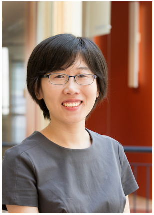
Statistical and Algorithmic Foundations of Reinforcement Learning

Yuting Wei
UPenn

Yuxin Chen
UPenn



Yuejie Chi
CMU

JSM Short Course, August 2023

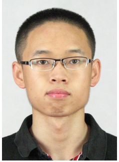
Statistical and Algorithmic Foundations of Reinforcement Learning (Part 1)

Yuting Wei

Statistics & Data Science, Wharton
University of Pennsylvania

JSM, August 2023

Our wonderful collaborators



Gen Li

UPenn $\rightarrow$ CUHK

Shicong Cen

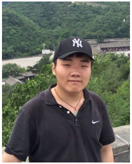
CMU

Chen Cheng

Stanford

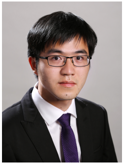
Laixi Shi

CMU $\rightarrow$ Caltech



Yuling Yan

Princeton $\rightarrow$ MIT

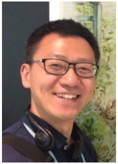


Changxiao Cai

UPenn $\rightarrow$ UMich

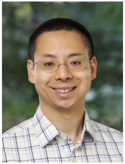
Wenhao Zhan

Princeton



Yuantao Gu

Tsinghua



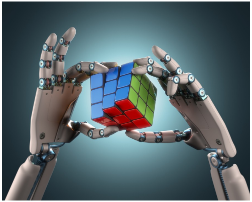
Jason Lee

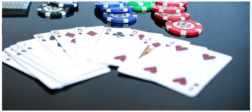
Princeton

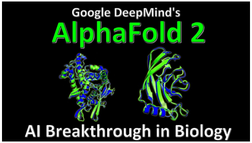
Jianqing Fan

Princeton

Recent successes in reinforcement learning (RL)







RL holds great promise in the next era of artificial intelligence.

Recap: Supervised learning

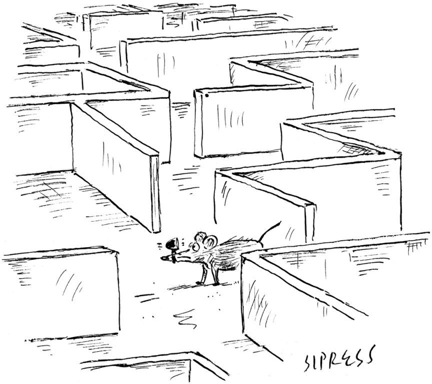
Given i.i.d training data, the goal is to make prediction on unseen data:

— *pic from internet*

Reinforcement learning (RL)

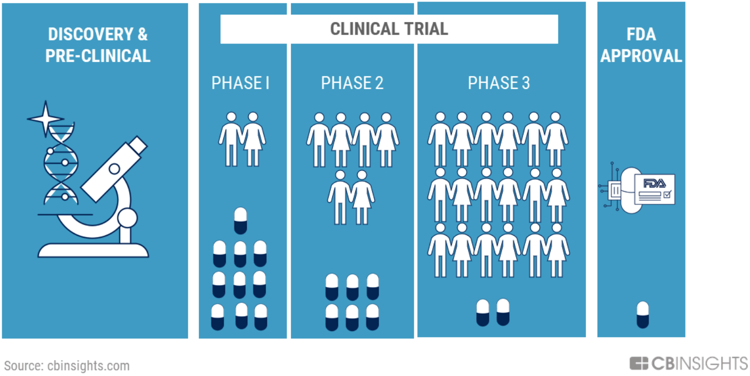
In RL, an agent learns by interacting with an environment.

- no training data
- trial-and-error
- maximize total rewards
- delayed reward



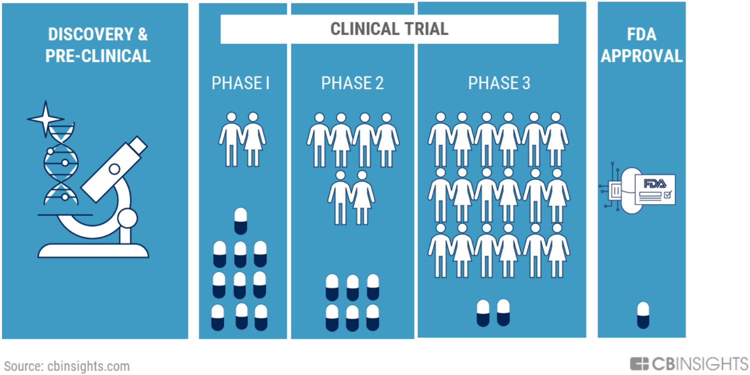
"Recalculating ... recalculating ..."

Sample efficiency



- prohibitively large state & action space
- collecting data samples can be expensive or time-consuming

Sample efficiency



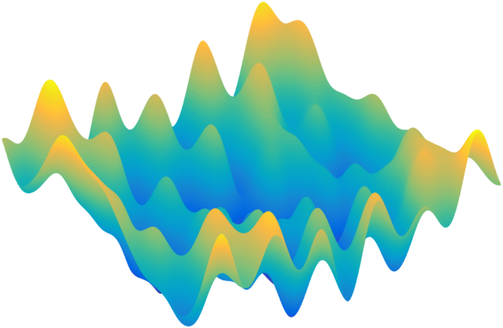
- prohibitively large state & action space
- collecting data samples can be expensive or time-consuming

Challenge: design sample-efficient RL algorithms

Computational efficiency

Running RL algorithms might take a long time . . .

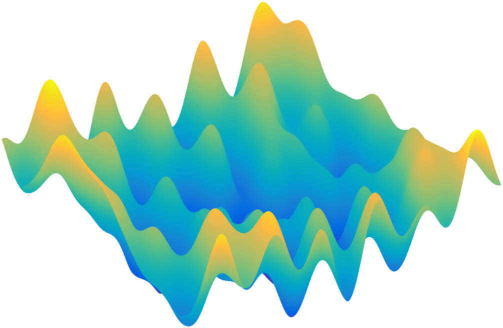
- enormous state-action space
- nonconvexity



Computational efficiency

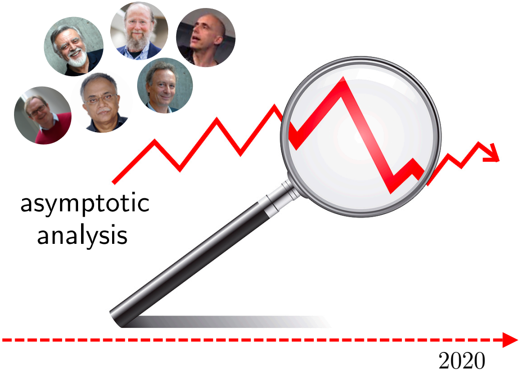
Running RL algorithms might take a long time . . .

- enormous state-action space
- nonconvexity



Challenge: design computationally efficient RL algorithms

Theoretical foundation of RL



Statistical Science
1986, Vol. 1, No. 2, 276–284

The Contributions of Herbert Robbins to Mathematical Statistics

Tze Leung Lai and David Siegmund

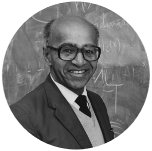
2. STOCHASTIC APPROXIMATION AND ADAPTIVE DESIGN

In 1951, Robbins and his student, Sutton Monro, founded the subject of stochastic approximation with the publication of their celebrated paper [26]. Consider the problem of finding the root θ (assumed unique) of an equation $g(x) = 0$. In the classical

4. SEQUENTIAL EXPERIMENTATION AND OPTIMAL STOPPING

The well known "multiarmed bandit problem" in the statistics and engineering literature, which is prototypical of a wide variety of adaptive control and design problems, was first formulated and studied by Robbins [28]. Let A, B denote two statistical populations with finite means μ_A, μ_B. How should we draw a

Herbert Robbins



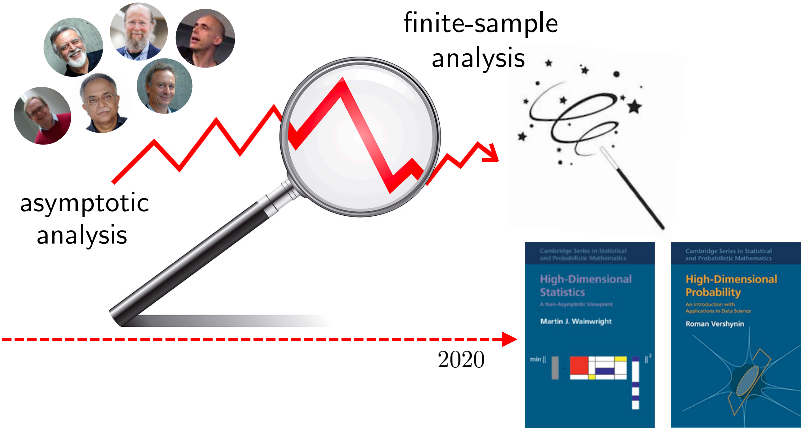
David Blackwell

David Blackwell, 1919–2010: An explorer in mathematics and statistics

Peter J. Bickel[a,1]

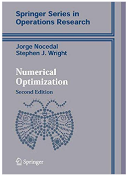
Blackwell channel. He also began to work in dynamic programming, which is now called reinforcement learning. In a series of papers, Blackwell gave a rigorous foundation to the theory of dynamic programming, introducing what have become known as Blackwell optimal policies.

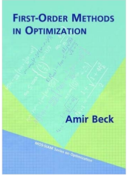
Theoretical foundation of RL

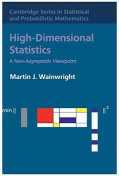


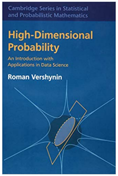
Understanding sample efficiency of RL requires a modern suite of non-asymptotic analysis tools

This tutorial





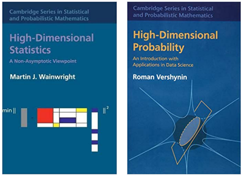


(large-scale) optimization

(high-dimensional) statistics

Demystify sample- and computational efficiency of RL algorithms

This tutorial



(large-scale) optimization (high-dimensional) statistics

Demystify sample- and computational efficiency of RL algorithms

Part 1. **basics, model-based and model-free RL**

Part 2. **online/offline RL, reward-free RL, hybrid RL**

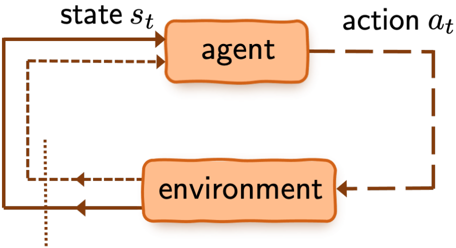
Part 3. **federated RL, robust RL, policy optimization**

Outline (Part 1)

- Basics: Markov decision processes
- Basic dynamic programming algorithms
- Model-based RL ("plug-in" approach)
- Value-based RL (a model-free approach)

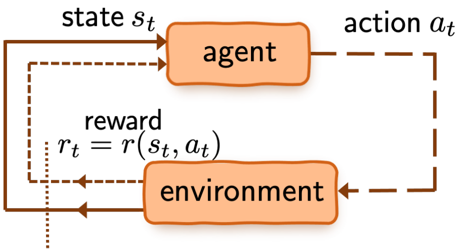
Basics: Markov decision processes

Markov decision process (MDP)



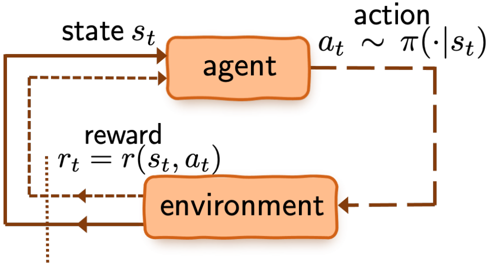
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space

Markov decision process (MDP)



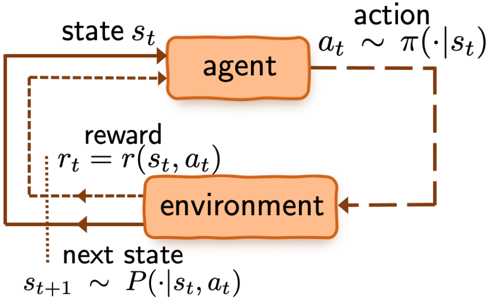
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward

Infinite-horizon Markov decision process



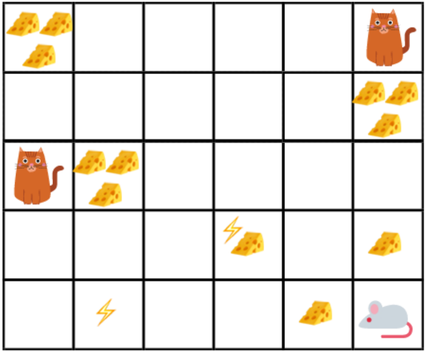
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)

Infinite-horizon Markov decision process

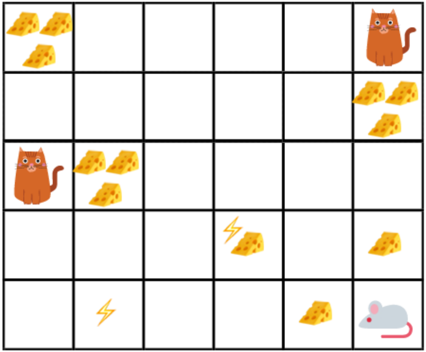


- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s,a) \in [0,1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)
- $P(\cdot|s,a)$: unknown transition probabilities

Help the mouse!

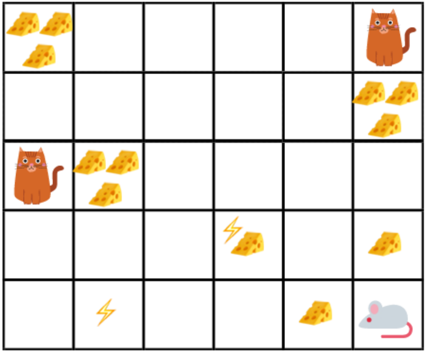


Help the mouse!



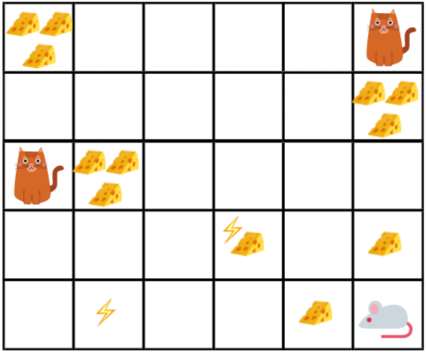
- state space $\mathcal{S}$: positions in the maze

Help the mouse!



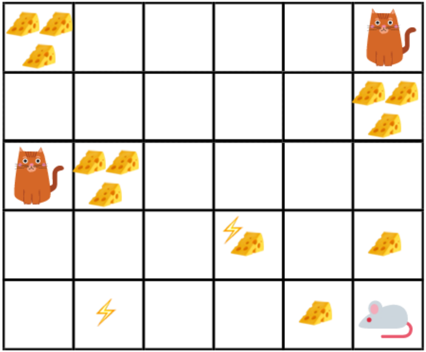
- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right

Help the mouse!



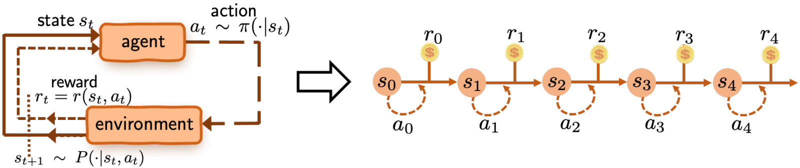
- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right
- immediate reward r: cheese, electricity shocks, cats

Help the mouse!



- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right
- immediate reward r: cheese, electricity shocks, cats
- policy $\pi(\cdot|s)$: the way to find cheese

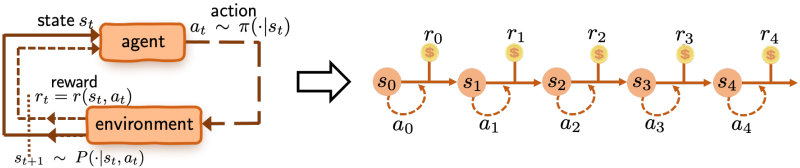
Value function



Value of policy π: cumulative discounted reward

$$\forall s \in \mathcal{S}: \quad V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s\right]$$

Value function



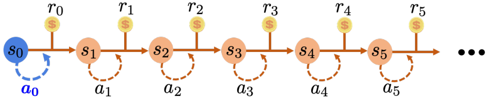
Value of policy π: cumulative discounted reward

$$\forall s \in \mathcal{S}: \quad V^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t) \,\middle|\, s_0 = s\right]$$

- $\gamma \in [0, 1)$: discount factor
  - take $\gamma \to 1$ to approximate long-horizon MDPs
  - **effective horizon**: $\frac{1}{1-\gamma}$

Q-function (action-value function)

$Q^\pi(s_0, a_0)$

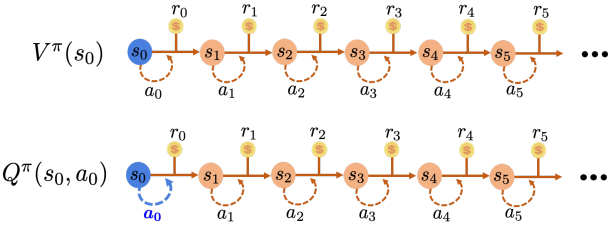


Q-function of policy π:

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A} : \quad Q^\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a \right]$$

- $(\cancel{a_0,}\ s_1, a_1, s_2, a_2, \cdots)$: induced by policy π

Q-function (action-value function)

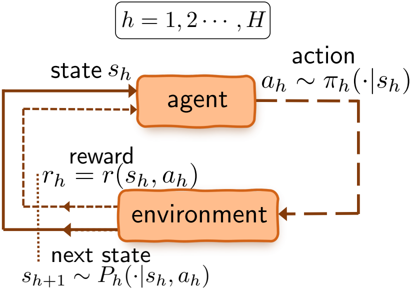


Q-function of policy π:

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad Q^\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a\right]$$

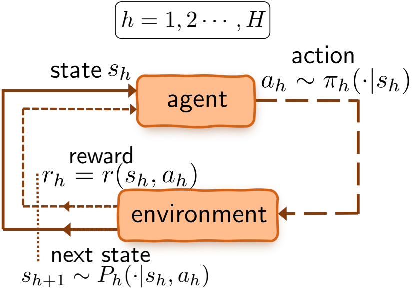
- $(\cancel{a_0,}\, s_1, a_1, s_2, a_2, \cdots)$: induced by policy π

Finite-horizon MDPs



- 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

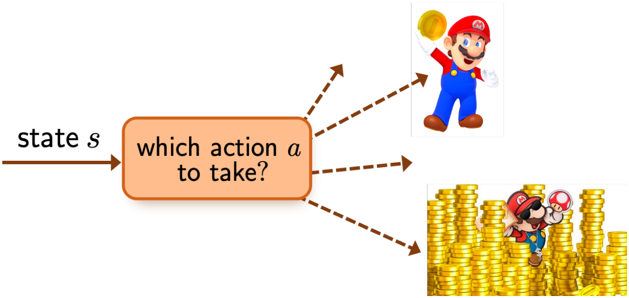
Finite-horizon MDPs



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

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

Optimal policy and optimal value



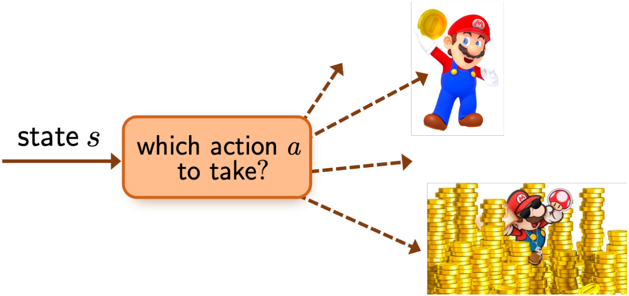
optimal policy $\pi^\star$: maximizing value function $\max_\pi V^\pi$

Proposition (Puterman'94)

For infinite horizon discounted MDP, there always exists a deterministic policy $\pi^\star$*, such that*

$$V^{\pi^\star}(s) \geq V^\pi(s), \quad \forall s, \text{ and } \pi.$$

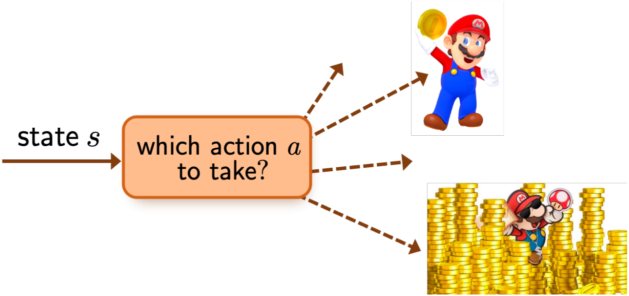
Optimal policy and optimal value



optimal policy $\pi^\star$: maximizing value function $\max_\pi V^\pi$

- optimal value / Q function: $V^\star := V^{\pi^\star}$, $Q^\star := Q^{\pi^\star}$

Optimal policy and optimal value



optimal policy $\pi^\star$: maximizing value function $\max_\pi V^\pi$

- optimal value / Q function: $V^\star := V^{\pi^\star}$, $Q^\star := Q^{\pi^\star}$
- How to find this $\pi^\star$?

Basic dynamic programming algorithms when MDP specification is known

Policy evaluation: Given MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$ and policy $\pi : \mathcal{S} \mapsto \mathcal{A}$, how good is π? (i.e., how to compute $V^\pi(s), \ \forall s$?)

Policy evaluation: Given MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$ and policy $\pi : \mathcal{S} \mapsto \mathcal{A}$, how good is π? (i.e., how to compute $V^{\pi}(s), \ \forall s$?)

