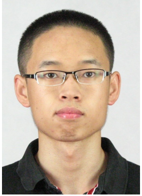
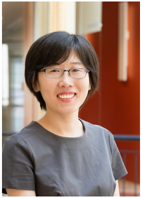
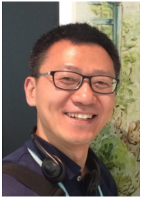
Sample-Efficient Reinforcement Learning Is Feasible for Linearly Realizable MDPs with Limited Revisiting




Gen Li
Princeton ECE

Yuxin Chen
Princeton ECE

Yuejie Chi
CMU ECE

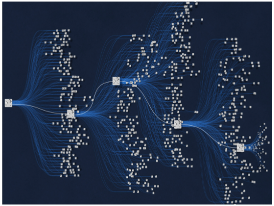
Yuantao Gu
Tsinghua EE

Yuting Wei
UPenn Stats

Reinforcement learning (RL): challenges

In RL, an agent learns by interacting with an environment

- unknown environments
- delayed rewards or feedback
- astronomically large state and action space



Sample efficiency despite huge state/action space?

Collecting data samples might be expensive or time-consuming

- enormous sampling burden in the face of huge state/action space

clinical trials

online ads

Sample efficiency despite huge state/action space?

Collecting data samples might be expensive or time-consuming

- enormous sampling burden in the face of huge state/action space

clinical trials

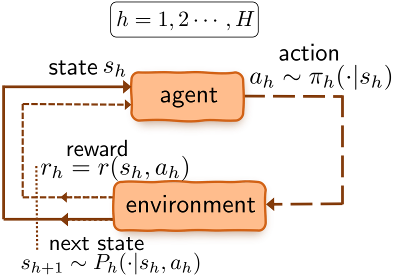
online ads

Key solution: exploiting low-complexity models
(a.k.a. function approximation)

This talk: MDPs with

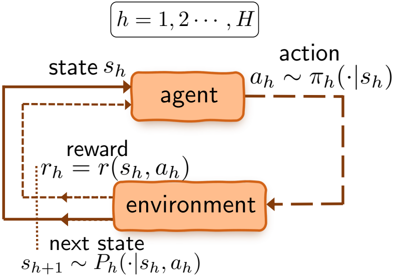
$\underbrace{\textit{linearly realizable optimal Q-functions}}_{\textit{linear } Q^\star}$

Episodic Markov decision process (MDP)



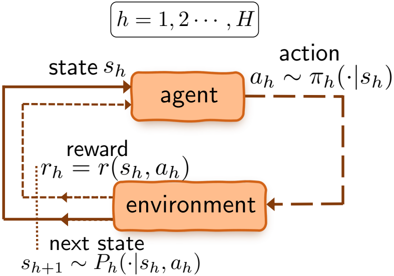
- H: horizon length

Episodic Markov decision process (MDP)



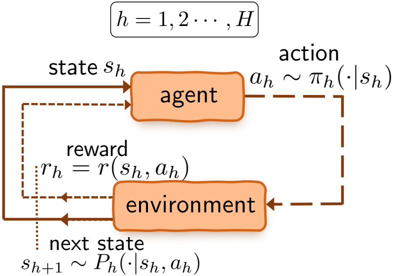
- H: horizon length
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space

Episodic Markov decision process (MDP)



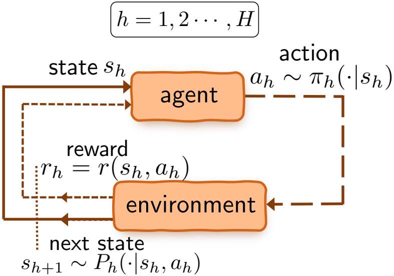
- H: horizon length
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r_h(s_h, a_h) \in [0, 1]$: immediate reward in step h

Episodic Markov decision process (MDP)



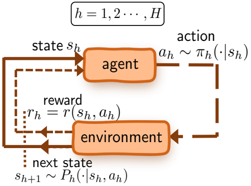
- H: horizon length
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r_h(s_h, a_h) \in [0, 1]$: immediate reward in step h
- $\pi = \{\pi_h\}_{h=1}^{H}$: policy (or action selection rule)

