Foundations of Reinforcement Learning

Sample-efficient RL under linear MDP and realizability assumptions

Yuejie Chi

Department of Electrical and Computer Engineering

Carnegie Mellon University

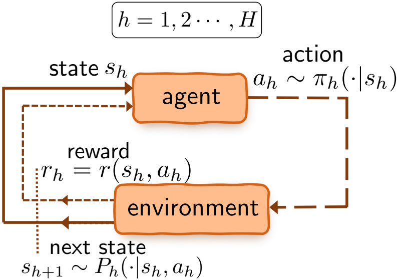
Spring 2023

Outline

Sample-efficient RL in linear MDP

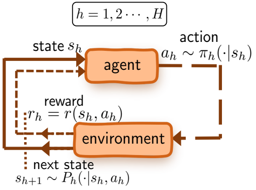
Sample-efficient RL under realizability

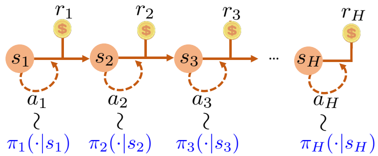
Recap: finite-horizon episodic MDP



- H: horizon length
- $\mathcal{S}$: state space with size S
- $\mathcal{A}$: action space with size A
- $r_h(s_h, a_h) \in [0, 1]$: immediate reward in step h
- $\pi = \{\pi_h\}_{h=1}^{H}$: policy (or action selection rule)
- $P_h(\cdot \mid s, a)$: transition probabilities in step h

Value function and Q-function



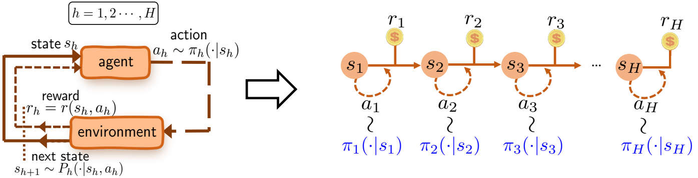


$$V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^{H} r_t(s_t, a_t) \,\middle|\, s_h = s\right]$$

$$Q_h^\pi(s, a) := \mathbb{E}\left[\sum_{t=h}^{H} r_t(s_t, a_t) \,\middle|\, s_h = s, a_h = a\right]$$

- execute policy π to generate sample trajectory

Bellman's optimality eq. for finite-horizon MDPs



Let $Q_h^\star(s,a) = \max_\pi Q_h^\pi(s,a)$ and $V_h^\star(s) = \max_\pi V_h^\pi(s)$.

1. Begin with the terminal step $h = H + 1$:

$$V_{H+1}^\star = 0, \quad Q_{H+1}^\star = 0.$$

2. Backtrack $h = H, H-1, \ldots, 1$:

$$Q_h^\star(s,a) := \underbrace{\mathbb{E}\left[r_h(s_h, a_h)\right]}_{\text{immediate reward}} + \underbrace{\mathbb{E}_{s' \sim P_h(\cdot|s,a)} V_{h+1}^\star(s')}_{\text{next step's value}}$$

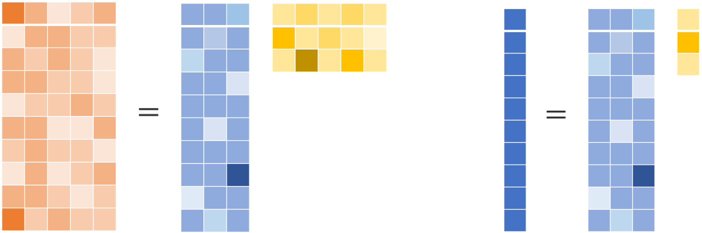
$$V_h^\star(s) := \max_{a \in \mathcal{A}} Q_h^\star(s,a), \qquad \pi_h^\star(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q_h^\star(s,a).$$

Sample-efficient RL in linear MDP

Linear MDP

Linear MDP: the transition kernel $P_h(s'|s,a)$ and the reward $r_h(s,a)$ can be decomposed by

$$P_h(s'|s,a) = \langle \phi(s,a),\, \mu_h^\star(s') \rangle \qquad\qquad r_h(s,a) = \langle \phi(s,a),\, \theta_h^\star \rangle$$



where

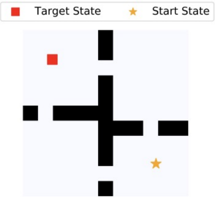
$$\mu_h^\star : \mathcal{S} \mapsto \mathbb{R}^d \quad \text{and} \quad \theta_h^\star \in \mathbb{R}^d.$$