Possible scheme:

- execute policy evaluation for each π
- find the optimal one

Policy evaluation: Bellman's consistency equation

- V^π / Q^π: value / action-value function under policy π

Policy evaluation: Bellman's consistency equation

- V^π / Q^π: value / action-value function under policy π

Bellman's consistency equation

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q^\pi(s,a)\right]$$

$$Q^\pi(s,a) = \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)}\Big[\underbrace{V^\pi(s')}_{\text{next state's value}} \Big]$$

Richard Bellman

Policy evaluation: Bellman's consistency equation

- V^π / Q^π: value / action-value function under policy π

Bellman's consistency equation

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[Q^\pi(s,a)\big]$$

$$Q^\pi(s,a) = \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)}\Big[\underbrace{V^\pi(s')}_{\text{next state's value}}\Big]$$

- one-step look-ahead

Richard Bellman

Policy evaluation: Bellman's consistency equation

- V^π / Q^π: value / action-value function under policy π

Bellman's consistency equation

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}\left[Q^\pi(s,a)\right]$$

$$Q^\pi(s,a) = \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)}\Big[\underbrace{V^\pi(s')}_{\text{next state's value}}\Big]$$

- one-step look-ahead
- let P^π be the state-action transition matrix induced by π:

$$Q^\pi = r + \gamma P^\pi Q^\pi \quad \Longrightarrow \quad Q^\pi = (I - \gamma P^\pi)^{-1} r$$

Richard Bellman

Optimal policy $\pi^\star$: Bellman's optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s',a')\right]}_{\text{next state's value}}$$

- one-step look-ahead

Optimal policy $\pi^\star$: Bellman's optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \underset{s' \sim P(\cdot|s,a)}{\mathbb{E}} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s',a')\right]}_{\text{next state's value}}$$

- one-step look-ahead

Bellman equation: $Q^\star$ is *unique* solution to

$$\mathcal{T}(Q^\star) = Q^\star$$

γ-contraction of Bellman operator:

$$\|\mathcal{T}(Q_1) - \mathcal{T}(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

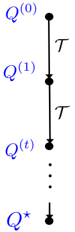
Richard Bellman

Two dynamic programming algorithms

Value iteration (VI)

For $t = 0, 1, \ldots,$

$$Q^{(t+1)} = \mathcal{T}(Q^{(t)})$$

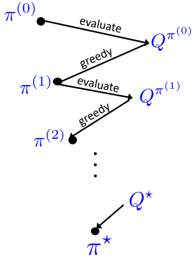


Policy iteration (PI)

For $t = 0, 1, \ldots,$

policy evaluation: $Q^{(t)} = Q^{\pi^{(t)}}$

policy improvement: $\pi^{(t+1)}(s) = \arg\max_{a \in \mathcal{A}} Q^{(t)}(s, a)$



Iteration complexity

Theorem (Linear convergence of policy/value iteration)

$$\|Q^{(t)} - Q^{\star}\|_{\infty} \leq \gamma^{t}\|Q^{(0)} - Q^{\star}\|_{\infty}$$

Iteration complexity

Theorem (Linear convergence of policy/value iteration)

$$\big\|Q^{(t)} - Q^{\star}\big\|_{\infty} \le \gamma^t \big\|Q^{(0)} - Q^{\star}\big\|_{\infty}$$

Implications: to achieve $\|Q^{(t)} - Q^{\star}\|_{\infty} \le \varepsilon$, it takes no more than

$$\frac{1}{1-\gamma} \log\left(\frac{\|Q^{(0)} - Q^{\star}\|_{\infty}}{\varepsilon}\right) \quad \text{iterations}$$

Iteration complexity

Theorem (Linear convergence of policy/value iteration)

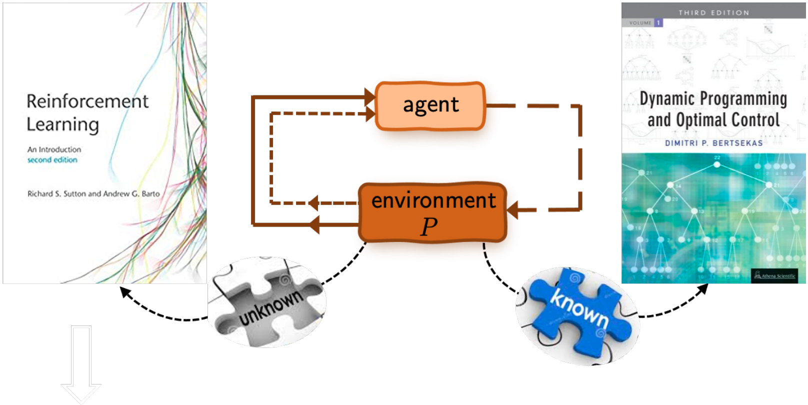
$$\|Q^{(t)} - Q^\star\|_\infty \le \gamma^t \|Q^{(0)} - Q^\star\|_\infty$$

Implications: to achieve $\|Q^{(t)} - Q^\star\|_\infty \le \varepsilon$, it takes no more than

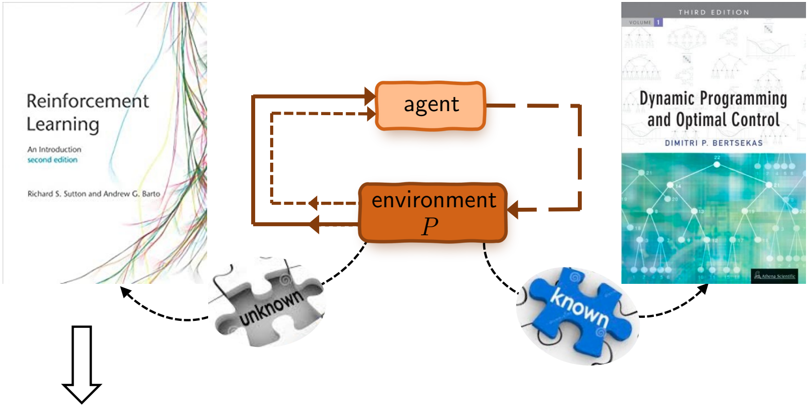
$$\frac{1}{1-\gamma} \log\left(\frac{\|Q^{(0)} - Q^\star\|_\infty}{\varepsilon}\right) \quad \text{iterations}$$

Linear convergence at a **dimension-free** rate!

When the model is unknown . . .

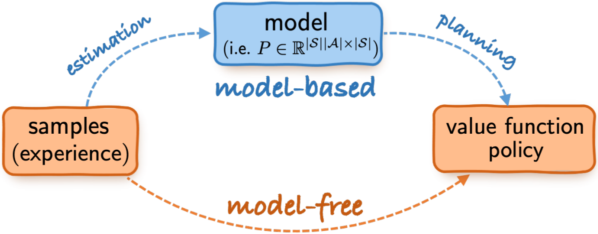


When the model is unknown ...



Need to learn optimal policy from samples w/o model specification

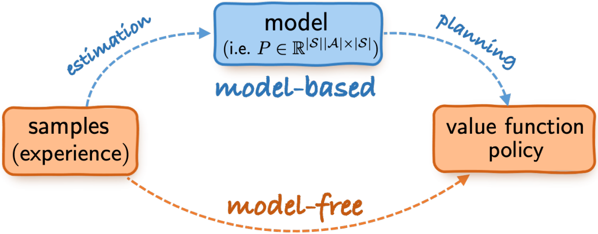
Three approaches



Model-based approach ("plug-in")

1. build an empirical estimate $\widehat{P}$ for P
2. planning based on the empirical $\widehat{P}$

Three approaches



Model-based approach ("plug-in")

1. build an empirical estimate $\widehat{P}$ for P
2. planning based on the empirical $\widehat{P}$

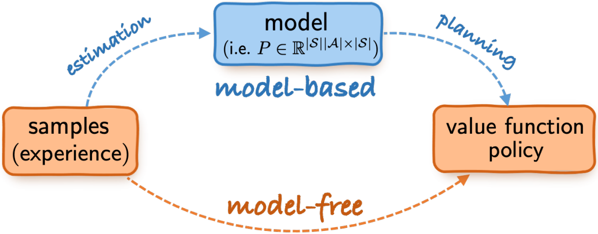
Value-based approach

— learning w/o estimating the model explicitly

Policy-based approach

— optimization in the space of policies

Three approaches



Model-based approach ("plug-in")

1. build an empirical estimate $\widehat{P}$ for P
2. planning based on the empirical $\widehat{P}$

Value-based approach

— learning w/o estimating the model explicitly

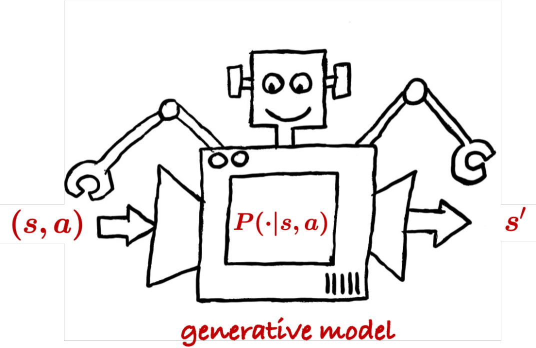
Policy-based approach

— optimization in the space of policies

Model-based RL (a "plug-in" approach)

A generative model / simulator

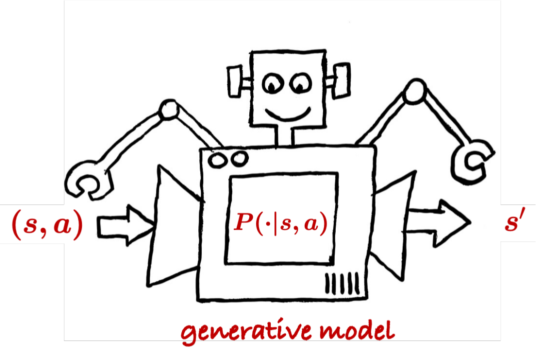
— *Kearns and Singh, 1999*



- **sampling:** for each (s, a), collect N samples $\{(s, a, s'_{(i)})\}_{1 \le i \le N}$

A generative model / simulator

— *Kearns and Singh, 1999*



- **sampling:** for each (s, a), collect N samples $\{(s, a, s'_{(i)})\}_{1 \le i \le N}$
- construct $\widehat{\pi}$ based on samples (in total $|\mathcal{S}||\mathcal{A}| \times N$)

ℓ_∞-**sample complexity:** how many samples are required to learn an $\underbrace{\varepsilon\text{-optimal policy}}_{\forall s:\ V^{\hat{\pi}}(s) \geq V^\star(s) - \varepsilon}$?

An incomplete list of works

- Kearns and Singh, 1999
- Kakade, 2003
- Kearns et al., 2002
- Azar et al., 2012
- Azar et al., 2013
- Sidford et al., 2018a, 2018b
- Wang, 2019
- Agarwal et al., 2019
- Wainwright, 2019a, 2019b
- Pananjady and Wainwright, 2019
- Yang and Wang, 2019
- Khamaru et al., 2020
- Mou et al., 2020
- Cui and Yang, 2021
- ...

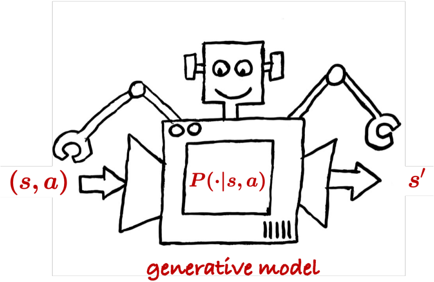
An even shorter list of prior art

algorithm	sample size range	sample complexity	ε-range
Empirical QVI Azar et al., 2013	$\big[\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert}{(1-\gamma)^2},\infty\big)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$\big(0,\frac{1}{\sqrt{(1-\gamma)\lvert\mathcal{S}\rvert}}\big]$
Sublinear randomized VI Sidford et al., 2018b	$\big[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2},\infty\big)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\varepsilon^2}$	$\big(0,\frac{1}{1-\gamma}\big]$
Variance-reduced QVI Sidford et al., 2018a	$\big[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3},\infty\big)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$(0,1]$
Randomized primal-dual Wang 2019	$\big[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2},\infty\big)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\varepsilon^2}$	$\big(0,\frac{1}{1-\gamma}\big]$
Empirical MDP + planning Agarwal et al., 2019	$\big[\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^2},\infty\big)$	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\varepsilon^2}$	$\big(0,\frac{1}{\sqrt{1-\gamma}}\big]$

important parameters $\Longrightarrow$

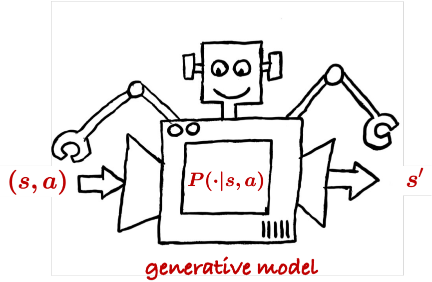
- # states $|\mathcal{S}|$, # actions $|\mathcal{A}|$
- the discounted complexity $\frac{1}{1-\gamma}$
- approximation error $\varepsilon \in (0, \frac{1}{1-\gamma}]$

Model estimation



Sampling: for each (s, a), collect N ind. samples $\{(s, a, s'_{(i)})\}_{1 \le i \le N}$

Model estimation



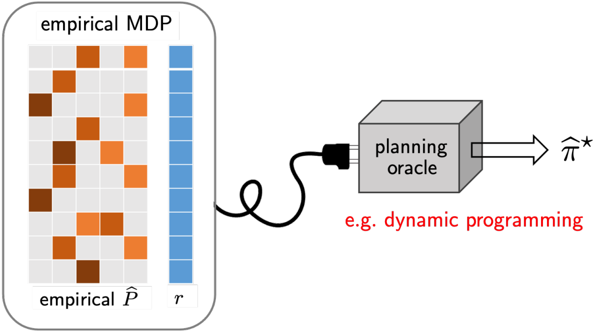
Sampling: for each (s,a), collect N ind. samples $\{(s,a,s'_{(i)})\}_{1\leq i\leq N}$

Empirical estimates:

$$\widehat{P}(s'|s,a) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\{s'_{(i)} = s'\}}_{\text{empirical frequency}}$$

Empirical MDP + planning

— Azar et al., 2013, Agarwal et al., 2019

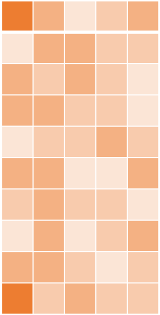


Find policy based on the empirical MDP (*empirical maximizer*)

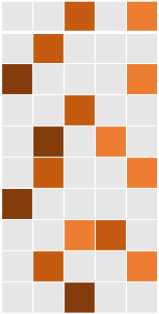
using, e.g., policy iteration

$(\widehat{P}, r)$

Challenges in the sample-starved regime



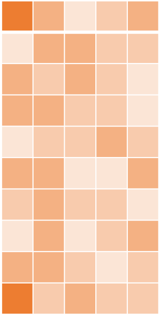
truth: $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$



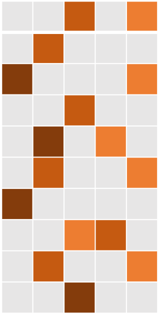
empirical estimate: $\widehat{P}$

- Can't recover P faithfully if sample size $\ll |\mathcal{S}|^2|\mathcal{A}|$!

Challenges in the sample-starved regime



truth: $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$



empirical estimate: $\widehat{P}$

- Can't recover P faithfully if sample size $\ll |\mathcal{S}|^2|\mathcal{A}|$!
- Can we trust our policy estimate when reliable model estimation is infeasible?

ℓ_∞-based sample complexity

Theorem (Agarwal, Kakade, Yang '19)

For any $0 < \varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$*, the optimal policy* $\widehat{\pi}^\star$ *of empirical MDP achieves*

$$\|V^{\widehat{\pi}^\star} - V^\star\|_\infty \leq \varepsilon$$

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

ℓ_∞-based sample complexity

Theorem (Agarwal, Kakade, Yang '19)

For any $0 < \varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$*, the optimal policy* $\widehat{\pi}^\star$ *of empirical MDP achieves*

$$\|V^{\widehat{\pi}^\star} - V^\star\|_\infty \leq \varepsilon$$

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

- matches minimax lower bound: $\widetilde{\Omega}(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2})$ when $\varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$ (equivalently, when sample size exceeds $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$) Azar et al., 2013

ℓ_∞-based sample complexity

Theorem (Agarwal, Kakade, Yang '19)

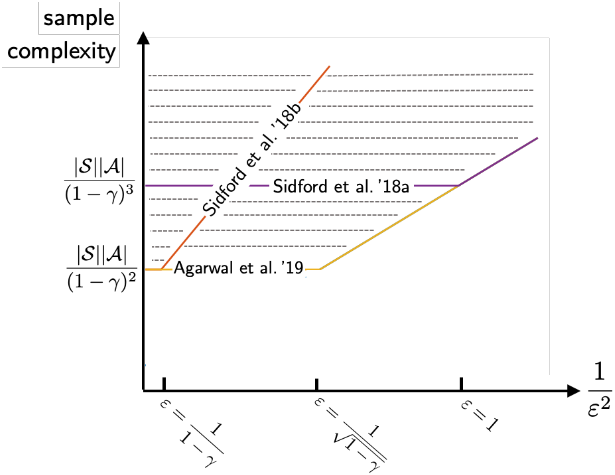
For any $0 < \varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$, *the optimal policy* $\widehat{\pi}^\star$ *of empirical MDP achieves*

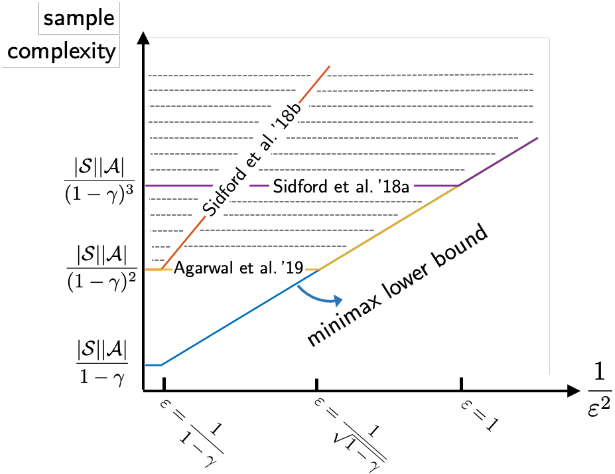
$$\|V^{\widehat{\pi}^\star} - V^\star\|_\infty \leq \varepsilon$$

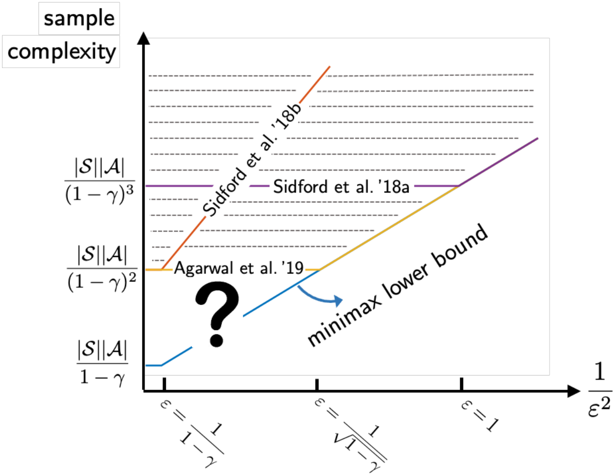
with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

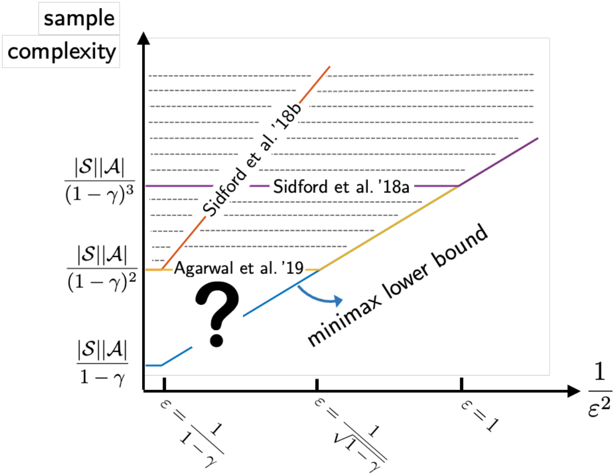
- matches minimax lower bound: $\widetilde{\Omega}(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2})$ when $\varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$ (equivalently, when sample size exceeds $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$) Azar et al., 2013
- established upon leave-one-out analysis framework


sample
complexity
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3}$
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$
Sidford et al. '18b
Sidford et al. '18a
Agarwal et al. '19
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = \frac{1}{\sqrt{1-\gamma}}$
$\varepsilon = 1$
$\frac{1}{\varepsilon^2}$


sample
complexity
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3}$
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$
$\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma}$
Sidford et al. '18b
Sidford et al. '18a
Agarwal et al. '19
minimax lower bound
$\frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = \frac{1}{\sqrt{1-\gamma}}$
$\varepsilon = 1$



Agarwal et al., 2019 still requires a burn-in sample size $\gtrsim \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$

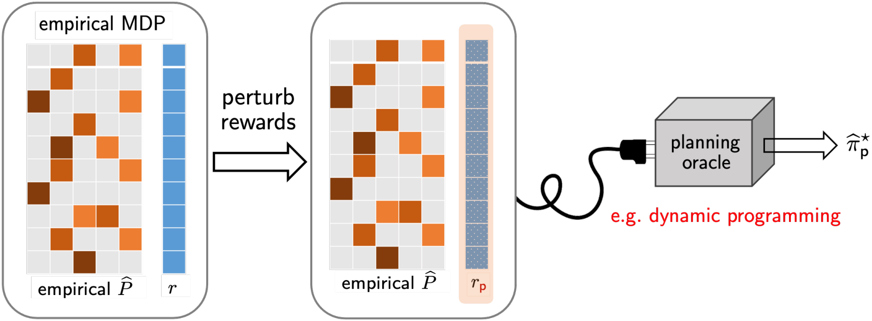


Agarwal et al., 2019 still requires a burn-in sample size $\gtrsim \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$

Question: is it possible to break this sample size barrier?