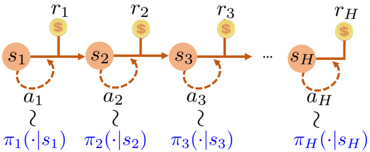
Episodic Markov decision process (MDP)



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

Value function and Q-function of policy π



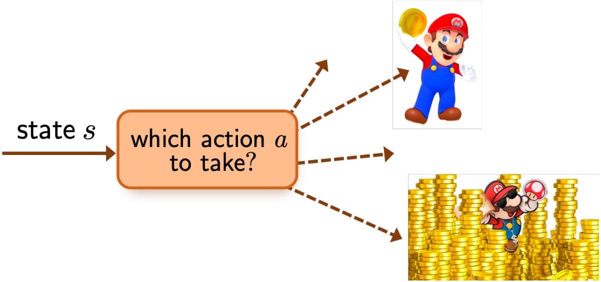


$$V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^{H} r_h(s_h, a_h) \,\middle|\, s_h = s\right]$$

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

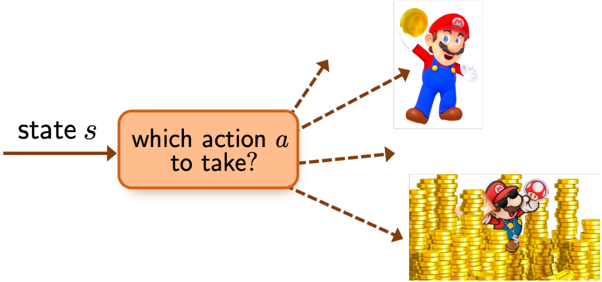
- execute policy π to generate sample trajectory

Optimal policy and optimal values



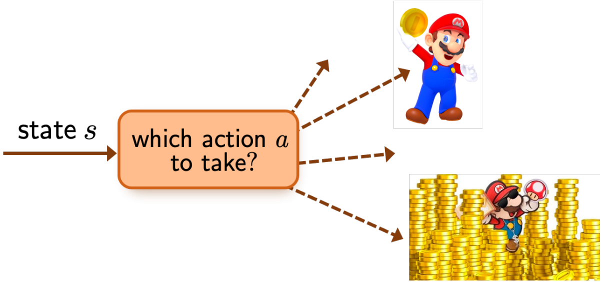
- Optimal policy $\pi^\star$: maximizing the value function

Optimal policy and optimal values



- Optimal policy $\pi^\star$: maximizing the value function
- Optimal value / Q function: $V_h^\star := V_h^{\pi^\star}$, $Q_h^\star := Q_h^{\pi^\star}$

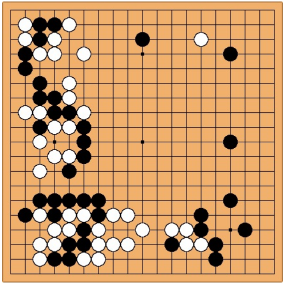
Optimal policy and optimal values



- Optimal policy $\pi^\star$: maximizing the value function
- Optimal value / Q function: $V_h^\star := V_h^{\pi^\star}$, $Q_h^\star := Q_h^{\pi^\star}$
- **Sub-optimality gap**:

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

Linear function representation



$$S \approx 2 \cdot 10^{170}$$

Exploiting **low-complexity model** is essential for sample-efficient RL!