Feature map in linear MDP

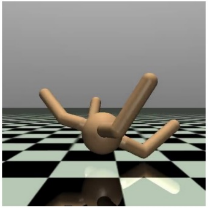
We assume the feature map $\phi(s,a)$ is known, and

$$\sup_{s,a} \|\phi(s,a)\|_2 \leq 1.$$

- *Tabular MDP:* pick $\phi(s,a)$ as one-hot vector for each (s,a) pair.
- *Soft state aggregation* [Singh et al., 1994]: think of $\mu_h^\star$ and $\theta_h^\star$ as hidden/latent states.
- *Learned features*, e.g. via contrastive learning [Zhang et al., 2022]:



Four Rooms



Mojoco

DM Control

Nice implications of linear MDP

- For any policy π,

$$\begin{aligned} Q_h^\pi(s,a) &= r_h(s,a) + P_h(\cdot|s,a)V_{h+1}^\pi \\ &= \langle \theta_h^\star, \phi(s,a)\rangle + \langle V_{h+1}^\pi, \mu_h^\star \phi(s,a)\rangle \\ &= \langle \underbrace{\theta_h^\star + (\mu_h^\star)^\top V_{h+1}^\pi}_{=: w_h^\pi}, \phi(s,a)\rangle \end{aligned}$$

is also linear in $\phi(s,a)$! Here, we overload the notation $\mu_h^\star \in \mathbb{R}^{|\mathcal{S}|\times d}$.

- Closedness under the Bellman operator: for any f_{h+1} linear in ϕ,

$$\begin{aligned} (\mathcal{T} f_{h+1})(s,a) &:= r_h(s,a) + \mathbb{E}_{s'\sim P_h(\cdot|s,a)}[\max_{a'} f_{h+1}(s',a')] \\ &= \langle \theta_h^\star, \phi(s,a)\rangle + \langle \max_{a'} f_{h+1}(s',a'), \mu_h^\star \phi(s,a)\rangle \\ &= \left\langle \theta_h^\star + (\mu_h^\star)^\top \max_{a'} f_{h+1}(s',a'),\ \phi(s,a) \right\rangle \end{aligned}$$

is linear in ϕ.

Planning in linear MDP

1. Begin with the terminal step $h = H + 1$:

$$V^\star_{H+1} = 0, \quad Q^\star_{H+1} = 0.$$

2. Backtrack $h = H, H-1, \ldots, 1$:

$$\begin{aligned} Q^\star_h(s,a) &= r_h(s,a) + P_h(\cdot|s,a)V^\star_{h+1} \\ &= \langle \theta^\star_h, \phi(s,a)\rangle + \langle V^\star_{h+1}, \mu^\star_h \phi(s,a)\rangle \\ &= \langle \underbrace{\theta^\star_h + (\mu^\star_h)^\top V^\star_{h+1}}_{=: w^\star_h}, \phi(s,a)\rangle \end{aligned}$$

Therefore, $Q^\star_h(s,a)$ is also linear in $\phi(s,a)$!

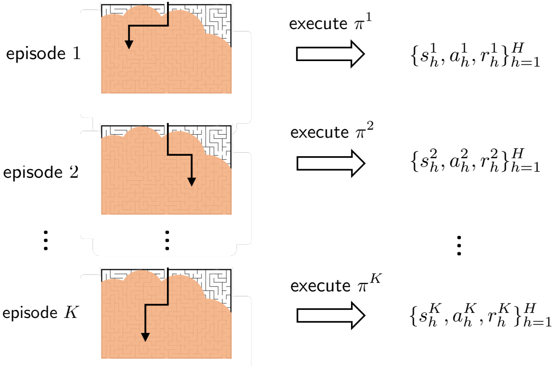
3. Update

$$V^\star_h(s) = \max_a Q^\star_h(s,a)$$

Online RL with linear MDP

Sequentially execute MDP for K episodes, each consisting of H steps

— *sample size:* $T = KH$



How to balance exploration and exploitation in linear MDP?

Recall: UCB-VI

For each episode k:

1. Backtrack $h = H, H-1, \ldots, 1$: run **optimistic value iteration**

$$Q_h(s,a) \leftarrow \min\left\{H-h+1, \quad \underbrace{r_h(s_h,a_h)}_{\text{immediate reward}} \; + \; \underbrace{\widehat{P}_{h,s,a}V_{h+1}}_{\text{next step's value}} \; + \underbrace{b_h(s_h,a_h)}_{\text{bonus}}\right\},$$

$$V_h(s) \leftarrow \max_{a \in \mathcal{A}} Q_h(s,a),$$

2. Forward $h = 1, \ldots, H$: take action according to the greedy policy

$$\pi_h(s) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} Q_h(s,a)$$

and collect $\{s_h, a_h, r_h\}_{h=1}^{H}$.