Perturbed model-based approach (Li et al. '20)

—*Li et al., 2020*



Find policy based on the empirical MDP with slightly perturbed rewards

Optimal ℓ_∞-based sample complexity

Theorem (Li, Wei, Chi, Chen '20)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$*, the optimal policy* $\widehat{\pi}_{\mathrm{p}}^{\star}$ *of perturbed empirical MDP achieves*

$$\|V^{\widehat{\pi}_{\mathrm{p}}^{\star}} - V^{\star}\|_{\infty} \leq \varepsilon$$

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

Optimal ℓ_∞-based sample complexity

Theorem (Li, Wei, Chi, Chen '20)

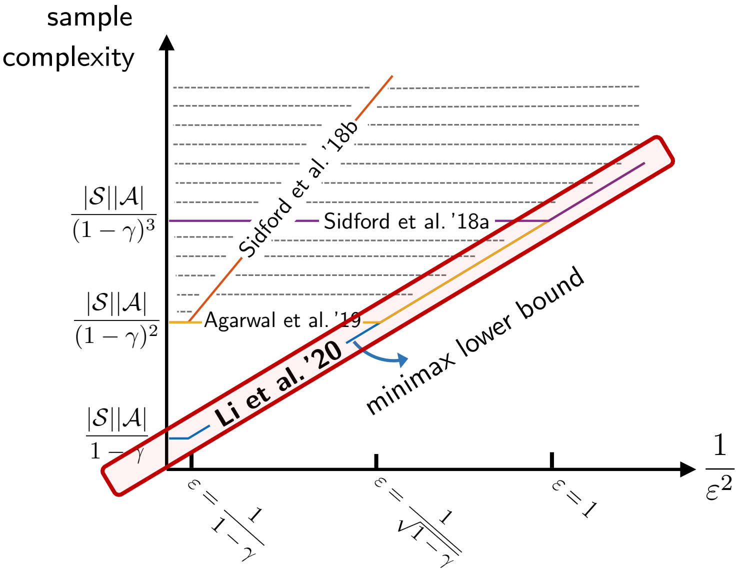
For any $0 < \varepsilon \le \frac{1}{1-\gamma}$, *the optimal policy* $\widehat{\pi}_{\mathrm{p}}^\star$ *of perturbed empirical MDP achieves*

$$\|V^{\widehat{\pi}_{\mathrm{p}}^\star} - V^\star\|_\infty \le \varepsilon$$

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

- matches minimax lower bound: $\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$ Azar et al., 2013
- full ε-range: $\varepsilon \in \left(0, \frac{1}{1-\gamma}\right]$ $\longrightarrow$ no burn-in cost
- established upon more refined leave-one-out analysis and a perturbation argument



sample complexity
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3}$
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$
$\frac{|\mathcal{S}||\mathcal{A}|}{1-\gamma}$
Sidford et al. '18b
Sidford et al. '18a
Agarwal et al. '19
Li et al. '20
minimax lower bound
$\frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = \frac{1}{\sqrt{1-\gamma}}$
$\varepsilon = 1$

A sketch of the main proof ingredients

Notation and Bellman equation

Bellman equation: $V^\pi = r_\pi + \gamma P_\pi V^\pi$

- V^π: value function under policy π
  - Bellman equation: $V^\pi = (I - \gamma P_\pi)^{-1} r_\pi$
- $\widehat{V}^\pi$: empirical version value function under policy π
  - Bellman equation: $\widehat{V}^\pi = (I - \gamma \widehat{P}_\pi)^{-1} r_\pi$

Notation and Bellman equation

Bellman equation: $V^\pi = r_\pi + \gamma P_\pi V^\pi$

- V^π: value function under policy π
  - Bellman equation: $V^\pi = (I - \gamma P_\pi)^{-1} r_\pi$
- $\widehat{V}^\pi$: empirical version value function under policy π
  - Bellman equation: $\widehat{V}^\pi = (I - \gamma \widehat{P}_\pi)^{-1} r_\pi$
- $\pi^\star$: optimal policy for V^π
- $\widehat{\pi}^\star$: optimal policy for $\widehat{V}^\pi$

Main steps

Elementary decomposition:

$$\begin{aligned} V^\star - V^{\widehat{\pi}^\star} &= \big(V^\star - \widehat{V}^{\pi^\star}\big) + \big(\widehat{V}^{\pi^\star} - \widehat{V}^{\widehat{\pi}^\star}\big) + \big(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\big) \\ &\leq \big(V^{\pi^\star} - \widehat{V}^{\pi^\star}\big) + 0 + \big(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\big) \end{aligned}$$

Main steps

Elementary decomposition:

$$V^\star - V^{\widehat{\pi}^\star} = \left(V^\star - \widehat{V}^{\pi^\star}\right) + \left(\widehat{V}^{\pi^\star} - \widehat{V}^{\widehat{\pi}^\star}\right) + \left(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\right)$$
$$\leq \left(V^{\pi^\star} - \widehat{V}^{\pi^\star}\right) + 0 + \left(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\right)$$

- **Step 1:** control $V^\pi - \widehat{V}^\pi$ for a fixed π (called "policy evaluation") (Bernstein inequality + a peeling argument)

Main steps

Elementary decomposition:

$$V^\star - V^{\widehat{\pi}^\star} = \big(V^\star - \widehat{V}^{\pi^\star}\big) + \big(\widehat{V}^{\pi^\star} - \widehat{V}^{\widehat{\pi}^\star}\big) + \big(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\big)$$
$$\leq \big(V^{\pi^\star} - \widehat{V}^{\pi^\star}\big) + 0 + \big(\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}\big)$$

- **Step 1:** control $V^\pi - \widehat{V}^\pi$ for a fixed π (called "policy evaluation") (Bernstein inequality + a peeling argument)
- **Step 2:** extend it to control $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$ ($\widehat{\pi}^\star$ depends on samples) (decouple statistical dependency)

Key idea 1: a peeling argument (for fixed policy)

First-order expansion

$$\widehat{V}^{\pi} - V^{\pi} = \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\widehat{V}^{\pi}$$ [Agarwal et al., 2019]

Key idea 1: a peeling argument (for fixed policy)

First-order expansion

$$\widehat{V}^{\pi} - V^{\pi} = \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\widehat{V}^{\pi} \qquad \text{[Agarwal et al., 2019]}$$

Ours: higher-order expansion + Bernstein $\longrightarrow$ tighter control

$$\widehat{V}^{\pi} - V^{\pi} = \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)V^{\pi} + \\ + \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\big(\widehat{V}^{\pi} - V^{\pi}\big)$$

Bernstein's inequality: $|\big(\widehat{P}_{\pi} - P_{\pi}\big)V^{\pi}| \leq \sqrt{\frac{Var[V^{\pi}]}{N}} + \frac{\|V^{\pi}\|_{\infty}}{N}$

Key idea 1: a peeling argument (for fixed policy)

First-order expansion

$$\widehat{V}^{\pi} - V^{\pi} = \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\widehat{V}^{\pi} \qquad \text{[Agarwal et al., 2019]}$$

Ours: higher-order expansion + Bernstein $\longrightarrow$ tighter control

$$\begin{aligned}\widehat{V}^{\pi} - V^{\pi} = \gamma\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)V^{\pi} + \\ + \gamma^{2}\Big(\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\Big)^{2}V^{\pi} \\ + \gamma^{3}\Big(\big(I - \gamma P_{\pi}\big)^{-1}\big(\widehat{P}_{\pi} - P_{\pi}\big)\Big)^{3}V^{\pi} \\ + \dots\end{aligned}$$

Bernstein's inequality: $\big|\big(\widehat{P}_{\pi} - P_{\pi}\big)V^{\pi}\big| \le \sqrt{\frac{Var[V^{\pi}]}{N}} + \frac{\|V^{\pi}\|_{\infty}}{N}$

Byproduct: policy evaluation

Theorem (Li, Wei, Chi, Gu, Chen'20)

Fix any policy π*. For every* $0 < \varepsilon \leq \frac{1}{1-\gamma}$*, plug-in estimator* $\widehat{V}^{\pi}$ *obeys*

$$\|\widehat{V}^{\pi} - V^{\pi}\|_{\infty} \leq \varepsilon$$

with sample complexity at most

$$\widetilde{O}\Big(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\Big).$$

Byproduct: policy evaluation

Theorem (Li, Wei, Chi, Gu, Chen'20)

Fix any policy π*. For every* $0 < \varepsilon \le \frac{1}{1-\gamma}$*, plug-in estimator* $\widehat{V}^{\pi}$ *obeys*

$$\|\widehat{V}^{\pi} - V^{\pi}\|_{\infty} \le \varepsilon$$

with sample complexity at most

$$\widetilde{O}\Big(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\Big).$$

- minimax lower bound [Azar et al., 2013, Pananjady and Wainwright, 2019]

Byproduct: policy evaluation

Theorem (Li, Wei, Chi, Gu, Chen'20)

Fix any policy π. *For every* $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *plug-in estimator* $\widehat{V}^{\pi}$ *obeys*

$$\|\widehat{V}^{\pi} - V^{\pi}\|_{\infty} \leq \varepsilon$$

with sample complexity at most

$$\widetilde{O}\Big(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\Big).$$

- minimax lower bound [Azar et al., 2013, Pananjady and Wainwright, 2019]
- tackle sample size barrier: prior work requires sample size $> \frac{|\mathcal{S}|}{(1-\gamma)^2}$ [Agarwal et al., 2013, Pananjady and Wainwright, 2019, Khamaru et al,, 2020]

Step 2: controlling $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

A natural idea: apply our policy evaluation theory + union bound

Step 2: controlling $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

A natural idea: apply our policy evaluation theory + union bound

- highly suboptimal!

Step 2: controlling $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

A natural idea: apply our policy evaluation theory + union bound

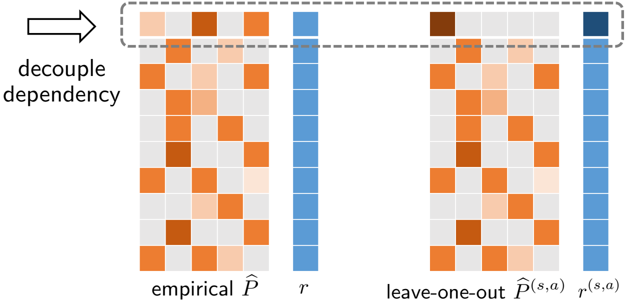
- highly suboptimal!

key idea 2: a leave-one-out argument to decouple stat. dependency btw $\widehat{\pi}$ and samples

— *inspired by [Agarwal et al., 2019] but quite different ...*

Key idea 2: decouple dependency for $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

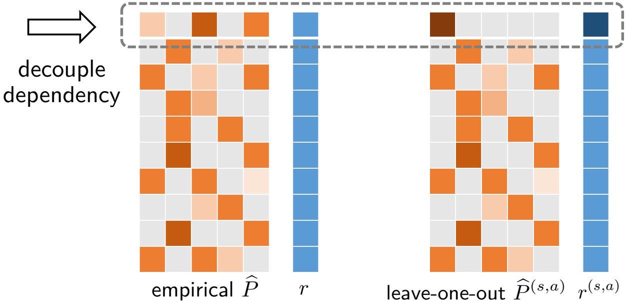
— *inspired by [Agarwal et al., 2019] but quite different ...*



- define $\widehat{\pi}^\star_{(s,a)} \xrightarrow{\text{empirical maximizer}} \left(\widehat{P}^{(s,a)}, r^{(s,a)}\right)$

Key idea 2: decouple dependency for $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

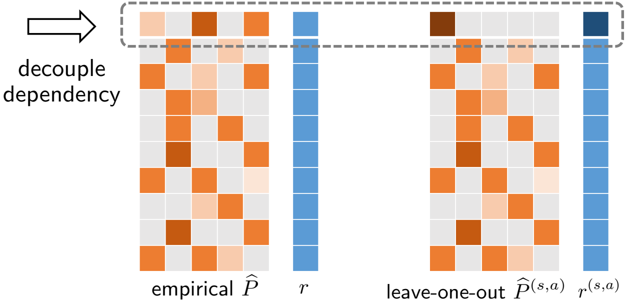
— *inspired by [Agarwal et al., 2019] but quite different ...*



- define $\widehat{\pi}^\star_{(s,a)} \xrightarrow{\text{empirical maximizer}} \big(\widehat{P}^{(s,a)}, r^{(s,a)}\big)$
  - decouple dependency by dropping randomness in $\widehat{P}(\cdot \mid s,a)$
  - scalar $r^{(s,a)}$ ensures $\widehat{Q}^\star$ and $\widehat{V}^\star$ unchanged

Key idea 2: decouple dependency for $\widehat{V}^{\widehat{\pi}^\star} - V^{\widehat{\pi}^\star}$

— *inspired by [Agarwal et al., 2019] but quite different ...*



- define $\widehat{\pi}^\star_{(s,a)} \xrightarrow{\text{empirical maximizer}} \big(\widehat{P}^{(s,a)}, r^{(s,a)}\big)$
- $\widehat{\pi}^\star_{(s,a)} = \widehat{\pi}^\star$ can be determined under separation condition

$$\forall s \in \mathcal{S}, \quad \widehat{Q}^\star(s, \widehat{\pi}^\star(s)) - \max_{a:\, a \neq \widehat{\pi}^\star(s)} \widehat{Q}^\star(s,a) > 0$$

Key idea 3: tie-breaking via perturbation

- How to ensure the optimal policy stand out from other policies?

$$\forall s \in \mathcal{S}, \quad \widehat{Q}^{\star}(s, \widehat{\pi}^{\star}(s)) - \max_{a: a \neq \widehat{\pi}^{\star}(s)} \widehat{Q}^{\star}(s, a) \geq \omega$$

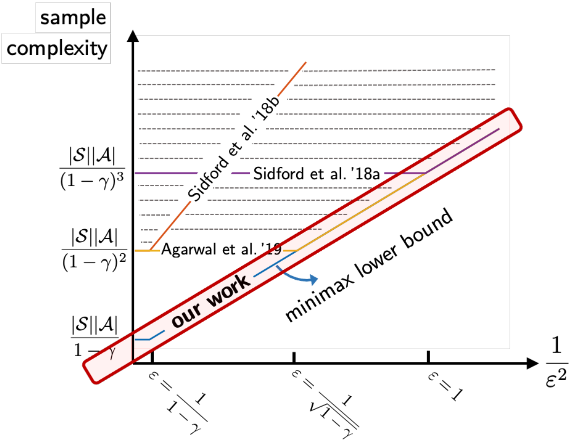
Key idea 3: tie-breaking via perturbation

- How to ensure the optimal policy stand out from other policies?

$$\forall s \in \mathcal{S}, \quad \widehat{Q}^{\star}(s, \widehat{\pi}^{\star}(s)) - \max_{a: a \neq \widehat{\pi}^{\star}(s)} \widehat{Q}^{\star}(s, a) \geq \omega$$

- **Solution:** slightly perturb rewards $r \implies \widehat{\pi}_{\mathrm{p}}^{\star}$
  - ensures the uniqueness of $\widehat{\pi}_{\mathrm{p}}^{\star}$
  - $V^{\widehat{\pi}_{\mathrm{p}}^{\star}} \approx V^{\widehat{\pi}^{\star}}$

Summary of model-based RL

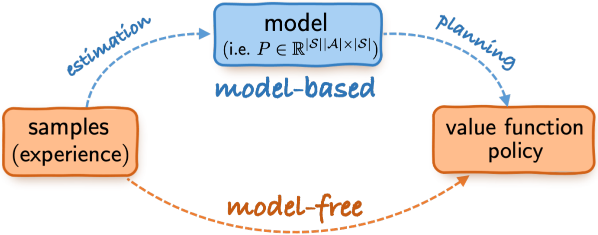


Model-based RL is minimax optimal & does not suffer from a sample size barrier!

Model-free / value-based RL

1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)

Model-based vs. model-free RL



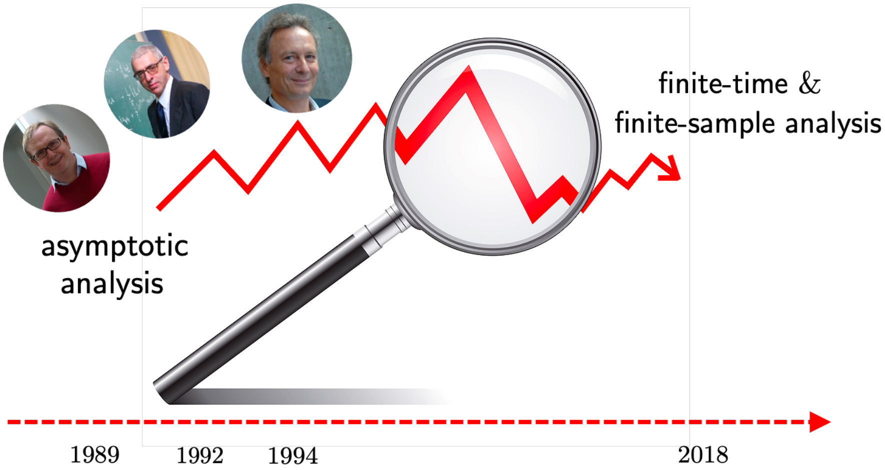
Model-based approach ("plug-in")

1. build empirical estimate $\widehat{P}$ for P
2. planning based on empirical $\widehat{P}$

Model-free / value-based approach

— learning w/o modeling & estimating environment explicitly

— memory-efficient, online, ...



Focus of this part: classical **Q-learning** algorithm and its variants

A starting point: Bellman optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s',a') \right]}_{\text{next state's value}}$$

- one-step look-ahead

A starting point: Bellman optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s', a')\right]}_{\text{next state's value}}$$

- one-step look-ahead

Bellman equation: $Q^\star$ is *unique* solution to

$$\mathcal{T}(Q^\star) = Q^\star$$

A starting point: Bellman optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \Big[\underbrace{\max_{a' \in \mathcal{A}} Q(s',a')}_{\text{next state's value}} \Big]$$

- one-step look-ahead

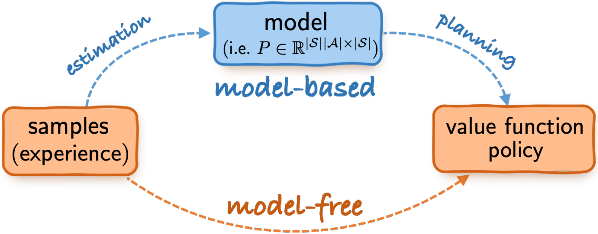
Bellman equation: $Q^\star$ is *unique* solution to

$$\mathcal{T}(Q^\star) = Q^\star$$

- **takeaway message:** it suffices to solve the Bellman equation
- **challenge:** how to solve it using stochastic samples?

Richard Bellman

Three approaches



Model-based approach ("plug-in")

- build an empirical estimate $\widehat{P}$ for P
- planning based on the empirical $\widehat{P}$

Value-based approach

— learning w/o estimating the model explicitly

Policy-based approach

— optimization in the space of policies

Value-based RL (a model-free approach)

Q-learning: a stochastic approximation algorithm

Chris Watkins

Peter Dayan

$\underbrace{\text{Stochastic approximation}}_{\text{Robbins \& Monro, 1951}}$ for solving the **Bellman equation**

$$\mathcal{T}(Q) - Q = 0$$

where

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s',a')\right]}_{\text{next state's value}}.$$

Q-learning: a stochastic approximation algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation $\mathcal{T}(Q) - Q = 0$

$$\underbrace{Q_{t+1}(s,a) = Q_t(s,a) + \eta_t\big(\mathcal{T}_t(Q_t)(s,a) - Q_t(s,a)\big)}_{\text{sample transition } (s,a,s')}, \quad t \geq 0$$

Q-learning: a stochastic approximation algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation $\mathcal{T}(Q) - Q = 0$

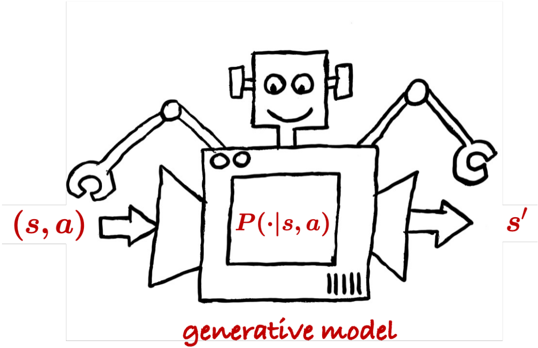
$$\underbrace{Q_{t+1}(s,a) = Q_t(s,a) + \eta_t\big(\mathcal{T}_t(Q_t)(s,a) - Q_t(s,a)\big)}_{\text{sample transition } (s,a,s')}, \quad t \geq 0$$

$$\mathcal{T}_t(Q)(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

$$\mathcal{T}(Q)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \big[\max_{a'} Q(s',a')\big]$$

A generative model / simulator

— *Kearns, Singh '99*



Each iteration, draw an independent sample (s, a, s') for given (s, a)

Synchronous Q-learning

Chris Watkins

Peter Dayan

for $t = 0, 1, \ldots, T$

  for each $(s,a) \in \mathcal{S} \times \mathcal{A}$

    draw a sample (s, a, s'), run

$$Q_{t+1}(s,a) = (1-\eta_t)Q_t(s,a) + \eta_t\Big\{r(s,a) + \gamma \max_{a'} Q_t(s',a')\Big\}$$

synchronous: all state-action pairs are updated simultaneously

- total sample size: $T|\mathcal{S}||\mathcal{A}|$

Sample complexity of synchronous Q-learning

Theorem (Li, Cai, Chen, Wei, Chi '21)

For any $0 < \varepsilon \leq 1$, synchronous Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ with high prob. and $\mathbb{E}[\|\widehat{Q} - Q^\star\|_\infty] \leq \varepsilon$, with sample size at most

$$\begin{cases} \widetilde{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\Big) & \text{if } |\mathcal{A}| \geq 2 \\ \widetilde{O}\Big(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\Big) & \text{if } |\mathcal{A}| = 1 \qquad (\textit{TD learning}) \end{cases}$$

Sample complexity of synchronous Q-learning

Theorem (Li, Cai, Chen, Wei, Chi '21)

For any $0 < \varepsilon \leq 1$*, synchronous Q-learning yields* $\|\widehat{Q} - Q^{\star}\|_{\infty} \leq \varepsilon$ *with high prob. and* $\mathbb{E}[\|\widehat{Q} - Q^{\star}\|_{\infty}] \leq \varepsilon$*, with sample size at most*

$$\begin{cases} \widetilde{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\Big) & \text{if } |\mathcal{A}| \geq 2 \\ \widetilde{O}\Big(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\Big) & \text{if } |\mathcal{A}| = 1 \qquad (\textit{TD learning}) \end{cases}$$

- Covers both *constant* and *rescaled linear* learning rates:

$$\eta_t \equiv \frac{1}{1 + \frac{c_1(1-\gamma)T}{\log^2 T}} \quad \text{or} \quad \eta_t = \frac{1}{1 + \frac{c_2(1-\gamma)t}{\log^2 T}}$$

Sample complexity of synchronous Q-learning

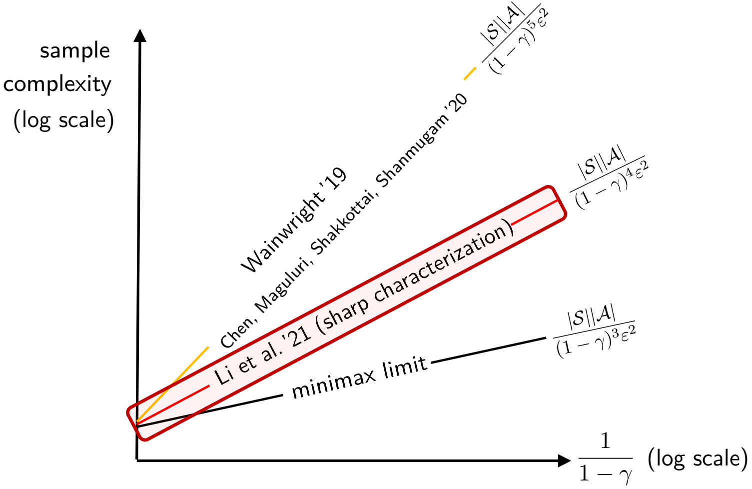
Theorem (Li, Cai, Chen, Wei, Chi '21)

For any $0 < \varepsilon \leq 1$, synchronous Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ with high prob. and $\mathbb{E}[\|\widehat{Q} - Q^\star\|_\infty] \leq \varepsilon$, with sample size at most

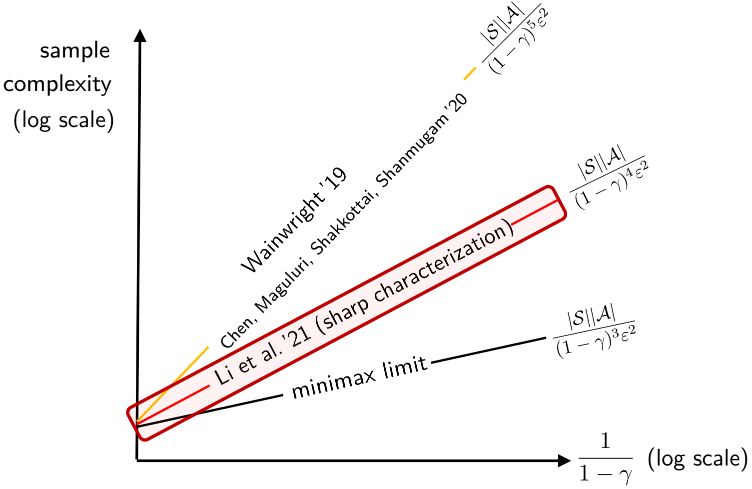
$$\begin{cases} \widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\right) & \text{if } |\mathcal{A}| \geq 2 \qquad \text{(?)} \\ \widetilde{O}\left(\frac{|\mathcal{S}|}{(1-\gamma)^3\varepsilon^2}\right) & \text{if } |\mathcal{A}| = 1 \qquad \text{(minimax optimal)} \end{cases}$$

other papers	sample complexity
Even-Dar & Mansour '03	$2^{\frac{1}{1-\gamma}} \frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^4\varepsilon^2}$
Beck & Srikant '12	$\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert^2}{(1-\gamma)^5\varepsilon^2}$
Wainwright '19	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\varepsilon^2}$
Chen, Maguluri, Shakkottai, Shanmugam '20	$\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\varepsilon^2}$

All this requires sample size at least $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ *(*$|\mathcal{A}| \geq 2$*) ...*



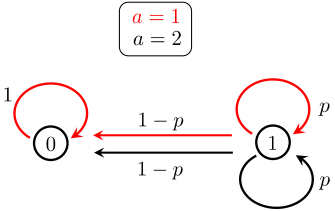
All this requires sample size at least $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ *(*$|\mathcal{A}| \geq 2$*) ...*



Question: *Is Q-learning sub-optimal, or is it an analysis artifact?*

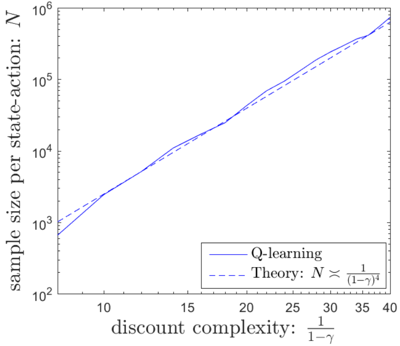
A numerical example: $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ samples seem necessary ...