Linear function representation

- Model-based (linear MDP): $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

Linear function representation

- Model-based (linear MDP): $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

$$\forall (s,a,h): \qquad P_h(\cdot \mid s,a) = \langle \varphi_h(s,a),\, \mu_h(\cdot) \rangle$$
$$r_h(s,a) = \langle \varphi_h(s,a),\, w_h(s,a) \rangle$$

Linear function representation

- Model-based (linear MDP): $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

$$\forall (s,a,h): \qquad P_h(\cdot \mid s,a) = \langle \varphi_h(s,a),\, \mu_h(\cdot) \rangle$$
$$r_h(s,a) = \langle \varphi_h(s,a),\, w_h(s,a) \rangle$$

$\Longrightarrow$ any $Q_h = r_h + P_h V_{h+1}$ is linearly representable

Linear function representation

- Model-based (linear MDP): $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

$$\forall (s,a,h): \qquad P_h(\cdot \mid s,a) = \langle \varphi_h(s,a),\, \mu_h(\cdot) \rangle$$
$$r_h(s,a) = \langle \varphi_h(s,a),\, w_h(s,a) \rangle$$

$\Longrightarrow$ any $Q_h = r_h + P_h V_{h+1}$ is linearly representable

- **Value-based (linear $Q^\star$)**: $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

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

Linear function representation

- Model-based (linear MDP): $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

$$\forall (s,a,h): \qquad P_h(\cdot \mid s,a) = \langle \varphi_h(s,a),\, \mu_h(\cdot) \rangle$$
$$r_h(s,a) = \langle \varphi_h(s,a),\, w_h(s,a) \rangle$$

$\Longrightarrow$ any $Q_h = r_h + P_h V_{h+1}$ is linearly representable

- **Value-based (linear $Q^\star$)**: $\exists$ features $\{\varphi_h(s,a) \in \mathbb{R}^d\}$ s.t.

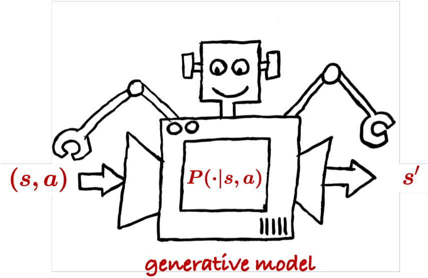
$$\forall (s,a,h): \qquad Q_h^\star(s,a) = \langle \varphi_h(s,a),\, \theta_h^\star \rangle$$

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

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

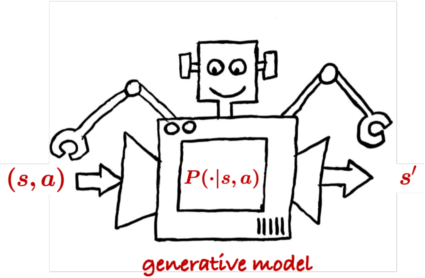
Prior art: RL with a generative model / simulator

Can query arbitrary state-action pairs to get samples



Prior art: RL with a generative model / simulator

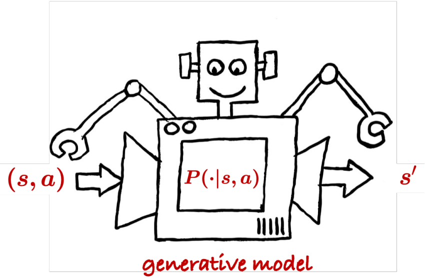
Can query arbitrary state-action pairs to get samples



- In general, needs $\min\left\{e^{\Omega(d)}, e^{\Omega(H)}\right\}$ samples (Weisz et al. '21)

Prior art: RL with a generative model / simulator

Can query arbitrary state-action pairs to get samples

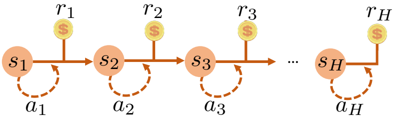


- In general, needs $\min\left\{e^{\Omega(d)}, e^{\Omega(H)}\right\}$ samples (Weisz et al. '21)
- With sub-optimality gap, needs only $\text{poly}(d, H, \frac{1}{\Delta_{\text{gap}}})$ samples (Du et al. '20)

Prior art: 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



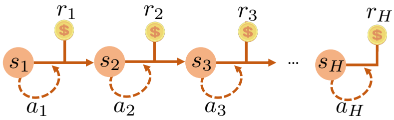
$$\Downarrow$$

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

Prior art: online RL

Obtain data samples via sequential interaction with environment

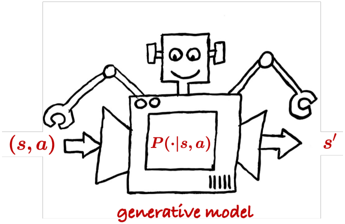
- collect N episodes of data, each consisting of H steps
- in the n-th episode, execute MDP using a policy π^n