Can we extend UCB-VI to linear MDP?

Key challenges:

- How do we estimate the model $\widehat{P}_{h,s,a}$?

  — *For simplicity, assume r is known.*

- How do we design the bonus term $b_h(s_h, a_h)$?

Step 1: learning the model

Model learning in linear MDP

Given the transitions

$$\mathcal{D}_h^n = \{s_h^i, a_h^i, s_{h+1}^i\}_{i=0}^{n-1},$$

how to learn $\mu_h^\star$?

- Define the S-dimensional one-hot vector

$$\delta(s_{h+1}^i) = [0, \ldots, 1, \ldots, 0]^\top$$

  then

$$\mathbb{E}[\delta(s_{h+1}^i) | \mathcal{H}_h^i] = P_h(\cdot | s_h^i, a_h^i) = \mu_h^\star \phi(s_h^i, a_h^i),$$

  where $\mathcal{H}_h^i$ is the history information up to the collected transition.

- Treat $\delta(s_{h+1}^i)$ as a regression target for $\mu_h^\star \phi(s_h^i, a_h^i)$.

Model learning via ridge regression

$$\widehat{\mu}_h^n = \arg\min_{\mu} \underbrace{\sum_{i=0}^{n-1} \|\mu_h \phi(s_h^i, a_h^i) - \delta(s_{h+1}^i)\|_2^2}_{\text{data fitting}} + \underbrace{\lambda \|\mu\|_{\mathsf{F}}^2}_{\text{regularization}}$$

- Closed-form solution:

$$\widehat{\mu}_h^n = \Big(\underbrace{\sum_{i=0}^{n-1} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I}_{=:\Lambda_h^n} \Big)^{-1} \Big(\sum_{i=0}^{n-1} \delta(s_{h+1}^i)\phi(s_h^i, a_h^i) \Big)$$

- For value iteration, we only need to compute, for any V,

$$\begin{aligned} \widehat{P}_{h,s,a} V &= (\widehat{\mu}_h^n \phi(s,a))^\top V \\ &= \phi(s,a)^\top (\Lambda_h^n)^{-1} \sum_{i=0}^{n-1} \phi(s_h^i, a_h^i) V(s_{h+1}^i), \end{aligned}$$

which admits an efficient computation.

Step 2: design the bonus

Bonus design in linear MDP

How do we quantify the uncertainty of

$$\left\|\left(\widehat{P}_{h,s,a} - P_{h,s,a}\right)V\right\|_\infty ?$$

- **Prediction error on $\mu_h^\star$:**

$$\widehat{\mu}_h^n - \mu_h^\star = -\lambda \mu_h^\star (\Lambda_h^n)^{-1} + \sum_{i=1}^{n-1} \epsilon_h^i \phi(s_h^i, a_h^i)^\top (\Lambda_h^n)^{-1},$$

where $\epsilon_h^i = \delta(s_{h+1}^i) - P_h(\cdot | s_h^i, a_h^i)$.

- **Prediction error on $P_{h,s,a}V$:**

$$\begin{aligned}\left(\widehat{P}_{h,s,a} - P_{h,s,a}\right)V &= \phi(s,a)^\top (\widehat{\mu}_h^n - \mu_h^\star)^\top V \\ &= -\lambda \phi(s,a)^\top (\Lambda_h^n)^{-1} \mu_h^{\star\top} V + \underbrace{\sum_{i=1}^{n-1} \phi(s,a)^\top (\Lambda_h^n)^{-1} \phi(s_h^i, a_h^i) \epsilon_h^{i\top} V}_{\text{self-normalized bounds for Martingales}}\end{aligned}$$

Bonus design

$$\left|\left(\widehat{P}_{h,s,a} - P_{h,s,a}\right)V\right| \lesssim H\sqrt{d}\|\phi(s,a)\|_{(\Lambda_h^n)^{-1}}$$

- For a fixed V, use self-normalized bounds for Martingales [Abbasi-Yadkori et al., 2011].
- Covering argument to obtain uniform convergence.