— *observed in Wainwright '19*



$$p = \frac{4\gamma - 1}{3\gamma}$$

$$r(0,1) = 0, \quad r(1,1) = r(1,2) = 1$$



Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi, 2021)

For any $0 < \varepsilon \leq 1$, there exists an MDP with $|\mathcal{A}| \geq 2$ such that to achieve $\|\widehat{Q} - Q^{\star}\|_{\infty} \leq \varepsilon$, synchronous Q-learning needs at least

$$\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\right) \quad \textit{samples}$$

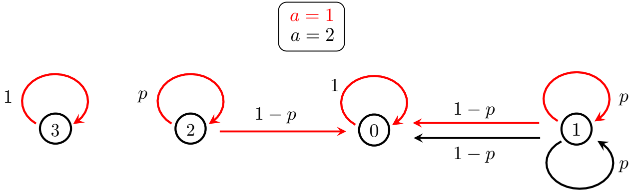
Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi, 2021)

For any $0 < \varepsilon \leq 1$, there exists an MDP with $|\mathcal{A}| \geq 2$ such that to achieve $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$, synchronous Q-learning needs at least

$$\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\right) \quad \textit{samples}$$

- Tight **algorithm-dependent** lower bound
- Holds for both constant and rescaled linear learning rates

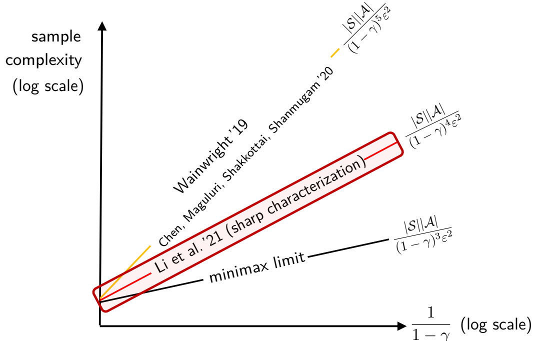


Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi, 2021)

For any $0 < \varepsilon \leq 1$, there exists an MDP with $|\mathcal{A}| \geq 2$ such that to achieve $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$, synchronous Q-learning needs at least

$$\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}\right) \quad \textit{samples}$$



Improving sample complexity via **variance reduction**

— *a powerful idea from finite-sum stochastic optimization*

Variance-reduced Q-learning updates (Wainwright '19)

— *inspired by SVRG (Johnson & Zhang '13)*

$$Q_t(s,a) = (1-\eta)Q_{t-1}(s,a) + \eta\Big(\mathcal{T}_t(Q_{t-1}) \underbrace{-\mathcal{T}_t(\overline{Q}) + \widetilde{\mathcal{T}}(\overline{Q})}_{\text{use } \overline{Q} \text{ to help reduce variability}}\Big)(s,a)$$

Variance-reduced Q-learning updates (Wainwright '19)

— *inspired by SVRG (Johnson & Zhang '13)*

$$Q_t(s,a) = (1-\eta)Q_{t-1}(s,a) + \eta\Big(\mathcal{T}_t(Q_{t-1}) \underbrace{-\mathcal{T}_t(\overline{Q}) + \widetilde{\mathcal{T}}(\overline{Q})}_{\text{use } \overline{Q} \text{ to help reduce variability}}\Big)(s,a)$$

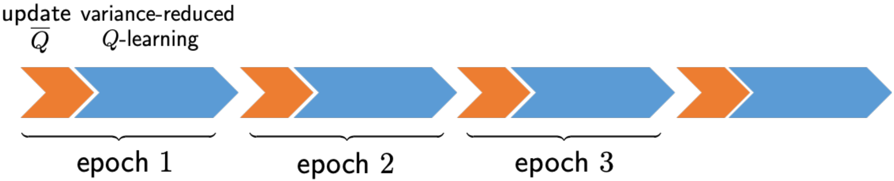
- $\overline{Q}$: some <u>reference</u> Q-estimate
- $\widetilde{\mathcal{T}}$: empirical Bellman operator (using a <u>*batch*</u> of samples)

$$\mathcal{T}_t(Q)(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

$$\widetilde{\mathcal{T}}(Q)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim \widetilde{P}(\cdot|s,a)} \big[\max_{a'} Q(s',a')\big]$$

An epoch-based stochastic algorithm

— *inspired by Johnson & Zhang '13*



for each epoch

1. update $\overline{Q}$ and $\widetilde{\mathcal{T}}(\overline{Q})$ (which stay fixed in the rest of the epoch)
2. run variance-reduced Q-learning updates iteratively

Sample complexity of variance-reduced Q-learning

Theorem (Wainwright '19)

For any $0 < \varepsilon \le 1$, sample complexity for **variance-reduced synchronous Q-learning** *to yield $\|\widehat{Q} - Q^{\star}\|_{\infty} \le \varepsilon$ is at most*

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

- allows for more aggressive learning rates

Sample complexity of variance-reduced Q-learning

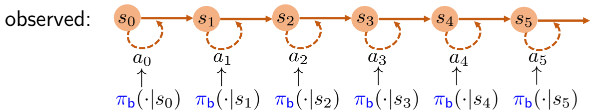
Theorem (Wainwright '19)

For any $0 < \varepsilon \leq 1$, *sample complexity for* **variance-reduced synchronous Q-learning** *to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *is at most*

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

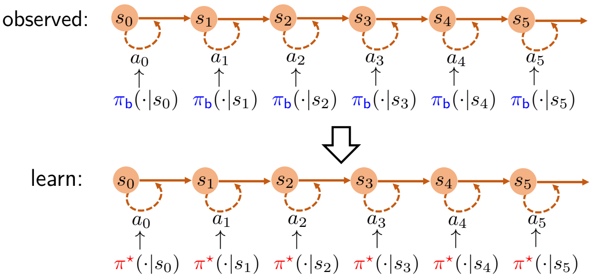
- allows for more aggressive learning rates
- minimax-optimal for $0 < \varepsilon \leq 1$
  - remains suboptimal if $1 < \varepsilon < \frac{1}{1-\gamma}$

Markovian samples and behavior policy



Observed: $\underbrace{\{s_t, a_t, r_t\}_{t\geq 0}}_{\text{Markovian trajectory}}$ induced by behavior policy π_b

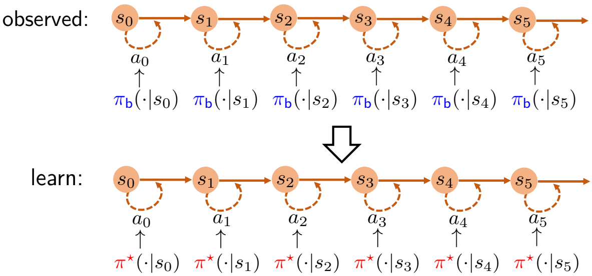
Markovian samples and behavior policy



Observed: $\underbrace{\{s_t, a_t, r_t\}_{t\geq 0}}_{\text{Markovian trajectory}}$ induced by behavior policy π_b

Goal: learn optimal value $V^\star$ and $Q^\star$ based on sample trajectory

Markovian samples and behavior policy



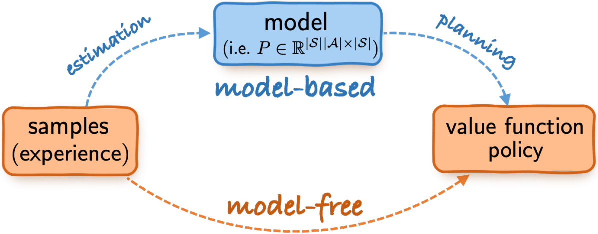
Key quantities of sample trajectory

- minimum state-action occupancy probability

$$\mu_{\mathsf{min}} := \min \underbrace{\mu_{\pi_{\mathsf{b}}}(s,a)}_{\text{stationary distribution}}$$

- mixing time: t_{mix}

Model-based vs. model-free RL



Model-free approach (e.g. Q-learning)

— learning w/o modeling & estimating environment explicitly

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

$\underbrace{\text{Stochastic approximation}}_{\text{Robbins \& Monro '51}}$ for solving **Bellman equation** $Q = \mathcal{T}(Q)$

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation $Q = \mathcal{T}(Q)$

$$\underbrace{Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \eta_t(\mathcal{T}_t(Q_t)(s_t, a_t) - Q_t(s_t, a_t))}_{\textit{only} \text{ update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation $Q = \mathcal{T}(Q)$

$$\underbrace{Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \eta_t(\mathcal{T}_t(Q_t)(s_t, a_t) - Q_t(s_t, a_t))}_{\textit{only}\text{ update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

$$\mathcal{T}_t(Q)(s_t, a_t) := r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

$$\mathcal{T}(Q)(s, a) = r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{a'} Q(s', a') \right]$$

Q-learning: a classical model-free algorithm

Chris Watkins

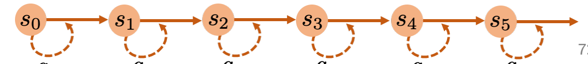
Peter Dayan

Stochastic approximation for solving Bellman equation $Q = \mathcal{T}(Q)$

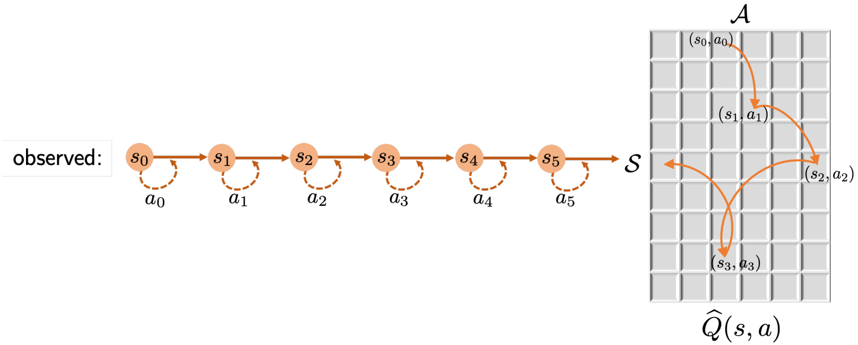
$$\underbrace{Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \eta_t(\mathcal{T}_t(Q_t)(s_t, a_t) - Q_t(s_t, a_t)),}_{\textit{only} \text{ update } (s_t, a_t)\text{-th entry}} \quad t \geq 0$$

— **asynchronous:** only a single entry is updated each iteration
(resembles Markov-chain *coordinate descent*)

observed:

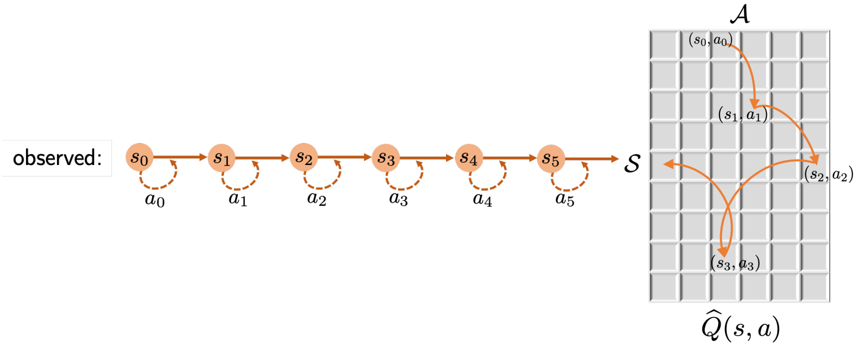


Q-learning on Markovian samples



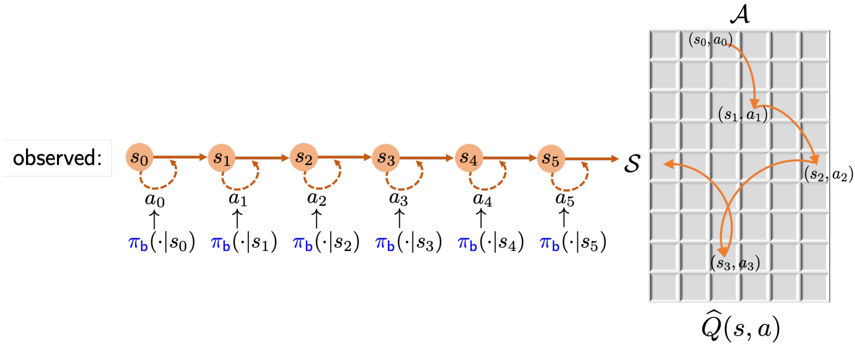
- **asynchronous:** only a single entry is updated each iteration

Q-learning on Markovian samples



- **asynchronous:** only a single entry is updated each iteration
  - resembles Markov-chain *coordinate descent*

Q-learning on Markovian samples



- **asynchronous:** only a single entry is updated each iteration
  - resembles Markov-chain *coordinate descent*
- **off-policy:** target policy $\pi^\star \neq$ behavior policy π_b

What is sample complexity of (async) Q-learning?

A highly incomplete list of works

- Watkins, Dayan ’92
- Tsitsiklis ’94
- Jaakkola, Jordan, Singh ’94
- Szepesvári ’98
- Borkar, Meyn ’00
- Even-Dar, Mansour ’03
- Beck, Srikant ’12
- Chi, Zhu, Bubeck, Jordan ’18
- Lee, He ’18
- Chen, Zhang, Doan, Maguluri, Clarke ’19
- Du, Lee, Mahajan, Wang ’20
- Chen, Maguluri, Shakkottai, Shanmugam ’20
- Qu, Wierman ’20
- Devraj, Meyn ’20
- Weng, Gupta, He, Ying, Srikant ’20
- Li, Wei, Chi, Gu, Chen ’20
- Li, Cai, Chen, Wei, Chi ’21
- Chen, Maguluri, Shakkottai, Shanmugam ’21
- ...

Prior art: async Q-learning

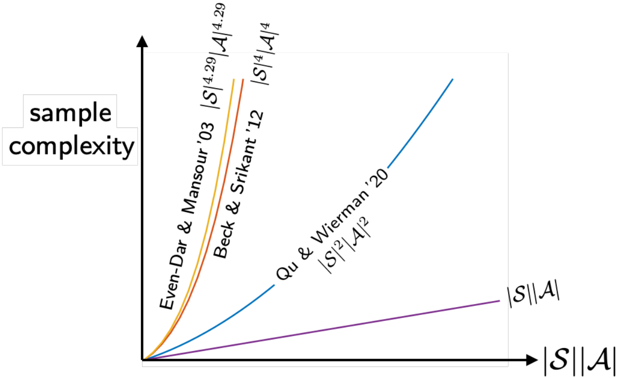
Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$?

other papers	sample complexity
Even-Dar, Mansour '03	$\frac{(t_{\text{cover}})^{\frac{1}{1-\gamma}}}{(1-\gamma)^4\varepsilon^2}$
Even-Dar, Mansour '03	$\big(\frac{t_{\text{cover}}^{1+3\omega}}{(1-\gamma)^4\varepsilon^2}\big)^{\frac{1}{\omega}} + \big(\frac{t_{\text{cover}}}{1-\gamma}\big)^{\frac{1}{1-\omega}}$, $\omega \in (\frac{1}{2}, 1)$
Beck & Srikant '12	$\frac{t_{\text{cover}}^3 \lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^5\varepsilon^2}$
Qu & Wierman '20	$\frac{t_{\text{mix}}}{\mu_{\text{min}}^2(1-\gamma)^5\varepsilon^2}$
Li, Wei, Chi, Gu, Chen '20	$\frac{1}{\mu_{\text{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\text{mix}}}{\mu_{\text{min}}(1-\gamma)}$
Chen, Maguluri, Shakkottai, Shanmugam '21	$\frac{1}{\mu_{\text{min}}^3(1-\gamma)^5\varepsilon^2}$ + other-term(t_{mix})

— cover time: $t_{\text{cover}} \asymp \frac{t_{\text{mix}}}{\mu_{\text{min}}}$

Prior art: async Q-learning

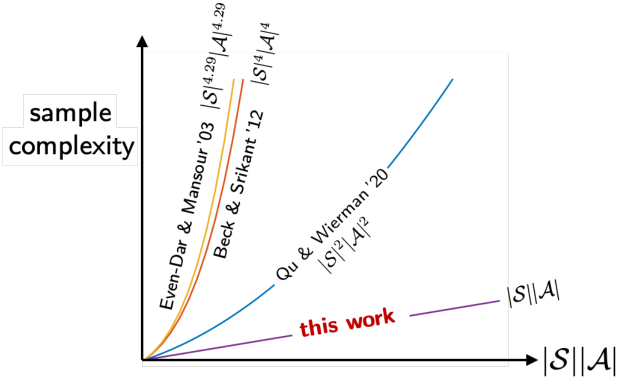
Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?



if we take $\mu_{\mathsf{min}} \asymp \frac{1}{|\mathcal{S}||\mathcal{A}|}$, $t_{\mathsf{cover}} \asymp \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}}$

Prior art: async Q-learning

Question: how many samples are needed to ensure $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$?



if we take $\mu_{\mathsf{min}} \asymp \frac{1}{|\mathcal{S}||\mathcal{A}|}$, $t_{\mathsf{cover}} \asymp \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}}$

All prior results require sample size of at least $t_{\mathsf{mix}}|\mathcal{S}|^2|\mathcal{A}|^2$!

Main result: ℓ_∞-based sample complexity

Theorem (Li, Wei, Chi, Gu, Chen '20)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$*, sample complexity of async Q-learning to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *is at most (up to some log factor)*

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

Main result: ℓ_∞-based sample complexity

Theorem (Li, Wei, Chi, Gu, Chen '20)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, sample complexity of async Q-learning to yield $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ is at most (up to some log factor)

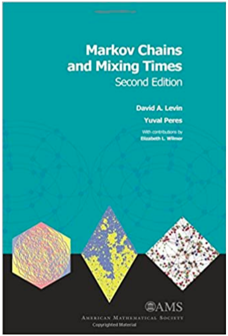
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

— *prior art:* $\frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}^2(1-\gamma)^5\varepsilon^2}$ (Qu & Wierman'20)

- Improves upon prior art by **at least** $|\mathcal{S}||\mathcal{A}|$!

Effect of mixing time on sample complexity

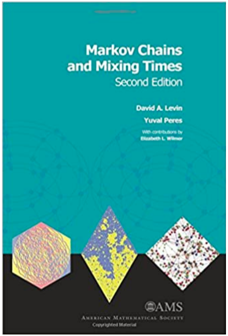
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$



- reflects cost taken to reach steady state
- one-time expense (almost independent of ε)
  — it becomes amortized as algorithm runs

Effect of mixing time on sample complexity

$$\frac{1}{\mu_{\text{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\text{mix}}}{\mu_{\text{min}}(1-\gamma)}$$



- reflects cost taken to reach steady state
- one-time expense (almost independent of ε)
  — it becomes amortized as algorithm runs

— *prior art:* $\frac{t_{\text{mix}}}{\mu_{\text{min}}^2(1-\gamma)^5\varepsilon^2}$ [Qu & Wierman 20]

Dependence on effective horizon

minimax lower bound
(Azar et al. '13)

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^3\varepsilon^2}$$

asyn Q-learning
(ignoring dependency on t_{mix})

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2}$$

Dependence on effective horizon

minimax lower bound
(Azar et al. '13)

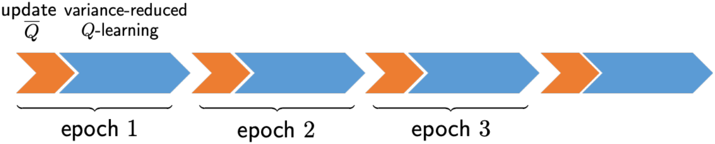
$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2}$$

asyn Q-learning
(ignoring dependency on t_{mix})

$$\frac{1}{\mu_{\min}(1-\gamma)^5\varepsilon^2}$$

The dependency on $\frac{1}{1-\gamma}$ can be tightened by *variance reduction*.