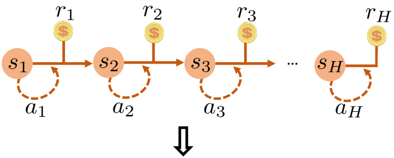


$$\Downarrow$$

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

Needs $\min\left\{e^{\Omega(d)}, e^{\Omega(H)}\right\}$ samples when $\Delta_{\text{gap}} \asymp 1$! (Wang et al. '21)



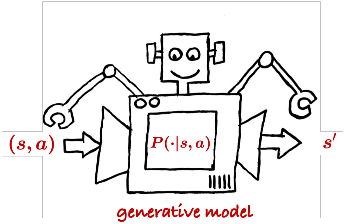


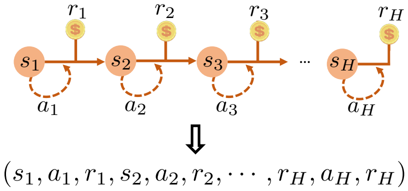
$$(s_1, a_1, r_1, s_2, a_2, r_2, \cdots, r_H, a_H, r_H)$$

generative model: idealistic

online RL: more restrictive/practical

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

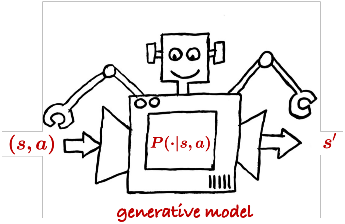


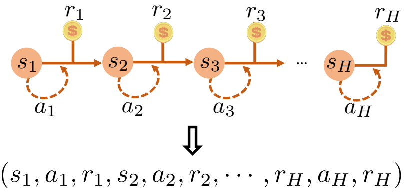


generative model: idealistic

online RL: more restrictive/practical

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





generative model: idealistic

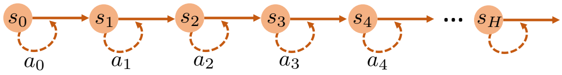
online RL: more restrictive/practical

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

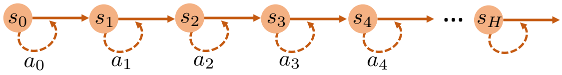
— *also called local access to generative model (Yin et al. '21)*



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. '21)*

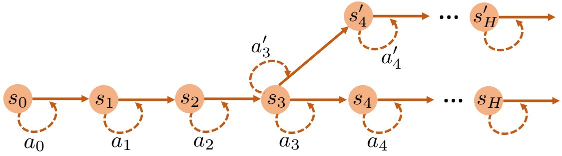


- **Input:** initial state (chosen by nature)
- generate a length-H trajectory

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. '21)*

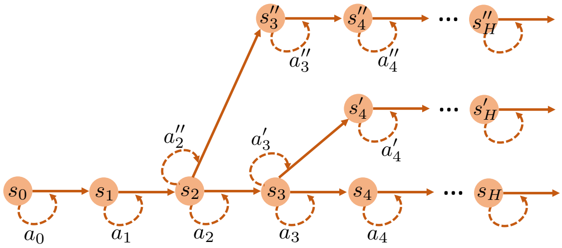


- **Input:** initial state (chosen by nature)
- generate a length-H trajectory
- Pick any previously visited state s_h in this episode, and repeat

A new sampling protocol: state revisiting

Allow one to revisit previous states in the same episode

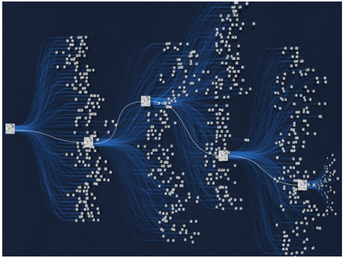
— *also called local access to generative model (Yin et al. '21)*



