI-UCRL has a much smaller sample complexity!

Summary

- The sample complexity of...

	Upper Bound	Lower Bound
BPI	$\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ BPI-UCRL / RF-UCRL	$\frac{H^3 SA}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)$ [Darwiche Domingues et al. 2020]
RFE	$\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ RF-UCRL	$\frac{H^3 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ + [Jin et al. 2020]

Follow-up work: shaving the remaining H factor for BPI and RFE for more sophisticated algorithms using Bernstein bonuses

➜ BPI-UCBVI for Best Policy Identification

➜ RF-Express for Reward Free Exploration

Ménard et al. 2020, *Fast active learning for pure exploration in reinforcement learning*, arXiv:2007.13442

Summary

- The sample complexity of...

	Upper Bound	Lower Bound
BPI	$\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ BPI-UCRL / RF-UCRL	$\frac{H^3 SA}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)$ [Darwiche Domingues et al. 2020]
RFE	$\frac{H^4 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ RF-UCRL	$\frac{H^3 SA}{\varepsilon^2}\left[\log\left(\frac{1}{\delta}\right) + S\right]$ + [Jin et al. 2020]

Future work: beyond worst-case guarantees

➜ problem-dependent sample complexity for the simpler *planning* problem (= find the best first action) [Jonsson et al., 2020]

➜ problem-dependent regret guarantees [Simchowitz and Jamieson, 2019]

... how about BPI?

References

- Azar et al., *Minimax Regret Bounds for Reinforcement Learning*, ICML 2017
- Darwiche Domingues et al., *Episodic Reinforcement Learning in Finite MDPs: Minimax Lower Bounds Revisited* , arXiv:2010.03531, 2020
- Jaksch et al., *Near-optimal Regret Bounds for Reinforcement Learning*, JMLR 2010
- Jin et al., *Reward-Free Exploration for Reinforcement Learning*, ICML 2020
- Jin et al., *Is Q-Learning Provably Efficient?*, NeurIPS 2018
- Jonsson et al., *Planning in Markov Decision Processes with Gap-Dependent Sample Complexity*, NeurIPS 2020
- Fiechter, *Efficient Reinforcement Learning*, COLT 1994
- Kaufmann et al., *Adaptive Reward-Free Exploration*, arXiv:2006.06294, 2020
- Ménard et al., *Fast active learning for pure exploration in reinforcement learning*, arXiv:2007.13442, 2020
- Simchowitz and Jamieson, *Non-Asymptotic Gap-Dependent Regret Bounds for Tabular MDPs*, NeurIPS 2019