— *inspired by [Johnson & Zhang, 2013], [Wainwright, 2019]*



Sample complexity for variance-reduced Q-learning

Theorem (Li, Wei, Chi, Gu, Chen '20)

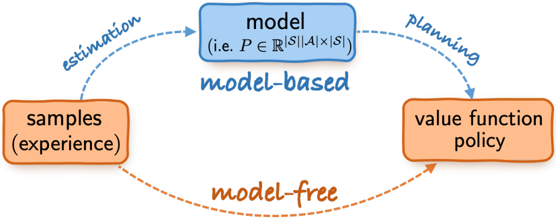
For any $0 < \varepsilon \leq 1$, sample complexity for **(async) variance-reduced Q-learning** *to yield $\|\widehat{Q} - Q^{\star}\|_{\infty} \leq \varepsilon$ is at most on the order of*

$$\frac{1}{\mu_{\min}(1-\gamma)^3\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\min}(1-\gamma)}$$

- more aggressive learning rates: $\eta_t \equiv \min\left\{\frac{\cancel{(1-\gamma)^4}(1-\gamma)^2}{\gamma^2}, \frac{1}{t_{\mathsf{mix}}}\right\}$
- minimax-optimal for $0 < \varepsilon \leq 1$

Summary of this part

- basics of MDP and DP algorithms
- break the sample size barrier using model-based approach
- obtain tight sample complexity for Q-learning



Key references for this part

Papers:

- "Model-based reinforcement learning with a generative model is minimax optimal," A Agarwal, S Kakade, L Yang, *Conference on Learning Theory (COLT)'20*
- "Breaking the sample size barrier in model-based reinforcement learning with a generative model," G Li, Y Wei, Y Chi, Y Chen, *NeurIPS'20, Operators Research'23*
- "Is Q-learning minimax optimal? a tight sample complexity analysis," G Li, C Cai, Y Chen, Y Wei, Y Chi, *Operations Research'23*
- "Finite-time analysis of asynchronous stochastic approximation and Q-learning." G Qu, A Wierman, *Conference on Learning Theory (COLT)'20*
- "Variance-reduced Q-learning is minimax optimal," M Wainwright'19.
- "Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction, " G Li, Y Wei, Y Chi, Y Gu, Y Chen, *IEEE Transactions on Information Theory'21*

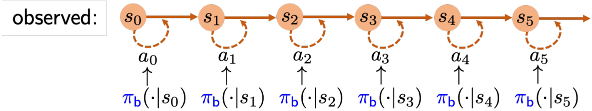
Statistical and Algorithmic Foundations of Reinforcement Learning (Part 2)

Yuxin Chen

Wharton Statistics & Data Science, JSM 2023

1. Online RL
2. Offline RL
3. Reward-agnostic exploration
4. Hybrid RL (policy finetuning)

Recap: Q-learning following a behavior policy



To achieve $\|Q_T - Q^\star\|_\infty \le \varepsilon$, needs a sample size (Li et al. '23)

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

- $\mu_{\mathsf{min}} \coloneqq \min \underbrace{\mu_{\pi_b}(s,a)}_{\text{stationary distribution}}$: min state-action occupancy prob.
- t_{mix}: mixing time under behavior policy π_b

Limitations

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

μ_{min} need to be positive $\implies$ π_{b} covers entire state-action space

Limitations

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

μ_{min} need to be positive $\Longrightarrow$ π_{b} covers entire state-action space

- π_{b} must be randomized
- can we find such π_{b} for all MDPs?
- μ_{min} might be exponentially small $\Longrightarrow$ need enormous samples!

Limitations

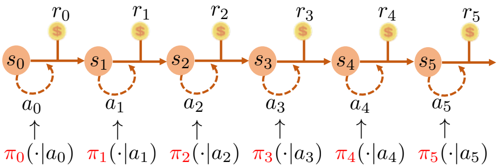
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$$

μ_{min} need to be positive $\implies$ π_{b} covers entire state-action space

- π_{b} must be randomized
- can we find such π_{b} for all MDPs?
- μ_{min} might be exponentially small $\implies$ need enormous samples!

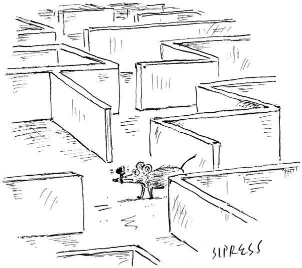
Can exploration help mitigate this issue?

Online RL: interacting with real environment



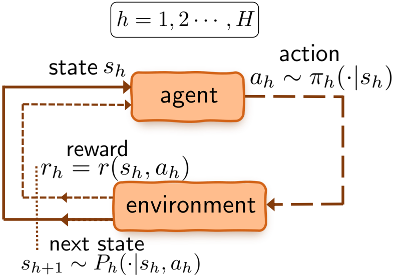
exploration via adaptive policies

- trial-and-error
- sequential and online
- adaptive learning from data



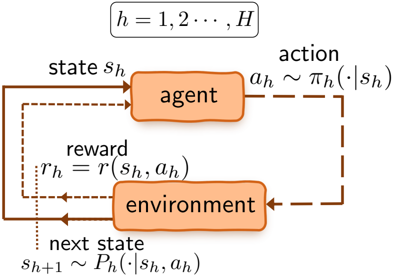
"Recalculating ... recalculating ..."

Recap: finite-horizon Markov decision process



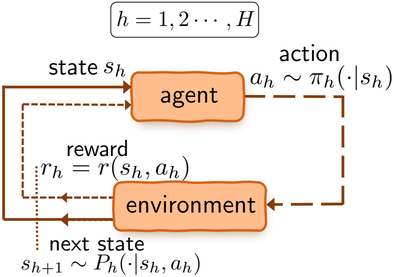
- H: horizon length

Recap: finite-horizon Markov decision process



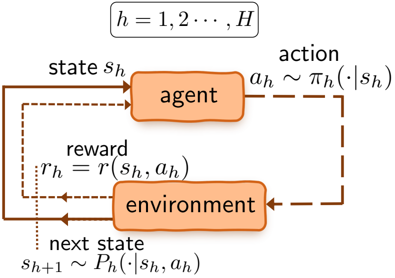
- H: horizon length
- $\mathcal{S}$: state space with size S
- $\mathcal{A}$: action space with size A

Recap: finite-horizon Markov decision process



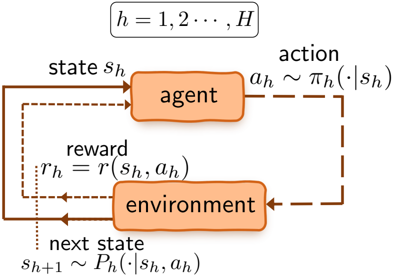
- H: horizon length
- $\mathcal{S}$: state space with size S
- $\mathcal{A}$: action space with size A
- $r_h(s_h, a_h) \in [0, 1]$: immediate reward in step h

Recap: finite-horizon Markov decision process



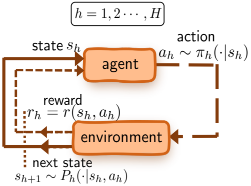
- H: horizon length
- $\mathcal{S}$: state space with size S
- $\mathcal{A}$: action space with size A
- $r_h(s_h, a_h) \in [0, 1]$: immediate reward in step h
- $\pi = \{\pi_h\}_{h=1}^{H}$: policy (or action selection rule)

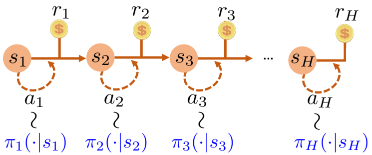
Recap: finite-horizon Markov decision process



- 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

Recap: value function and Q-function of policy π



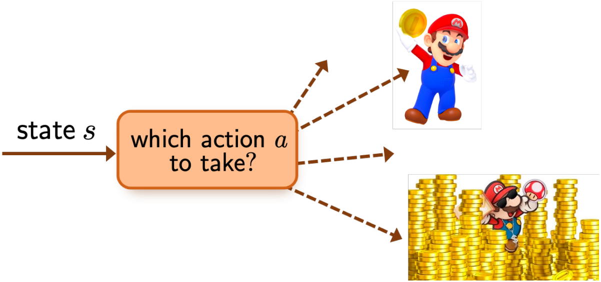


$$V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^{H} r_h(s_h, a_h) \,\Big|\, s_h = s\right]$$

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

- execute policy π to generate sample trajectory

Recap: optimal policy and optimal values

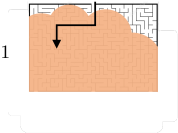


- **Optimal policy** $\pi^\star$: maximizing the value function
- Optimal values: $V^\star := V^{\pi^\star}$, $Q^\star := Q^{\pi^\star}$

Online episodic RL

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

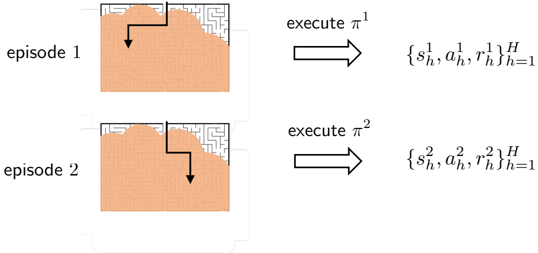
episode 1



execute π^1 $\Longrightarrow$ $\{s_h^1, a_h^1, r_h^1\}_{h=1}^H$

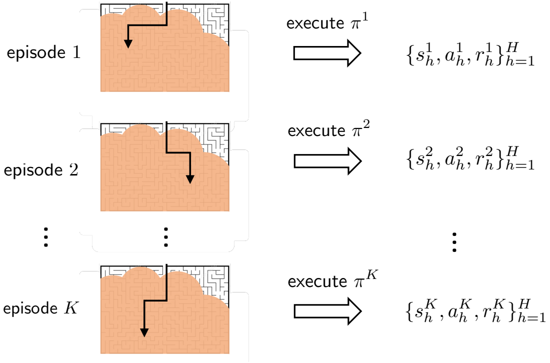
Online episodic RL

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



Online episodic RL

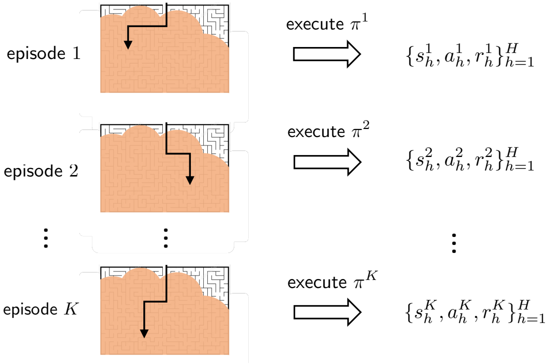
Sequentially execute MDP for K episodes, each consisting of H steps



Online episodic RL

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

— *sample size:* $T = KH$



exploration (exploring unknowns) vs. **exploitation** (exploiting learned info)

Regret: gap between learned policy & optimal policy

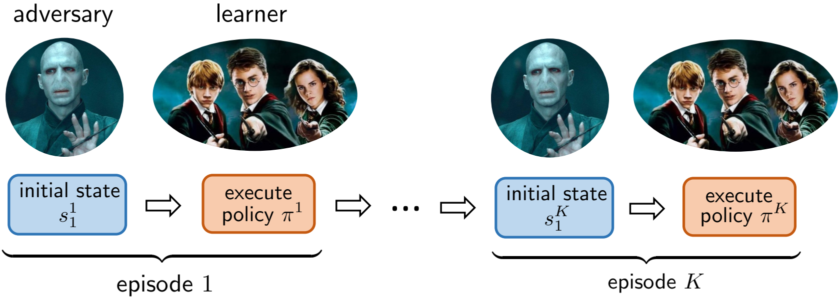
adversary

learner

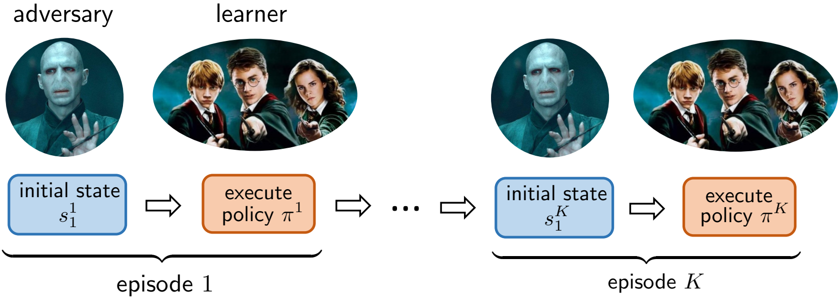
initial state s_1^1 $\Rightarrow$ execute policy π^1

episode 1

Regret: gap between learned policy & optimal policy



Regret: gap between learned policy & optimal policy



Performance metric: given $\underbrace{\text{initial states } \{s_1^k\}_{k=1}^K}_{\text{chosen by nature/adversary}}$, define

$$\mathsf{Regret}(T) \quad := \quad \sum_{k=1}^{K} \Big(V_1^\star(s_1^k) - V_1^{\pi^k}(s_1^k) \Big)$$

Regret lower bounds

Theorem 1 (Domingues et al. '21)

Consider any $T \geq H^2SA$. For any algorithm, there exists an episodic nonstationary MDP $\mathcal{M}_\pi$ such that

$$\mathbb{E}[\textit{Regret}(T)] \geq \frac{1}{48\sqrt{6}}\sqrt{H^2SAT}$$

- Ignoring other factors, the regret is at least on the order of

$$\sqrt{T}$$

- The lower bound also reflects impacts of horizon H and size of state-action space SA

Lower bound

(Domingues et al. '21)

$$\text{Regret}(T) \gtrsim \sqrt{H^2SAT}$$

Existing algorithms

- UCB-VI: Azar et al. '17
- UBEV: Dann et al. '17
- UCB-Q-Hoeffding: Jin et al. '18
- UCB-Q-Bernstein: Jin et al. '18
- UCB2-Q-Bernstein: Bai et al. '19
- EULER: Zanette et al. '19
- UCB-Q-Advantage: Zhang et al. '20
- MVP: Zhang et al. '20
- UCB-M-Q: Menard et al. '21
- Q-EarlySettled-Advantage: Li et al. '21
- (modified) MVP: Zhang et al. '23

Lower bound

(Domingues et al. '21)

$$\text{Regret}(T) \gtrsim \sqrt{H^2SAT}$$

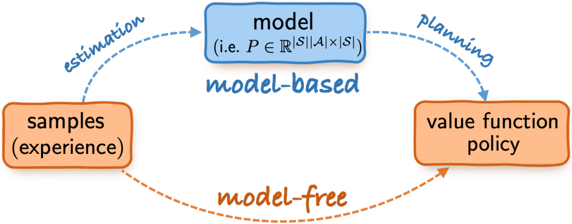
Existing algorithms

- UCB-VI: Azar et al. '17
- UBEV: Dann et al. '17
- UCB-Q-Hoeffding: Jin et al. '18
- UCB-Q-Bernstein: Jin et al. '18
- UCB2-Q-Bernstein: Bai et al. '19
- EULER: Zanette et al. '19
- UCB-Q-Advantage: Zhang et al. '20
- MVP: Zhang et al. '20
- UCB-M-Q: Menard et al. '21
- Q-EarlySettled-Advantage: Li et al. '21
- (modified) MVP: Zhang et al. '23

Which online RL algorithms achieve near-minimal regret?

Model-based online RL with UCB exploration

Model-based vs. model-free approaches



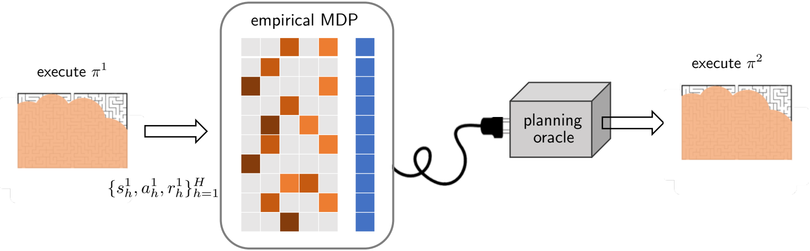
Model-based approach ("plug-in")

1. build an empirical estimate $\widehat{P}$ for P
2. planning based on the empirical $\widehat{P}$

Model-free approach

— learning w/o estimating the model explicitly

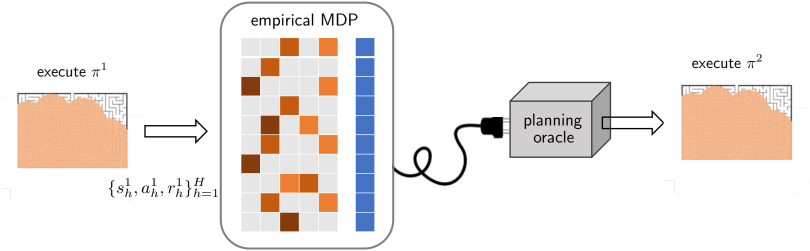
Online RL with the model-based approach



repeat:

- use **all** previous data to estimate transition probabilities
- apply planning (e.g., value iteration) to the estimated model to learn an updated policy for the next episode

Online RL with the model-based approach



repeat:

- use **all** previous data to estimate transition probabilities
- apply planning (e.g., value iteration) to the estimated model to learn an updated policy for the next episode

How to balance exploration and exploitation in this framework?

T. L. Lai

H. Robbins

Optimism in the face of uncertainty:

- explore based on the best possible values (i.e., optimistic estimates) associated with the actions!
- a common framework based on $\underbrace{\text{upper confidence bounds (UCB)}}_{\text{accounts for estimates + uncertainty level}}$

Example: UCB algorithm for multi-arm bandits

— *Auer et al. '02*

Idea: always try the best arm, where "best" includes exploration & exploitation

Example: UCB algorithm for multi-arm bandits

— *Auer et al. '02*

Idea: always try the best arm, where "best" includes exploration & exploitation

In each round t:

- calculate UCB index for each arm i:

$$\text{UCB}_{i,t} = \overline{r}_{i,t} + \sqrt{\frac{\log t}{N_{i,t}}}$$

  - $\overline{r}_{i,t}$: empirical average of reward for arm i
  - $N_{i,t}$: number of times arm i has been played up to round t

Example: UCB algorithm for multi-arm bandits

— *Auer et al. '02*

Idea: always try the best arm, where "best" includes exploration & exploitation

In each round t:

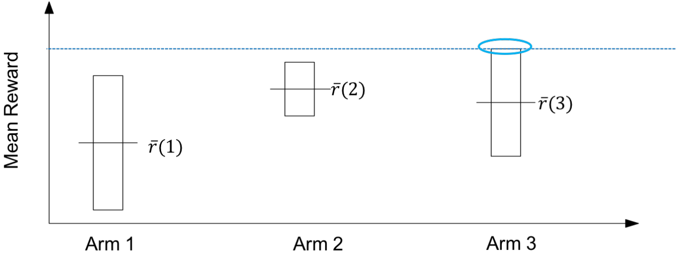
- calculate UCB index for each arm i:

$$\text{UCB}_{i,t} = \overline{r}_{i,t} + \sqrt{\frac{\log t}{N_{i,t}}}$$

  - $\overline{r}_{i,t}$: empirical average of reward for arm i
  - $N_{i,t}$: number of times arm i has been played up to round t

- play the arm with highest UCB index

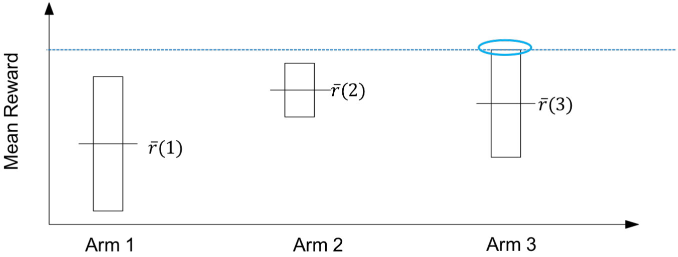
Understanding UCB



$$\text{UCB}_{i,t} = \overline{r}_{i,t} + \sqrt{\frac{\log t}{N_{i,t}}}$$

- **exploitation:** $\overline{r}_{i,t}$ is the average observed reward. High observed rewards of an arm leads to high UCB index

Understanding UCB



$$\mathrm{UCB}_{i,t} = \overline{r}_{i,t} + \sqrt{\frac{\log t}{N_{i,t}}}$$

- **exploitation:** $\overline{r}_{i,t}$ is the average observed reward. High observed rewards of an arm leads to high UCB index
- **exploration:** $\sqrt{\frac{\log t}{N_{i,t}}}$ decreases as we make more observations. Fewer observations of an arm leads to higher UCB index

UCB-VI (Azar et al. '17)

Idea: incorporate the upper confidence bound (UCB) framework into a model-based algorithm (i.e., value iteration (VI)) . . .