The algorithm: LSVI-UCB

For each episode $k = 1, 2, \dots, K$,

1. Collect a trajectory $\{(s_h^k, a_h^k, r_h^k)\}_{h=1}^H$ according to the greedy policy π^k w.r.t. $\widehat{Q}_h^k$.
2. For $h = H, H-1, \dots, 1$:
    1. Define $\Lambda_h^k = \lambda I + \sum_{i=1}^k \phi_h^i (\phi_h^i)^\top$, where $\phi_h^i = \phi(s_h^i, a_h^i)$.
    2. Let $\widetilde{Q}_h^k$ be the estimate from ridge regression:
    $$\widetilde{Q}_h^k(s,a) = \phi(s,a)^\top (\Lambda_h^k)^{-1} \sum_{i=1}^k \phi_h^i \big(r_h^i + \widehat{V}_{h+1}(s_{h+1}^i) \big)$$
    3. Add bonus to ensure optimism:
    $$\widehat{Q}_h^k(s,a) = \widetilde{Q}_h^k(s,a) + \beta \sqrt{\phi(s,a)^\top (\Lambda_h^k)^{-1} \phi(s,a)}$$
    4. Obtain the value estimate:
    $$\widehat{V}_h^k(s) := \min \big\{ H, \max_a \widehat{Q}_h^k(s,a) \big\}.$$

Theory of LSVI-UCB

Given K initial states $\{s_1^k\}_{1\le k\le K}$ chosen by nature, define

$$\text{Regret}(K) := \sum_{k=1}^{K} \left(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \right)$$

Theorem 1 ([Jin et al., 2020])

LSVI-UCB achieves (up to log factor)

$$\frac{1}{K}\text{Regret}(K) \lesssim \sqrt{\frac{d^3 H^3}{T}}$$

where T is sample size.

- Sublinear regret $O(\sqrt{T})$.
- The regret depends on the dimension of the feature space d, rather than the ambient dimension SA.

Sample-efficient RL under realizability

Realizability assumption

Linear $Q^\star$ (Realizability) assumption: $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

$$\forall (s,a,h): \qquad Q_h^\star(s,a) = \langle \varphi_h(s,a),\, \theta_h^\star \rangle$$

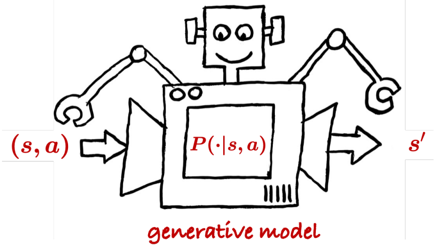
$\implies$ only $Q_h^\star = r_h + P_h V_{h+1}^\star$ is linearly realizable

Arguably the weakest linear function approximation assumption.

Can we hope to achieve sample efficiency in linear $Q^\star$ problem?

Case 1: RL with a generative model / simulator

Can query arbitrary state-action pairs to get samples



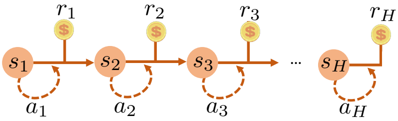
- In general, needs $\min\left\{e^{\Omega(d)}, e^{\Omega(H)}\right\}$ samples [Weisz et al., 2021]
- With constant sub-optimality gap, needs only $\mathsf{poly}\left(d, H, \frac{1}{\Delta_{\mathsf{gap}}}\right)$ samples [Du et al., 2020].

$$\Delta_{\mathsf{gap}} := \min_{\substack{s,h \\ a:\ \text{suboptimal action}}} \left\{V_h^\star(s) - Q_h^\star(s, a)\right\}$$

Case 2: online RL

Obtain data samples via sequential interaction with environment

- collect N episodes of data, each consisting of H steps
- in the n-th episode, execute MDP using a policy π^n

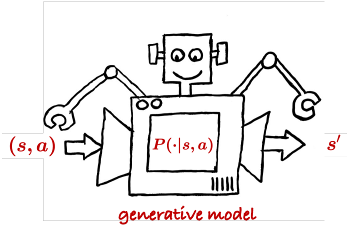


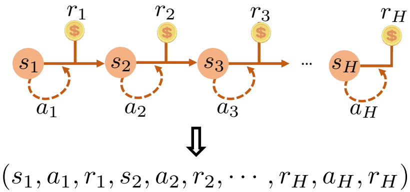
$$\Downarrow$$

$$(s_1, a_1, r_1, s_2, a_2, r_2, \cdots, r_H, a_H, r_H)$$

Needs $\min\left\{e^{\Omega(d)}, e^{\Omega(H)}\right\}$ samples when $\Delta_{\text{gap}} \asymp 1$ [Wang et al., 2021]

	generative model	online RL
no sub-optimality gap	inefficient	inefficient
with sub-optimality gap	efficient	inefficient





generative model: idealistic

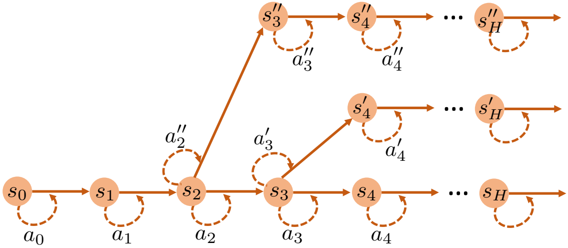
online RL: more restrictive/practical

Is there a sampling mechanism — more flexible than standard online RL, yet practically relevant — that still promises efficient learning?