- **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

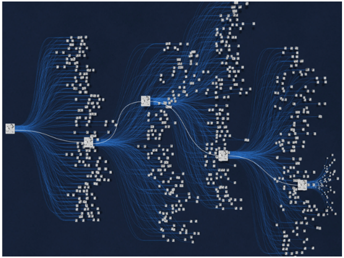
“save files” feature in video games



Monte Carlo Tree Search

A new sampling protocol: state revisiting

"save files" feature in video games

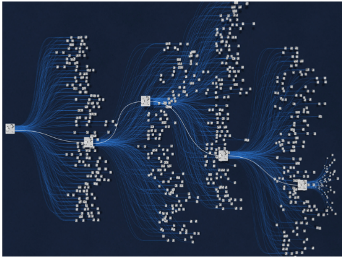


Monte Carlo Tree Search

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

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

Our contributions: a sample-efficient algorithm

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

$$\mathsf{Regret}(N) \coloneqq \sum_{n=1}^{N}\Big(V_1^\star(s_1^n) - V_1^{\pi^n}(s_1^n)\Big)$$

Our contributions: a sample-efficient algorithm

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

$$\mathsf{Regret}(N) := \sum_{n=1}^{N} \Big(V_1^{\star}(s_1^n) - V_1^{\pi^n}(s_1^n) \Big)$$

Theorem 1 (Li, Chen, Chi, Gu, Wei '21)

We propose an algorithm that achieves (up to log factor)

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

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

Implications

Theorem 2 (Li, Chen, Chi, Gu, Wei '21)

We propose an algorithm that achieves (up to log factor)

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

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

- Sample size needed to get ε average regret: $\mathsf{poly}\big(d, H, \frac{1}{\Delta_{\mathsf{gap}}}, \frac{1}{\varepsilon}\big)$, independent of S and A

Implications

Theorem 2 (Li, Chen, Chi, Gu, Wei '21)

We propose an algorithm that achieves (up to log factor)

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

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

- Sample size needed to get ε average regret: $\mathsf{poly}(d, H, \frac{1}{\Delta_{\mathsf{gap}}}, \frac{1}{\varepsilon})$, independent of S and A
- Limited state revisits: $\mathsf{poly}(d, H, \frac{1}{\Delta_{\mathsf{gap}}})$, almost independent of ε

Implications

Theorem 2 (Li, Chen, Chi, Gu, Wei '21)

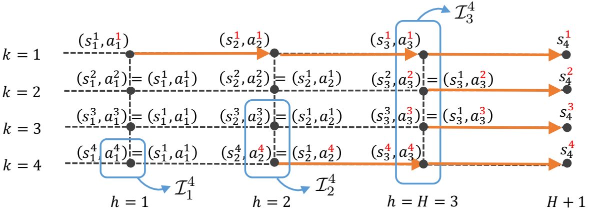
We propose an algorithm that achieves (up to log factor)

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

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

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

A glimpse of our algorithm: LinQ-LSVI-UCB

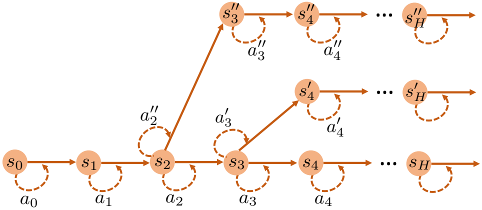


Key ingredients:

- Adapted from LSVI-UCB (originally designed for linear MDPs)
  Jin, Yang, Wang, Jordan '20
- Check exploration bonus: if this uncertainty term exceeds $\Delta_{\text{gap}}/2$, then revisit states to draw more samples

— *see our paper for detailed procedures*

Concluding remarks



- A new sampling protocol (more flexible than standard online RL yet still practically relevant)
- A sample-efficient solution: exploiting state revisiting to help remedy error accumulation/blowup across layers

"Sample-Efficient Reinforcement Learning Is Feasible for Linearly Realizable MDPs with Limited Revisiting," NeurIPS2021, arXiv:2105.08024