UCB-VI (Azar et al. '17)

Original VI: for $h = H, H-1, \ldots, 1$:

$$Q_h(s,a) \leftarrow \underbrace{r_h(s,a)}_{\text{immediate reward}} + \underbrace{\widehat{P}_{h,s,a} V_{h+1}}_{\text{next step's value}}$$

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

where $\widehat{P}_{h,s,a}$: empirical estimate of $P_{h,s,a}$

UCB-VI (Azar et al. '17)

Original VI: for $h = H, H-1, \ldots, 1$:

$$Q_h(s,a) \leftarrow \underbrace{r_h(s,a)}_{\text{immediate reward}} + \underbrace{\widehat{P}_{h,s,a} V_{h+1}}_{\text{next step's value}}$$

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

where $\widehat{P}_{h,s,a}$: empirical estimate of $P_{h,s,a}$

- pure exploitation; no exploration
- *to encourage exploration, why don't we replace* $Q_h(s,a)$ *w/ its UCB?*

UCB-VI (Azar et al. '17)

Uncertainty quantification in the next-step value $\widehat{P}_{h,s,a}V_{h+1}$: by Hoeffding's inequality $\&$ union bound, with prob. at least $1-\delta$,

$$\left\|(\widehat{P}_{h,s,a} - P_{h,s,a})V^{\star}_{h+1}\right\|_{\infty} \leq \widetilde{O}\left(\sqrt{\frac{H^2}{N_h(s,a)}}\right)$$

where $N_h(s,a)$: number of visits to (s,a) at step h

UCB-VI (Azar et al. '17)

Uncertainty quantification in the next-step value $\widehat{P}_{h,s,a}V_{h+1}$: by Hoeffding's inequality & union bound, with prob. at least $1-\delta$,

$$\left\|(\widehat{P}_{h,s,a} - P_{h,s,a})V^{\star}_{h+1}\right\|_{\infty} \le \widetilde{O}\Big(\sqrt{\frac{H^2}{N_h(s,a)}}\Big)$$

where $N_h(s,a)$: number of visits to (s,a) at step h

Optimistic VI: run VI using rewards $\{r_h(s,a) + b_h(s,a)\}$

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

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

where $b_h(s,a) = \widetilde{\Theta}\Big(\sqrt{\frac{H^2}{N_h(s,a)}}\Big)$

UCB-VI: algorithm

For each episode k:

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

$$Q_h(s_h, a_h) \leftarrow \min\left\{r_h(s_h, a_h) + \widehat{P}_{h,s_h,a_h} V_{h+1} + b_h(s_h, a_h),\ H - h + 1\right\}$$
$$V_h(s_h) \leftarrow \max_{a \in \mathcal{A}} Q_h(s_h, a)$$

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

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

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

Optimism in the face of uncertainty

Lemma 2

With prob. at least $1-\delta$, one has

$$Q_h(s,a) \geq Q_h^\star(s,a), \qquad V_h(s) \geq V_h^\star(s)$$

for all (h,s,a) in all episodes

optimism in the face of uncertainty:

- act according to $\underbrace{Q_h(s,a)}_{\text{an upper bound on } Q_h^\star(s,a)}$

Regret bound for UCB-VI (Azar et al. '17)

Theorem 3 (Azar et al. '17)

With prob. at least $1-\delta$, UCB-VI with Hoeffding bonus achieves

$$\mathit{Regret}(T) \lesssim \sqrt{H^3 SAT\iota} + H^3 S^2 A\iota^3$$

where $\iota = \log(HSAT/\delta)$

Regret bound for UCB-VI (Azar et al. '17)

Theorem 3 (Azar et al. '17)

With prob. at least $1-\delta$*, UCB-VI with Hoeffding bonus achieves*

$$\textit{Regret}(T) \lesssim \sqrt{H^3 SAT\iota} + H^3 S^2 A \iota^3$$

where $\iota = \log(HSAT/\delta)$

- Regret bound scales as

$$\sqrt{H^3 SAT} \qquad \text{as soon as} \qquad T \gtrsim \underbrace{H^3 S^3 A}_{\text{burn-in cost}}$$

which is sub-optimal by a factor of $\sqrt{H}$

Regret bound for UCB-VI (Azar et al. '17)

Theorem 3 (Azar et al. '17)

With prob. at least $1-\delta$, UCB-VI with Hoeffding bonus achieves

$$\textit{Regret}(T) \lesssim \sqrt{H^3 SAT\iota} + H^3 S^2 A \iota^3$$

where $\iota = \log(HSAT/\delta)$

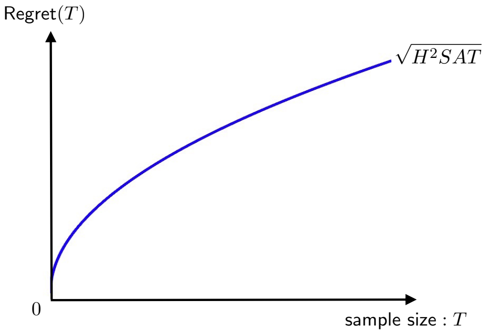
- Regret bound scales as

$$\sqrt{H^3 SAT} \qquad \text{as soon as} \qquad T \gtrsim \underbrace{H^3 S^3 A}_{\text{burn-in cost}}$$

  which is sub-optimal by a factor of $\sqrt{H}$
- Tighter bonus (e.g., Bernstein-style) leads to improved regret

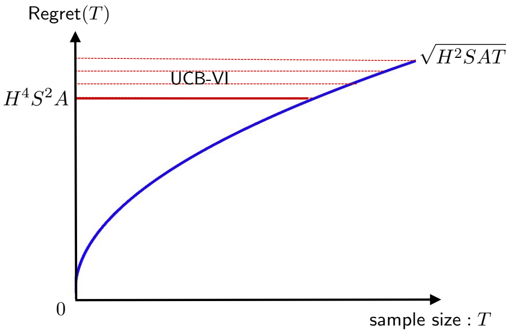
Asymptotically optimal regret

Using tighter variance-aware concentration, Azar et al. '17 developed the first method that is *asymptotically* regret-optimal



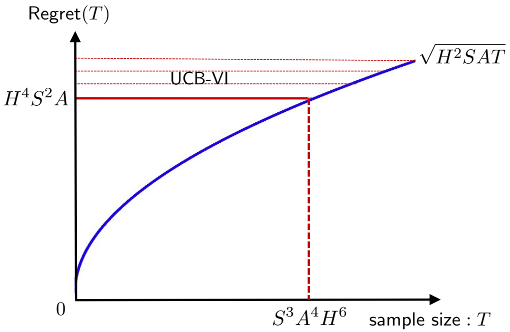
Asymptotically optimal regret

Using tighter variance-aware concentration, Azar et al. '17 developed the first method that is *asymptotically* regret-optimal



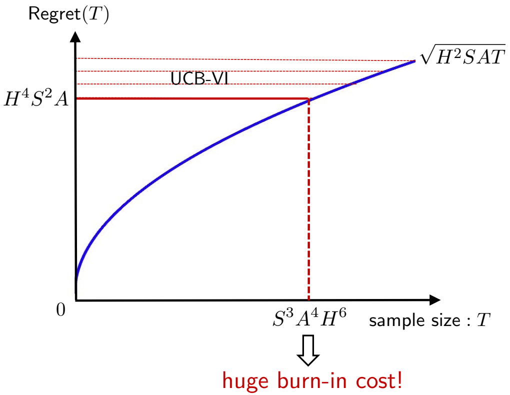
Asymptotically optimal regret

Using tighter variance-aware concentration, Azar et al. '17 developed the first method that is *asymptotically* regret-optimal



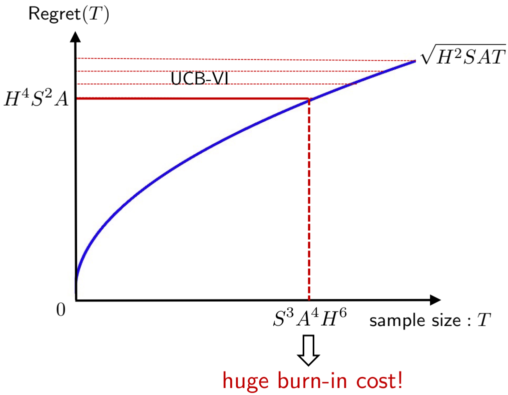
Asymptotically optimal regret

Using tighter variance-aware concentration, Azar et al. '17 developed the first method that is *asymptotically* regret-optimal



Asymptotically optimal regret

Using tighter variance-aware concentration, Azar et al. '17 developed the first method that is *asymptotically* regret-optimal



Issues: (1) large burn-in cost; (2) large memory complexity

model-based: S^2AH

Other asymptotically regret-optimal algorithms

Algorithm	Regret upper bound	Range of K that attains optimal regret
`UCBVI` (Azar et al. 17)	$\sqrt{SAH^2T} + S^2AH^3$	$[S^3AH^3, \infty)$
`ORLC` (Dann et al. '19)	$\sqrt{SAH^2T} + S^2AH^4$	$[S^3AH^5, \infty)$
`EULER` (Zanette et al. '19)	$\sqrt{SAH^2T} + S^{3/2}AH^3(\sqrt{S} + \sqrt{H})$	$[S^2AH^3(\sqrt{S} + \sqrt{H}), \infty)$
`UCB-Adv` (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2A^{3/2}H^{33/4}K^{1/4}$	$[S^6A^4H^{27}, \infty)$
`MVP` (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2AH^2$	$[S^3AH, \infty)$
`UCB-M-Q` (Menard et al. '21)	$\sqrt{SAH^2T} + SAH^4$	$[SAH^5, \infty)$
`Q-Earlysettled-Adv` (Li et al. '21)	$\sqrt{SAH^2T} + SAH^6$	$[SAH^9, \infty)$

Other asymptotically regret-optimal algorithms

Algorithm	Regret upper bound	Range of K that attains optimal regret
UCBVI (Azar et al. 17)	$\sqrt{SAH^2T} + S^2AH^3$	$[S^3AH^3, \infty)$
ORLC (Dann et al. '19)	$\sqrt{SAH^2T} + S^2AH^4$	$[S^3AH^5, \infty)$
EULER (Zanette et al. '19)	$\sqrt{SAH^2T} + S^{3/2}AH^3(\sqrt{S} + \sqrt{H})$	$[S^2AH^3(\sqrt{S} + \sqrt{H}), \infty)$
UCB-Adv (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2A^{3/2}H^{33/4}K^{1/4}$	$[S^6A^4H^{27}, \infty)$
MVP (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2AH^2$	$[S^3AH, \infty)$
UCB-M-Q (Menard et al. '21)	$\sqrt{SAH^2T} + SAH^4$	$[SAH^5, \infty)$
Q-Earlysettled-Adv (Li et al. '21)	$\sqrt{SAH^2T} + SAH^6$	$[SAH^9, \infty)$

Can we find a regre-optimal algorithm with no burn-in cost?

Regret-optimal algorithm w/o burn-in cost

Theorem 4 (Zhang, Chen, Lee, Du '23)

With prob. at least $1-\delta$, there is a model-based algorithm achieving

$$\textit{Regret}(T) \lesssim \widetilde{O}(\sqrt{H^2SAT})$$

- **algorithm:** Monotonic Value Propagation (MVP)
- the only algorithm so far that is regret-optimal w/o burn-ins
- key innovation: decoupling statistical dependency

Comparison with prior art

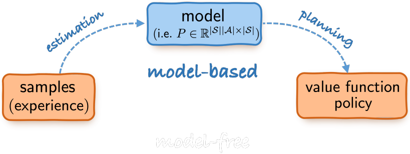
Algorithm	Regret upper bound	Range of K that attains optimal regret
`UCBVI` (Azar et al. 17)	$\sqrt{SAH^2T} + S^2AH^3$	$[S^3AH^3, \infty)$
`UCB-Adv` (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2A^{3/2}H^{33/4}K^{1/4}$	$[S^6A^4H^{27}, \infty)$
`MVP` (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2AH^2$	$[S^3AH, \infty)$
`UCB-M-Q` (Menard et al. '21)	$\sqrt{SAH^2T} + SAH^4$	$[SAH^5, \infty)$
`Q-Earlysettled-Adv` (Li et al. '21)	$\sqrt{SAH^2T} + SAH^6$	$[SAH^9, \infty)$
`MVP` **(Zhang et al. '23)**	$\sqrt{SAH^3K}$	$[1, \infty)$

How about memory complexity?

Algorithm	Regret upper bound	Range of K that attains optimal regret	Memory complexity
UCBVI (Azar et al. 17)	$\sqrt{SAH^2T} + S^2AH^3$	$[S^3AH^3, \infty)$	S^2AH
UCB-Adv (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2A^{3/2}H^{33/4}K^{1/4}$	$[S^6A^4H^{27}, \infty)$	SAH
MVP (Zhang et al. '20)	$\sqrt{SAH^2T} + S^2AH^2$	$[S^3AH, \infty)$	S^2AH
UCB-M-Q (Menard et al. '21)	$\sqrt{SAH^2T} + SAH^4$	$[SAH^5, \infty)$	S^2AH
Q-Earlysettled-Adv (Li et al. '21)	$\sqrt{SAH^2T} + SAH^6$	$[SAH^9, \infty)$	SAH
MVP **(Zhang et al. '23)**	$\sqrt{SAH^3K}$	$[1, \infty)$	S^2AH

Can we find a regre-optimal algorithm with
(1) low burn-in cost and (2) low memory complexity?

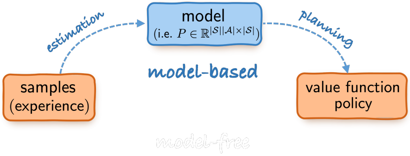
Model-free RL is often more memory-efficient



store transition kernel estimates

$\rightarrow$ $O(S^2AH)$ *memory*

Model-free RL is often more memory-efficient



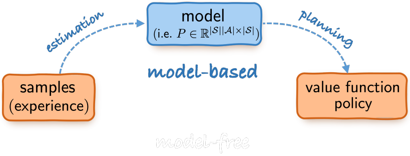
store transition kernel estimates

$\rightarrow$ $O(S^2AH)$ *memory*

maintain Q-estimates

$\rightarrow$ $O(SAH)$ *memory*

Model-free RL is often more memory-efficient



store transition kernel estimates
$\rightarrow$ $O(S^2AH)$ *memory*

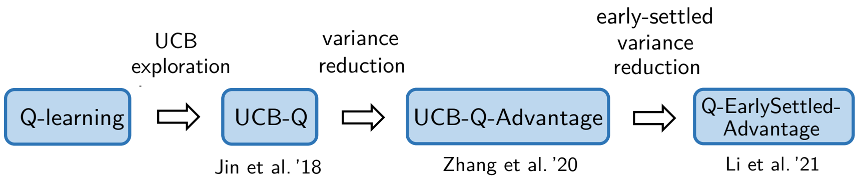
maintain Q-estimates
$\rightarrow$ $O(SAH)$ *memory*

Definition 5 (Jin et al. '18)

An RL algorithm is **model-free** if its space complexity is $o(S^2AH)$

Which model-free algorithms are sample-efficient for online RL?

Which model-free algorithms are sample-efficient for online RL?



Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

- $b_h(s, a)$: upper confidence bound; encourage exploration
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h,a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h,a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h,a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h,a_h)}_{\text{exploration bonus}}$$

- $b_h(s,a)$: upper confidence bound; encourage exploration
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Regret$(T) \lesssim \sqrt{H^3SAT} \quad \Longrightarrow \quad$ sub-optimal by a factor of $\sqrt{H}$

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k) Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

- $b_h(s, a)$: upper confidence bound; encourage exploration
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Regret$(T) \lesssim \sqrt{H^3 SAT} \quad \Longrightarrow \quad$ sub-optimal by a factor of $\sqrt{H}$

Issue: *large variability in stochastic update rules*

Our algorithm: Q-EarlySettled-Advantage

Theorem 6 (Li, Shi, Chen, Gu, Chi '21)

With high prob., Q-EarlySettled-Advantage *achieves (up to log factor)*

$$\mathsf{Regret}(T) \lesssim \sqrt{H^2SAT} + H^6SA$$

with a memory complexity of $O(SAH)$

Our algorithm: Q-EarlySettled-Advantage

Theorem 6 (Li, Shi, Chen, Gu, Chi '21)

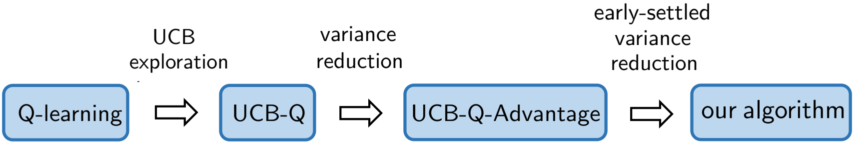
With high prob., Q-EarlySettled-Advantage *achieves (up to log factor)*

$$\text{Regret}(T) \lesssim \sqrt{H^2SAT} + H^6SA$$

with a memory complexity of $O(SAH)$

- regret-optimal with burn-in cost $O(SA\text{poly}(H))$
  - optimal in SA, suboptimal in H
- memory-efficient $O(SAH)$
- computationally efficient: runtime $O(T)$

A glimpse of our model-free algorithm design



A glimpse of our model-free algorithm design

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation

$$Q_h(s_h, a_h) \longleftarrow (1 - \eta_k) Q_h(s_h, a_h) + \eta_k \mathcal{T}_k(Q_{h+1})(s_h, a_h)$$

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving Bellman equation

$$Q_h(s_h, a_h) \longleftarrow (1-\eta_k) Q_h(s_h, a_h) + \eta_k \mathcal{T}_k(Q_{h+1})(s_h, a_h)$$

$$\mathcal{T}_k(Q_h)(s_h, a_h) = r(s_h, a_h) + \max_{a'} Q(s_{h+1}, a')$$

using sample in k-th episode

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

- $b_h(s, a)$: upper confidence bound
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h, a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h, a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h, a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h, a_h)}_{\text{exploration bonus}}$$

- $b_h(s, a)$: upper confidence bound
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Regret$(T) \lesssim \sqrt{H^3 SAT}$ $\Longrightarrow$ sub-optimal by a factor of $\sqrt{H}$

Q-learning with UCB exploration (Jin et al., 2018)

$$Q_h(s_h,a_h) \leftarrow \underbrace{(1-\eta_k)Q_h(s_h,a_h) + \eta_k \mathcal{T}_k\left(Q_{h+1}\right)(s_h,a_h)}_{\text{classical Q-learning}} + \eta_k \underbrace{b_h(s_h,a_h)}_{\text{exploration bonus}}$$

- $b_h(s,a)$: upper confidence bound
  — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins '85)

Regret$(T) \lesssim \sqrt{H^3SAT}$ $\implies$ sub-optimal by a factor of $\sqrt{H}$

Issue: *large variability in stochastic update rules*

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates variance reduction into UCB-Q:

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates reference-advantage decomposition into UCB-Q:

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates reference-advantage decomposition into UCB-Q:

$$Q_h(s_h,a_h) \leftarrow (1-\eta_k)Q_h(s_h,a_h) + \eta_k \underbrace{b_h(s_h,a_h)}_{\text{UCB bonus}}$$

$$+ \eta_k\Big(\underbrace{\mathcal{T}_k(Q_{h+1}) - \mathcal{T}_k(\overline{Q}_{h+1})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q}_{h+1})}_{\text{reference}}\Big)(s_h,a_h)$$

- Reference $\overline{Q}_{h+1}$, batch estimate $\widehat{\mathcal{T}}$: help reduce variability

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates reference-advantage decomposition into UCB-Q:

$$Q_h(s_h,a_h) \leftarrow (1-\eta_k)Q_h(s_h,a_h) + \eta_k \underbrace{b_h(s_h,a_h)}_{\text{UCB bonus}}$$
$$+ \eta_k \Big(\underbrace{\mathcal{T}_k(Q_{h+1}) - \mathcal{T}_k(\overline{Q}_{h+1})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q}_{h+1})}_{\text{reference}} \Big)(s_h,a_h)$$

- Reference $\overline{Q}_{h+1}$, batch estimate $\widehat{\mathcal{T}}$: help reduce variability

UCB-Q-Advantage is asymptotically regret-optimal

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates reference-advantage decomposition into UCB-Q:

$$Q_h(s_h,a_h) \leftarrow (1-\eta_k)Q_h(s_h,a_h) + \eta_k \underbrace{b_h(s_h,a_h)}_{\text{UCB bonus}}$$
$$+ \eta_k \Big(\underbrace{\mathcal{T}_k(Q_{h+1}) - \mathcal{T}_k(\overline{Q}_{h+1})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q}_{h+1})}_{\text{reference}} \Big)(s_h,a_h)$$

- Reference $\overline{Q}_{h+1}$, batch estimate $\widehat{\mathcal{T}}$: help reduce variability

UCB-Q-Advantage is asymptotically regret-optimal

Issue: *high burn-in cost* $O(S^6A^4H^{28})$

Diagnosis of UCB-Q-Advantage

Variance reduction requires sufficiently good references $\overline{Q}_h$

Diagnosis of UCB-Q-Advantage

Variance reduction requires sufficiently good references $\overline{Q}_h$

$$\Downarrow$$

Diagnosis of UCB-Q-Advantage

Variance reduction requires sufficiently good references $\overline{Q}_h$

$\Downarrow$

Updating references $\overline{Q}_h$ and $\overline{V}_h$ many times

$\Downarrow$

Diagnosis of UCB-Q-Advantage

Variance reduction requires sufficiently good references $\overline{Q}_h$

$\Downarrow$

Updating references $\overline{Q}_h$ and $\overline{V}_h$ many times

$\Downarrow$

Large burn-in cost

Diagnosis of UCB-Q-Advantage

Variance reduction requires sufficiently good references $\overline{Q}_h$

$\Downarrow$

Updating references $\overline{Q}_h$ and $\overline{V}_h$ many times

$\Downarrow$

Large burn-in cost

Key idea: early settlement of the reference as soon as it reaches a reasonable quality (e.g., $\overline{V}_h \leq V_h^\star + 1$)

How to implement our early-settlement idea?

$$\overline{V}_h(s) - V_h^\star(s) \le 1$$

How to implement our early-settlement idea?

$$\overline{V}_h(s) - V_h^\star(s) \le 1$$

$$\Uparrow$$

$$\overline{V}_h(s) - V_h^{\mathsf{LCB}}(s) \le 1 \quad \text{for some estimate } V_h^{\mathsf{LCB}} \le V_h^\star$$

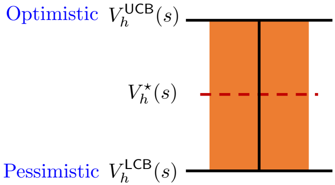
How to implement our early-settlement idea?

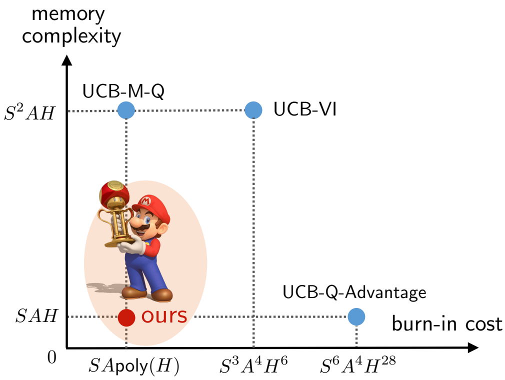
$$\overline{V}_h(s) - V_h^\star(s) \le 1$$

$$\Uparrow$$

$$\overline{V}_h(s) - V_h^{\mathsf{LCB}}(s) \le 1 \quad \text{for some estimate } V_h^{\mathsf{LCB}} \le V_h^\star$$

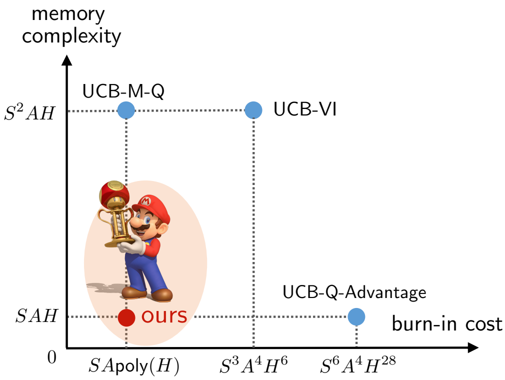
Q-EarlySettled-Advantage:
maintains auxiliary sequences V_h^{UCB} & V_h^{LCB} to help settle the reference early





Model-free algorithms can simultaneously achieve

(1) regret optimality; (2) low burn-in cost; (3) memory efficiency



Model-free algorithms can simultaneously achieve

(1) regret optimality; (2) low burn-in cost; (3) memory efficiency

Summary for online RL

- model-based approach is regret-optimal w/ no burn-in cost
- model-free approach is regret-optimal w/ low burn-in and low memory complexity

Summary for online RL

- model-based approach is regret-optimal w/ no burn-in cost
- model-free approach is regret-optimal w/ low burn-in and low memory complexity

open problems:

- how to design model-free algorithms w/o burn-in cost (i.e., w/ optimal H-dependency too)?

Summary for online RL

- model-based approach is regret-optimal w/ no burn-in cost
- model-free approach is regret-optimal w/ low burn-in and low memory complexity

open problems:

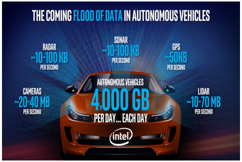
- how to design model-free algorithms w/o burn-in cost (i.e., w/ optimal H-dependency too)?
- how to achieve full-range regret-optimal algorithms for:
    - discounted infinite-horizon MDPs?
    - finite-horizon stationary MDPs?
    - ...