A new sampling protocol: state revisiting

Allow one to revisit previous states in the same episode

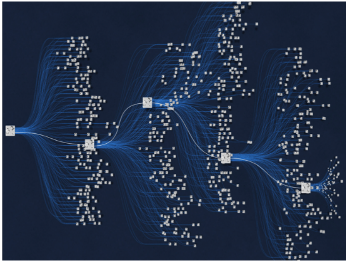
— *also called local access to generative model [Yin et al., 2022]*



- **Input:** initial state (chosen by nature)
- generate a length-H trajectory
- Pick any previously visited state s_h in this episode, and repeat

A new sampling protocol: state revisiting

“save files” feature in video games



Monte Carlo Tree Search

- more flexible than standard online RL
- more restrictive/practical than generative model

Issue: $\sharp$ revisit attempts might affect sample size

Theory

Theorem 2 ([Li et al., 2021])

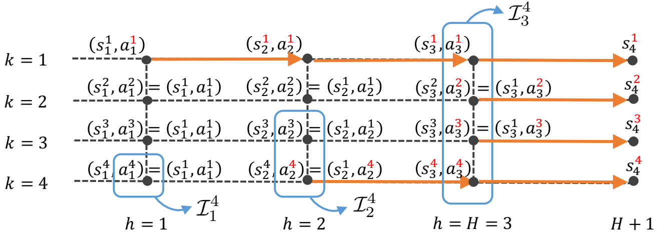
There exists an algorithm that achieves (up to log factor)

$$\frac{1}{K}\text{Regret}(K) \lesssim \sqrt{\frac{d^2 H^7}{T}}$$

where T is sample size, and $\sharp$ state revisits is at most $\widetilde{O}\big(\frac{d^2 H^5}{\Delta_{\text{gap}}^2}\big)$.

- Sample size needed to get ε average regret: $\text{poly}\big(d, H, \frac{1}{\Delta_{\text{gap}}}, \frac{1}{\varepsilon}\big)$, independent of S and A
- Limited state revisits: $\text{poly}(d, H, \frac{1}{\Delta_{\text{gap}}})$, almost independent of ε
- Can be easily refined to get logarithmic regret bound (in T)

A glimpse of the algorithm: LinQ-LSVI-UCB



Key ingredients:

- Adapted from LSVI-UCB [Jin et al., 2020]
- Check exploration bonus: if this uncertainty term exceeds $\Delta_{\text{gap}}/2$, then revisit states to draw more samples

References I

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011).
Improved algorithms for linear stochastic bandits.
Advances in neural information processing systems, 24.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2020).
Is a good representation sufficient for sample efficient reinforcement learning?
In *International Conference on Learning Representations*.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020).
Provably efficient reinforcement learning with linear function approximation.
In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Li, G., Chen, Y., Chi, Y., Gu, Y., and Wei, Y. (2021).
Sample-efficient reinforcement learning is feasible for linearly realizable MDPs with limited revisiting.
Advances in Neural Information Processing Systems, 34:16671–16685.

Singh, S., Jaakkola, T., and Jordan, M. (1994).
Reinforcement learning with soft state aggregation.
Advances in neural information processing systems, 7.

Wang, Y., Wang, R., and Kakade, S. (2021).
An exponential lower bound for linearly realizable MDP with constant suboptimality gap.
Advances in Neural Information Processing Systems, 34:9521–9533.

References II

Weisz, G., Amortila, P., and Szepesvári, C. (2021).
Exponential lower bounds for planning in MDPs with linearly-realizable optimal action-value functions.
In *Algorithmic Learning Theory*, pages 1237–1264. PMLR.

Yin, D., Hao, B., Abbasi-Yadkori, Y., Lazić, N., and Szepesvári, C. (2022).
Efficient local planning with linear function approximation.
In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR.

Zhang, T., Ren, T., Yang, M., Gonzalez, J., Schuurmans, D., and Dai, B. (2022).
Making linear MDPs practical via contrastive representation learning.
In *International Conference on Machine Learning*, pages 26447–26466. PMLR.