1. Online RL
2. Offline RL
3. Reward-agnostic exploration
4. Hybrid RL (policy finetuning)

Offline/batch RL

- Collecting new data might be costly, unsafe, unethical, or time-consuming

medical records



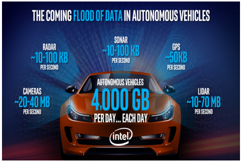
data of self-driving

clicking times of ads

Offline/batch RL

- Collecting new data might be costly, unsafe, unethical, or time-consuming
- But we have already stored tons of historical data

medical records



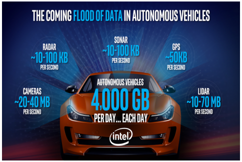
data of self-driving

clicking times of ads

Offline/batch RL

- Collecting new data might be costly, unsafe, unethical, or time-consuming
- But we have already stored tons of historical data

medical records

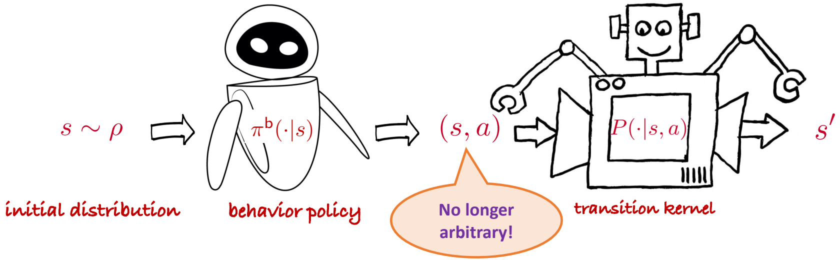


data of self-driving

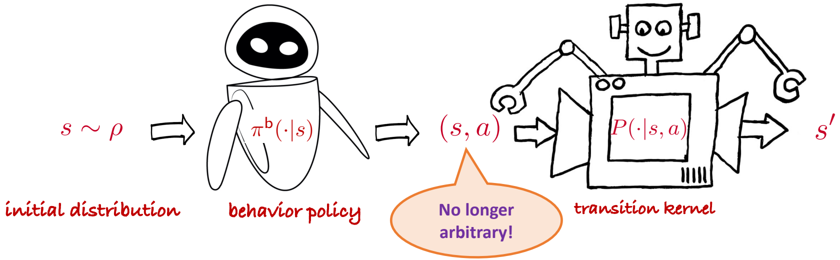
clicking times of ads

Question: can we learn based solely on historical data w/o active exploration?

A mathematical model of offline data



A mathematical model of offline data

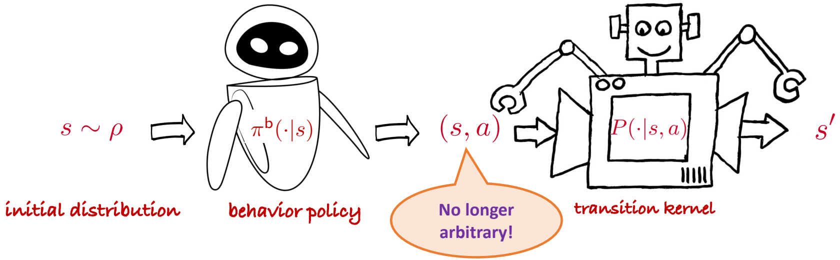


historical dataset $\mathcal{D} = \big\{\big(s^{(i)}, a^{(i)}, s'^{(i)}\big)\big\}$**:** N independent copies of

$$s \sim \rho, \qquad a \sim \pi^{\mathsf{b}}(\cdot \,|\, s), \qquad s' \sim P(\cdot \,|\, s, a)$$

- ρ: initial state distribution; π^{b}: behavior policy

A mathematical model of offline data



Goal: given a target accuracy level $\varepsilon \in (0, H]$, find $\widehat{\pi}$ s.t.

$$V^{\star}(\rho) - V^{\widehat{\pi}}(\rho) := \mathop{\mathbb{E}}_{s\sim\rho}\big[V^{\star}(s)\big] - \mathop{\mathbb{E}}_{s\sim\rho}\big[V^{\widehat{\pi}}(s)\big] \leq \varepsilon$$

— *in a sample-efficient manner*

Challenges of offline RL

- **Distribution shift**:

$$\text{distribution}(\mathcal{D}) \neq \text{target distribution under optimal } \pi^\star$$

Challenges of offline RL

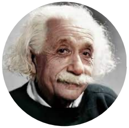
- **Distribution shift**:

$$\text{distribution}(\mathcal{D}) \neq \text{target distribution under optimal } \pi^\star$$

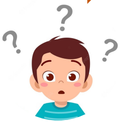
easier

harder

distance$(\pi^{\mathsf{b}}, \pi^\star)$



expert data

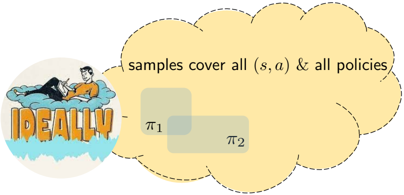


Challenges of offline RL

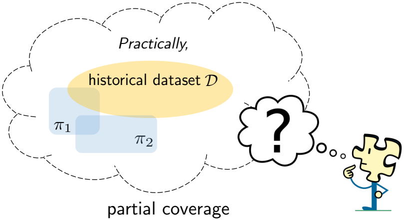
- **Distribution shift**:

$$\text{distribution}(\mathcal{D}) \;\neq\; \text{target distribution under optimal } \pi^\star$$

- **Partial coverage of state-action space**:



uniform coverage over entire space
(sufficiently explored)



partial coverage
(inadequately explored)

How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

Single-policy concentrability coefficient (Rashidineiad et al. '21)

$$C^{\star} := \max_{s,a} \frac{d^{\pi^{\star}}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)} = \left\| \frac{\textit{occupancy distribution of } \pi^{\star}}{\textit{occupancy distribution of } \pi^{\mathsf{b}}} \right\|_{\infty} \geq 1$$

How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

Single-policy concentrability coefficient (Rashidineiad et al. '21)

$$C^\star := \max_{s,a} \frac{d^{\pi^\star}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)} = \left\| \frac{\textit{occupancy distribution of } \pi^\star}{\textit{occupancy distribution of } \pi^{\mathsf{b}}} \right\|_\infty \geq 1$$

- captures distributional shift

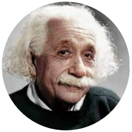
How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

Single-policy concentrability coefficient (Rashidineiad et al. '21)

$$C^\star := \max_{s,a} \frac{d^{\pi^\star}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)} = \left\| \frac{\textit{occupancy distribution of } \pi^\star}{\textit{occupancy distribution of } \pi^{\mathsf{b}}} \right\|_\infty \geq 1$$

- captures distributional shift

$C^\star = O(1)$ → large $C^\star$



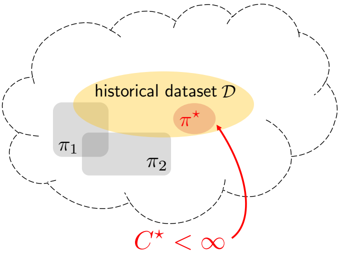
expert data

How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

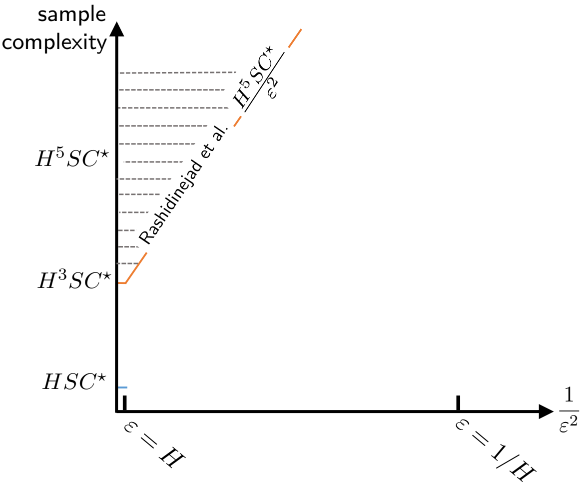
Single-policy concentrability coefficient (Rashidineiad et al. '21)

$$C^{\star} := \max_{s,a} \frac{d^{\pi^{\star}}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)} = \left\| \frac{\textit{occupancy distribution of } \pi^{\star}}{\textit{occupancy distribution of } \pi^{\mathsf{b}}} \right\|_{\infty} \geq 1$$

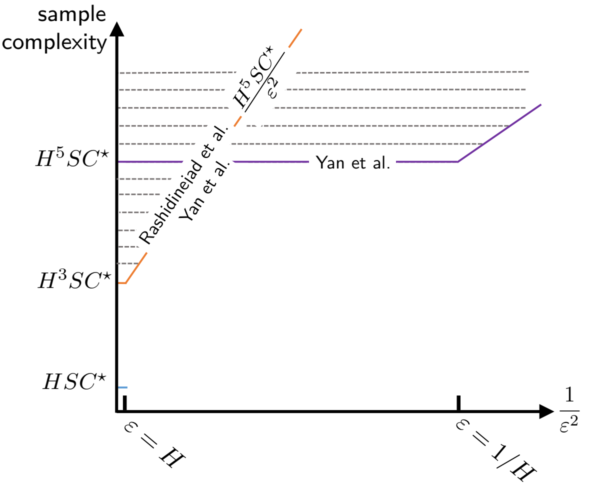
- captures distributional shift
- allows for partial coverage
  - as long as it covers the part reachable by $\pi^{\star}$



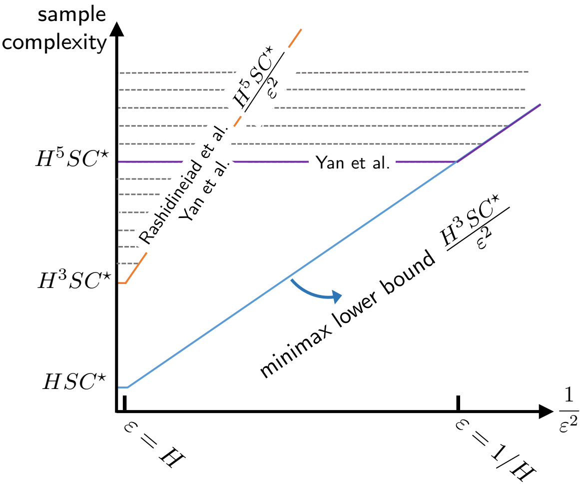
Prior art: sample complexity bounds



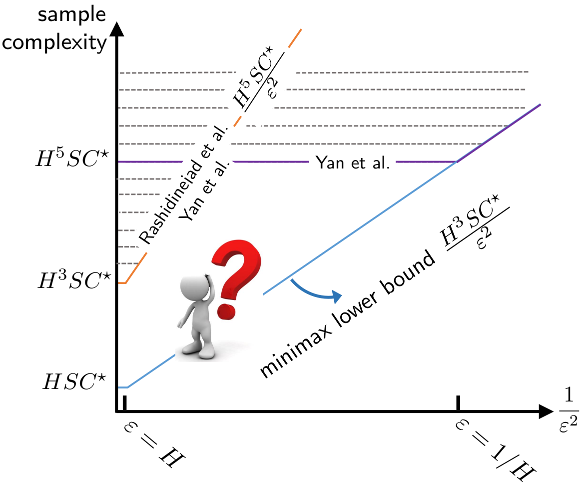
Prior art: sample complexity bounds



Prior art: sample complexity bounds



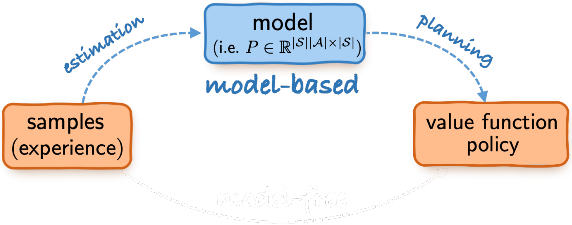
Prior art: sample complexity bounds



Can we close the gap between upper & lower bounds?

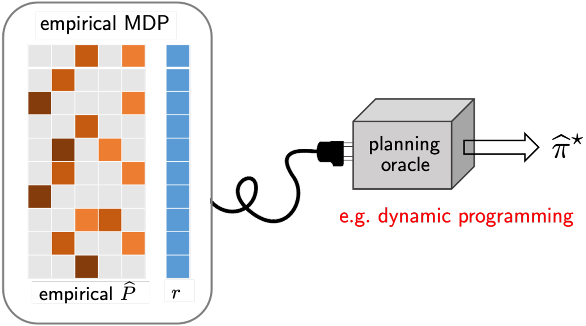
Model-based ("plug-in") approach?

— *Azar et al. '13, Agarwal et al. '19, Li et al. '20*



Model-based ("plug-in") approach?

— *Azar et al. '13, Agarwal et al. '19, Li et al. '20*

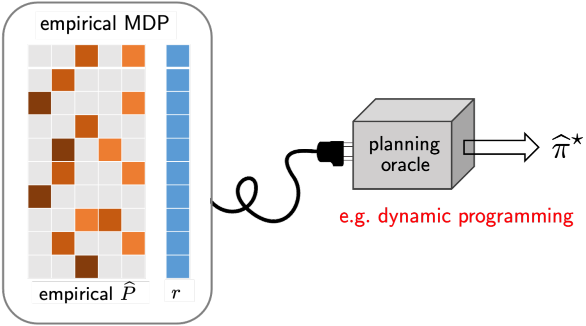


1. construct empirical model $\widehat{P}$:

$$\widehat{P}(s' \mid s, a) = \underbrace{\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{s'^{(i)} = s'\}}_{\text{empirical frequency}}$$

Model-based ("plug-in") approach?

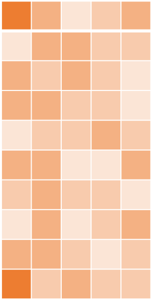
— Azar et al. '13, Agarwal et al. '19, Li et al. '20



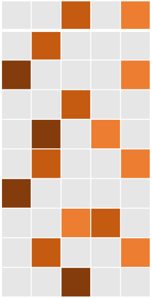
1. construct empirical model $\widehat{P}$
2. planning (e.g. value iteration) based on empirical MDP

— best under generative model (Li, Wei, Chi, Chen '20)

Issues & challenges in the sample-starved regime



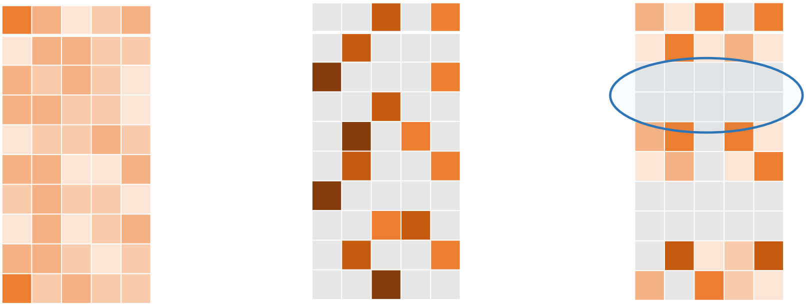
truth: $P \in \mathbb{R}^{SA \times S}$



empirical $\widehat{P}$ (simulator)

- can't recover P faithfully if sample size $\ll S^2 A$

Issues & challenges in the sample-starved regime



truth: $P \in \mathbb{R}^{SA \times S}$ empirical $\widehat{P}$ (simulator) empirical $\widehat{P}$ (offline)

- can't recover P faithfully if sample size $\ll S^2 A$
- (possibly) insufficient coverage under offline data

Key idea: pessimism in the face of uncertainty

— *Jin et al. '20, Rashidinejad et al. '21, Xie et al. '21*


online

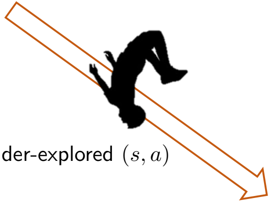
upper confidence bounds

— promote exploration of under-explored (s, a)

Key idea: pessimism in the face of uncertainty

— *Jin et al. '20, Rashidinejad et al. '21, Xie et al. '21*

online



upper confidence bounds

— promote exploration of under-explored (s, a)

offline

lower confidence bounds

— stay cautious about under-explored (s, a)

Key idea: pessimism in the face of uncertainty

— *Jin et al. '20, Rashidinejad et al. '21, Xie et al. '21*

1. build empirical model $\widehat{P}$
2. **(value iteration)** repeat: for all (s, a)

$$\widehat{Q}(s,a) \;\leftarrow\; \max\Big\{ r(s,a) + \gamma \langle \widehat{P}(\cdot \,|\, s,a), \widehat{V} \rangle,\; 0 \Big\}$$

where $\widehat{V}(s) = \max_a \widehat{Q}(s,a)$

Key idea: pessimism in the face of uncertainty

— *Jin et al. '20, Rashidinejad et al. '21, Xie et al. '21*

Penalize those poorly visited (s, a) ...

1. build empirical model $\widehat{P}$
2. **(pessimistic value iteration)** repeat: for all (s, a)

$$\widehat{Q}(s,a) \;\leftarrow\; \max\Big\{ r(s,a) + \gamma \langle \widehat{P}(\cdot \,|\, s,a), \widehat{V} \rangle - \underbrace{b(s,a;\widehat{V})}_{\text{uncertainty penalty}},\; 0 \Big\}$$

where $\widehat{V}(s) = \max_a \widehat{Q}(s,a)$

Key idea: pessimism in the face of uncertainty

— *Jin et al. '20, Rashidinejad et al. '21, Xie et al. '21*

Penalize those poorly visited (s, a) ...

1. build empirical model $\widehat{P}$
2. **(pessimistic value iteration)** repeat: for all (s, a)

$$\widehat{Q}(s,a) \;\leftarrow\; \max\Big\{ r(s,a) + \gamma \langle \widehat{P}(\cdot \,|\, s,a), \widehat{V} \rangle - \underbrace{b(s,a;\widehat{V})}_{\text{uncertainty penalty}},\; 0 \Big\}$$

compared w/ Rashidinejad et al. '21

- sample-reuse across iterations
- Bernstein-style penalty

Sample complexity of model-based offline RL

Theorem 7 (Li, Shi, Chen, Chi, Wei '22)

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *the policy* $\widehat{\pi}$ *returned by VI-LCB using a Bernstein-style penalty term achieves*

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

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{SC^{\star}}{(1-\gamma)^3\varepsilon^2}\right)$$

Sample complexity of model-based offline RL

Theorem 7 (Li, Shi, Chen, Chi, Wei '22)

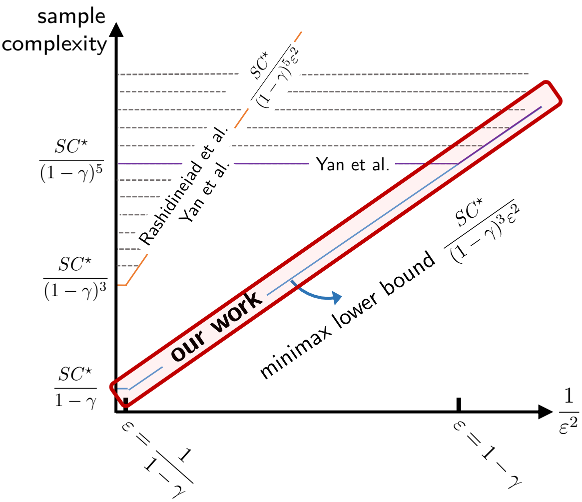
For any $0 < \varepsilon \le \frac{1}{1-\gamma}$, the policy $\widehat{\pi}$ returned by VI-LCB using a Bernstein-style penalty term achieves

$$V^{\star}(\rho) - V^{\widehat{\pi}}(\rho) \le \varepsilon$$

with high prob., with sample complexity at most

$$\widetilde{O}\left(\frac{SC^{\star}}{(1-\gamma)^3\varepsilon^2}\right)$$

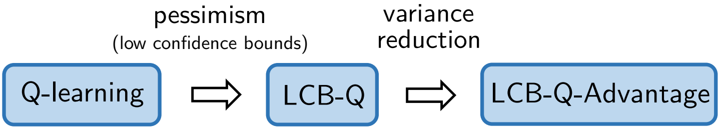
- depends on distribution shift (as reflected by $C^{\star}$)
- achieves minimax optimality
- full ε-range (no burn-in cost)



Model-based offline RL is minimax optimal with no burn-in cost!

Is it possible to design offline model-free algorithms with optimal sample efficiency?

Is it possible to design offline model-free algorithms with optimal sample efficiency?



LCB-Q: Q-learning with LCB penalty

— *Shi et al. '22, Yan et al. '22*

$$Q_{t+1}(s_t, a_t) \leftarrow \underbrace{(1-\eta_t)Q_t(s_t, a_t) + \eta_t \mathcal{T}_t\left(Q_t\right)(s_t, a_t)}_{\text{classical Q-learning}} - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$

LCB-Q: Q-learning with LCB penalty

— *Shi et al. '22, Yan et al. '22*

$$Q_{t+1}(s_t, a_t) \leftarrow \underbrace{(1-\eta_t)Q_t(s_t, a_t) + \eta_t \mathcal{T}_t\left(Q_t\right)(s_t, a_t)}_{\text{classical Q-learning}} - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$

- $b_t(s, a)$: Hoeffding-style confidence bound
- pessimism in the face of uncertainty

LCB-Q: Q-learning with LCB penalty

— *Shi et al. '22, Yan et al. '22*

$$Q_{t+1}(s_t, a_t) \leftarrow \underbrace{(1-\eta_t)Q_t(s_t, a_t) + \eta_t \mathcal{T}_t\left(Q_t\right)(s_t, a_t)}_{\text{classical Q-learning}} - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$

- $b_t(s, a)$: Hoeffding-style confidence bound
- pessimism in the face of uncertainty

sample size: $\widetilde{O}\left(\frac{SC^{\star}}{(1-\gamma)^5 \varepsilon^2}\right) \implies$ sub-optimal by a factor of $\frac{1}{(1-\gamma)^2}$

Issue: *large variability in stochastic update rules*

Q-learning with LCB and variance reduction

— *Shi et al. '22, Yan et al. '22*

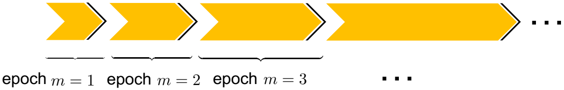
$$Q_{t+1}(s_t, a_t) \leftarrow (1-\eta_t)Q_t(s_t, a_t) - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$
$$+ \eta_t\Big(\underbrace{\mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q})}_{\text{reference}}\Big)(s_t, a_t)$$

Q-learning with LCB and variance reduction

— *Shi et al. '22, Yan et al. '22*

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\eta_t) Q_t(s_t, a_t) - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$
$$+ \eta_t \Big(\underbrace{\mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q})}_{\text{reference}} \Big)(s_t, a_t)$$

- incorporates variance reduction into LCB-Q

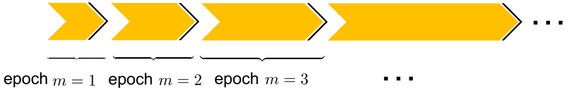


Q-learning with LCB and variance reduction

— *Shi et al. '22, Yan et al. '22*

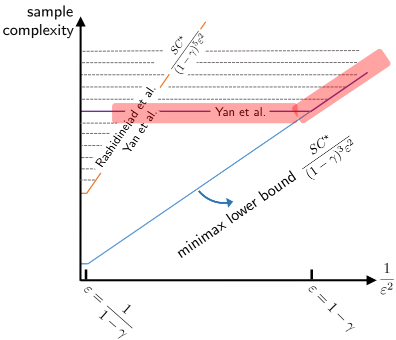
$$Q_{t+1}(s_t, a_t) \leftarrow (1-\eta_t) Q_t(s_t, a_t) - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$
$$+ \eta_t \Big(\underbrace{\mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q})}_{\text{reference}} \Big)(s_t, a_t)$$

- incorporates variance reduction into LCB-Q

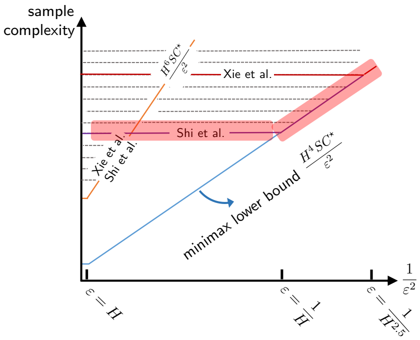


Theorem 8 (Yan, Li, Chen, Fan '22, Shi, Li, Wei, Chen, Chi '22)

For $\varepsilon \in (0, 1-\gamma]$, LCB-Q-Advantage *achieves* $V^\star(\rho) - V^{\widehat{\pi}}(\rho) \le \varepsilon$ *with optimal sample complexity* $\widetilde{O}\big(\frac{SC^\star}{(1-\gamma)^3\varepsilon^2}\big)$



infinite-horizon MDPs



finite-horizon MDPs

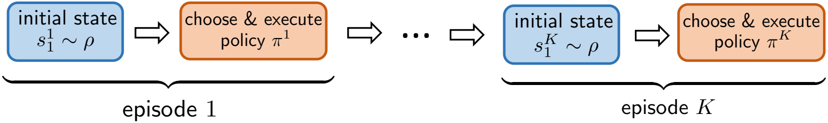
Model-free offline RL attains sample optimality too!

— with some burn-in cost though . . .

1. Online RL
2. Offline RL
3. Reward-agnostic exploration
4. Hybrid RL (policy finetuning)

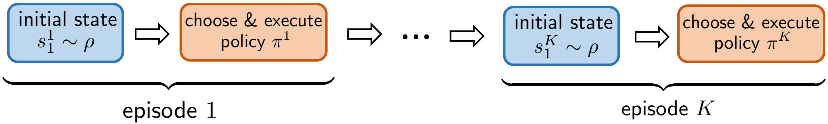
Online RL: interacting with real environments

Sequentially execute MDP for K episodes, each containing H steps



Online RL: interacting with real environments

Sequentially execute MDP for K episodes, each containing H steps



Key: exploration-exploitation tradeoff

- Lai & Robbins '85
- Jaksch, Ortner, Auer '10
- Azar, Osband, Munos '17
- Chi, Allen-Zhu, Bubeck, Jordan '18
- ...

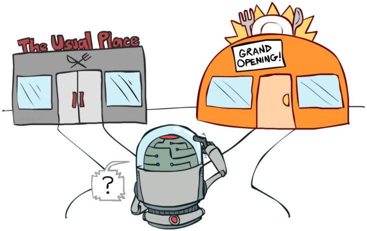
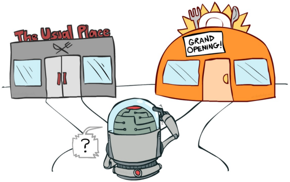


fig. credit: Berkeley CS188

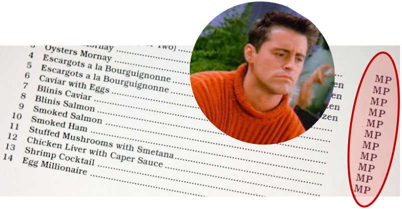
Reward-agnostic exploration?

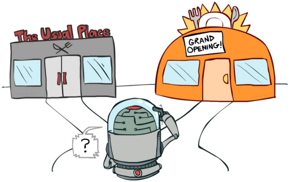
The learner is unaware of the rewards during exploration . . .



Reward-agnostic exploration?

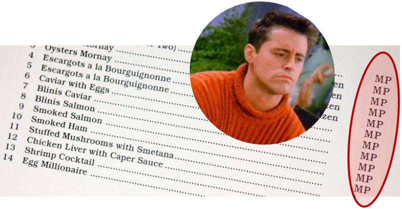
The learner is unaware of the rewards during exploration . . .

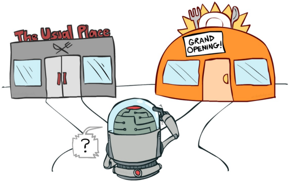




Reward-agnostic exploration?

The learner is unaware of the rewards during exploration . . .



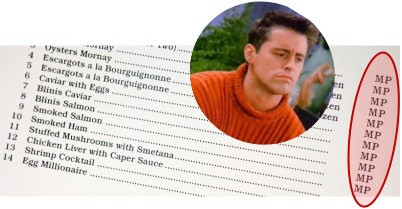


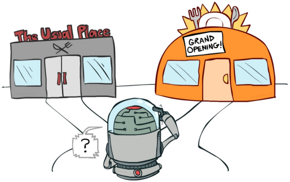
Motivation

- (significantly) delayed feedback
- reward functions keep changing
- offline RL
- many reward functions of interest

Reward-agnostic exploration?

The learner is unaware of the rewards during exploration ...





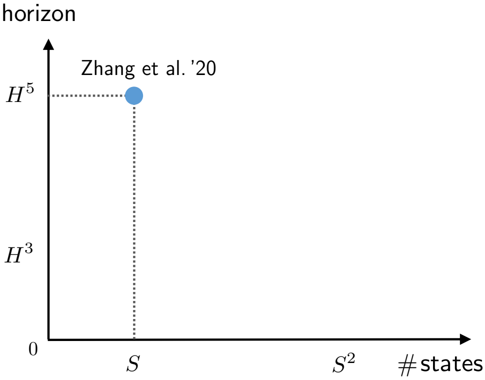
Motivation

- (significantly) delayed feedback
- reward functions keep changing
- offline RL
- many reward functions of interest

Question: can we perform pure exploration just once but achieve efficiency for many <u>unseen</u> reward functions at once?

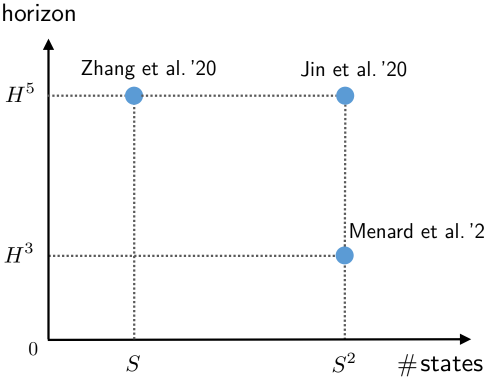
Prior art: sample complexity upper bounds

Suppose there is a fixed (but unseen) reward function of interest ...



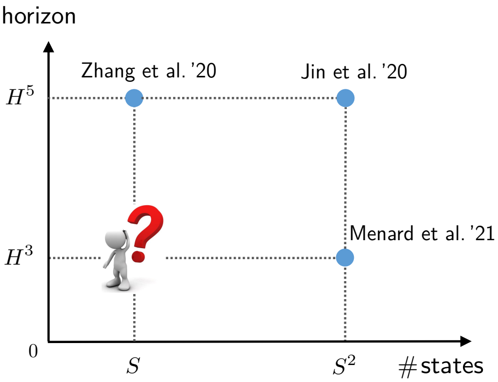
Prior art: sample complexity upper bounds

Suppose there is a fixed (but unseen) reward function of interest ...



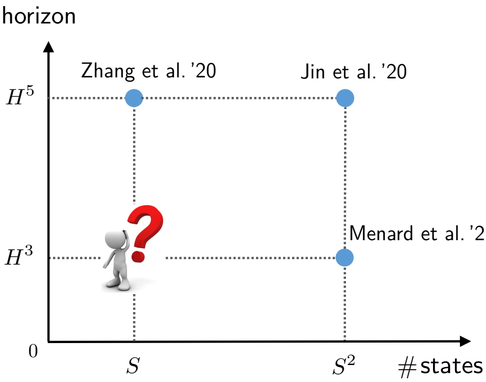
Prior art: sample complexity upper bounds

Suppose there is a fixed (but unseen) reward function of interest ...

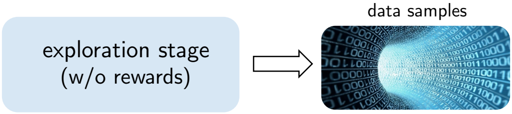


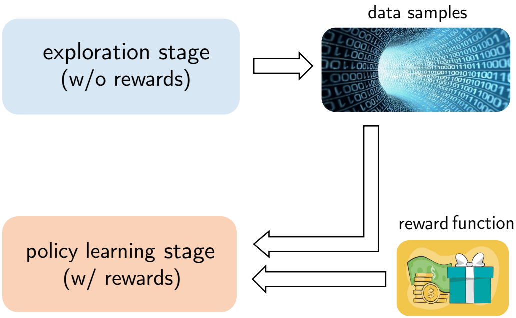
Prior art: sample complexity upper bounds

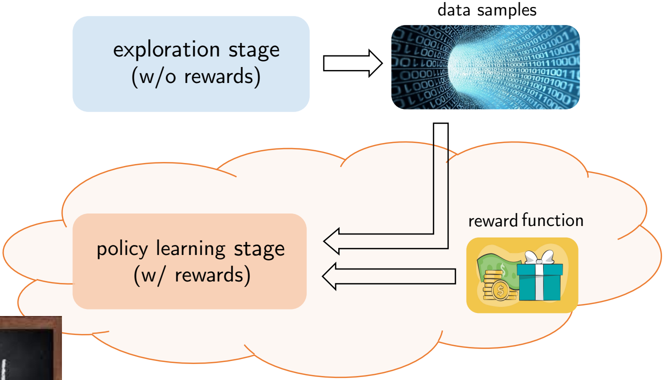
Suppose there is a fixed (but unseen) reward function of interest ...



Question: can we simultaneously optimize dependency on S & H?


exploration stage
(w/o rewards)
data samples

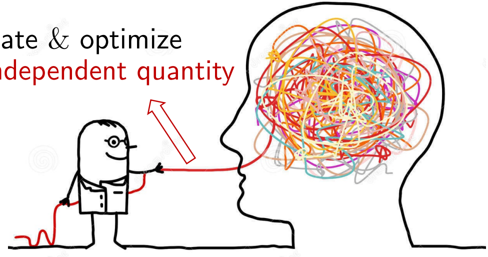

exploration stage
(w/o rewards)
data samples
reward function
policy learning stage
(w/ rewards)


exploration stage
(w/o rewards)
data samples
policy learning stage
(w/ rewards)
reward function

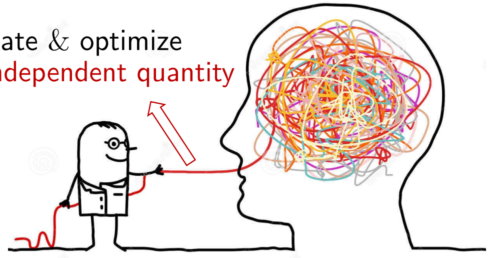

LESSON 1
Offline RL

isolate & optimize

reward-independent quantity



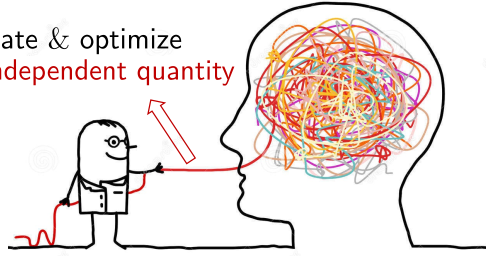
isolate & optimize
reward-independent quantity



lessons learned from offline RL: offline model-based alg. gives

$$V_1^{\star}(\rho) - V_1^{\widehat{\pi}}(\rho) \lesssim \frac{1}{\sqrt{K}} \sum_{h,s,a} d_h^{\pi^{\star}}(s,a) \min\left\{ \sqrt{\frac{\mathsf{Var}_{h,s,a}(V_{h+1}^{\star})}{d_h^{\mathsf{behavior}}(s,a)}},\, H \right\}$$

isolate & optimize
reward-independent quantity

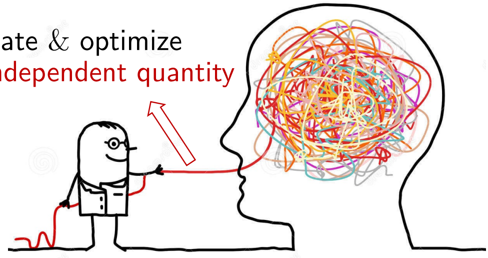


lessons learned from offline RL: offline model-based alg. gives

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho)$$

$$\lesssim \frac{1}{\sqrt{K}} \underbrace{\left(\max_{\pi} \sum_{h,s,a} \frac{d_h^\pi(s,a)}{\frac{1}{KH} + d_h^{\mathsf{behavior}}(s,a)} \right)^{\frac{1}{2}}}_{\text{reward-independent}} \underbrace{\left(\sum_{h,s,a} d_h^{\pi^\star}(s,a) \mathsf{Var}_{h,s,a}(V_{h+1}^\star) + H \right)^{\frac{1}{2}}}_{\text{reward-dependent}}$$

isolate & optimize
reward-independent quantity



lessons learned from offline RL: offline model-based alg. gives

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho)$$

$$\lesssim \frac{1}{\sqrt{K}} \underbrace{\left(\max_\pi \sum_{h,s,a} \frac{d_h^\pi(s,a)}{\frac{1}{KH} + d_h^{\text{behavior}}(s,a)} \right)^{\frac{1}{2}}}_{\text{reward-independent}} \underbrace{\left(\sum_{h,s,a} d_h^{\pi^\star}(s,a)\mathsf{Var}_{h,s,a}(V_{h+1}^\star) + H \right)^{\frac{1}{2}}}_{\text{reward-dependent}}$$

key: find behavior policy to optimize reward-independent quantity

Our algorithm

exploration stage
(w/o rewards)

policy learning
(w/ rewards)

Our algorithm

for $h = 1, \ldots, H$

draw samples to estimate occupancy distributions d_h^π for all π

exploration stage (w/o rewards)

policy learning (w/ rewards)

Our algorithm

exploration stage
(w/o rewards)

for $h = 1, \ldots, H$

draw samples to estimate occupancy distributions d_h^π for all π

compute behavior policy π^{b}

$$\underset{\mu \in \Delta(\text{det. policies})}{\text{maximize}} \sum_{h,s,a} \log\left(\frac{1}{KH} + \mathop{\mathbb{E}}_{\pi \sim \mu}\left[\widehat{d}_h^\pi(s,a)\right]\right)$$

via Frank-Wolfe

policy learning
(w/ rewards)

Our algorithm

exploration stage
(w/o rewards)

for $h = 1, \dots, H$

draw samples to estimate occupancy distributions d_h^π for all π

compute behavior policy π^{b}

execute π^{b} to draw sample episodes

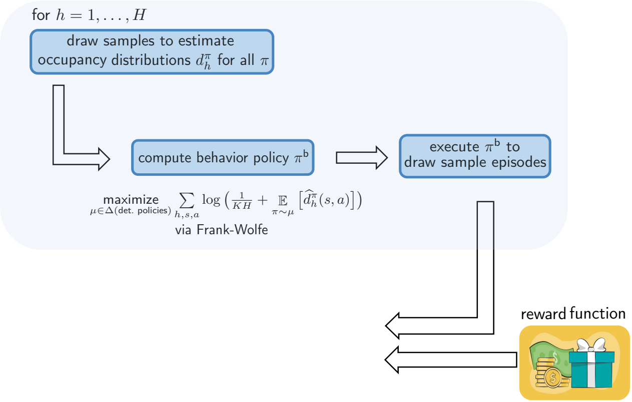
$$\underset{\mu \in \Delta(\text{det. policies})}{\text{maximize}} \sum_{h,s,a} \log \Big(\frac{1}{KH} + \mathop{\mathbb{E}}_{\pi \sim \mu} \big[\widehat{d}_h^\pi(s,a) \big] \Big)$$

via Frank-Wolfe

policy learning
(w/ rewards)

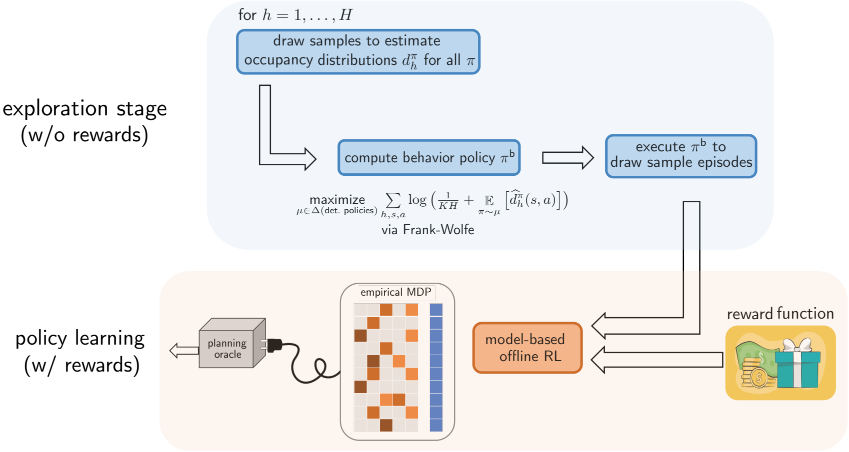
Our algorithm

exploration stage
(w/o rewards)



policy learning
(w/ rewards)

Our algorithm



Sample complexity of reward-agnostic RL

Theorem 9 (Li, Yan, Chen, Fan '23)

Suppose there are poly(H, S, A) *fixed reward functions of interest, and suppose* ε *is small enough. Using the same batch of samples w/*

$$\widetilde{O}\left(\frac{H^3 SA}{\varepsilon^2}\right) \text{ episodes,}$$

our algorithm can find, for each reward function, a policy $\widehat{\pi}$ *obeying*

$$V_1^{\star}(\rho) - V_1^{\widehat{\pi}}(\rho) \le \varepsilon$$

Sample complexity of reward-agnostic RL

Theorem 9 (Li, Yan, Chen, Fan '23)

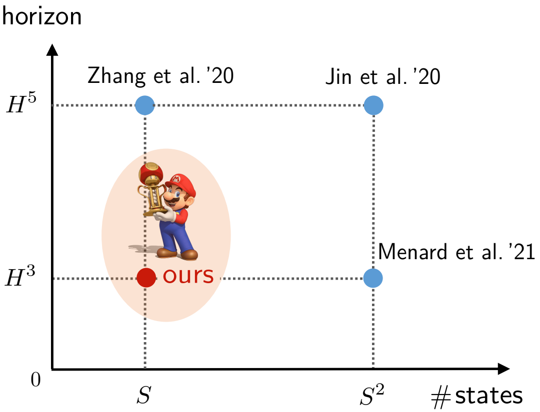
Suppose there are poly(H, S, A) *fixed reward functions of interest, and suppose* ε *is small enough. Using the same batch of samples w/*

$$\widetilde{O}\left(\frac{H^3SA}{\varepsilon^2}\right) \text{ episodes,}$$

our algorithm can find, for each reward function, a policy $\widehat{\pi}$ *obeying*

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

- optimal sample complexity
- collect data once $\longrightarrow$ work for poly(H, S, A) reward functions



The studies of offline RL inspire optimal reward-agnostic exploration!

1. Online RL
2. Offline RL
3. Reward-agnostic exploration
4. Hybrid RL (policy finetuning)

Hybrid RL

In practice, one might have access to both offline data and online sampling

- pre-training using offline data
- policy finetuning w/ aid of online data collection

Hybrid RL

In practice, one might have access to both offline data and online sampling

- pre-training using offline data
- policy finetuning w/ aid of online data collection

Question: what are the benefits of combining online & offline RL?

Prior sample complexity

pure offline RL: imagine there exists a behavior policy generating all offline data, then sample complexity is (Li et al. '22)

$$\frac{S{\color{red}C^\star}H^3}{\varepsilon^2}$$

Prior sample complexity

pure offline RL: imagine there exists a behavior policy generating all offline data, then sample complexity is (Li et al. '22)

$$\frac{SC^{\star}H^3}{\varepsilon^2}$$

pure online RL: sample complexity is (Azar et al. '17, Li et al. '22)

$$\frac{SAH^3}{\varepsilon^2}$$

Prior sample complexity

pure offline RL: imagine there exists a behavior policy generating all offline data, then sample complexity is (Li et al. '22)

$$\frac{SC^{\star}H^3}{\varepsilon^2}$$

pure online RL: sample complexity is (Azar et al. '17, Li et al. '22)

$$\frac{SAH^3}{\varepsilon^2}$$

prior work Xie et al. '21: sample complexity of hybrid RL is at most

$$\frac{S\min\{C^{\star}, A\}H^3}{\varepsilon^2}$$

- *not better than best of pure online and pure offline though . . .*

Does hybrid RL enjoy strict benefits over the best of offline and online RL?

Single-policy partial concentrability

Definition 10 (Li, Zhan, Lee, Chi, Chen '23)

For any $\sigma \in [0,1]$ (mis-coverage level),

$$C^\star(\sigma) := \min\left\{ \underbrace{\max_{1\leq h\leq H} \max_{(s,a)\in\mathcal{G}_h} \frac{d_h^\star(s,a)}{d_h^{\mathsf{offline}}(s,a)}}_{\text{distribution shift}} \;\middle|\; \{\mathcal{G}_h\}_{1\leq h\leq H} \subseteq \mathcal{G}(\sigma) \right\}$$

Single-policy partial concentrability

Definition 10 (Li, Zhan, Lee, Chi, Chen '23)

For any $\sigma \in [0,1]$ (mis-coverage level),

$$C^\star(\sigma) := \min\left\{ \underbrace{\max_{1\le h\le H}\max_{(s,a)\in\mathcal{G}_h} \frac{d_h^\star(s,a)}{d_h^{\mathsf{offline}}(s,a)}}_{\text{distribution shift}} \;\Big|\; \{\mathcal{G}_h\}_{1\le h\le H} \subseteq \mathcal{G}(\sigma) \right\}$$

where

$$\mathcal{G}(\sigma) := \left\{ \{\mathcal{G}_h\}_{1\le h\le H} \subseteq \mathcal{S}\times\mathcal{A} \;\Big|\; \underbrace{\frac{1}{H}\sum_{h=1}^{H}\sum_{(s,a)\notin\mathcal{G}_h} d_h^\star(s,a) \le \sigma}_{\text{mis-coverage}} \right\}$$

Single-policy partial concentrability

Definition 10 (Li, Zhan, Lee, Chi, Chen '23)

For any $\sigma \in [0,1]$ (mis-coverage level),

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

where

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

- reflects trade-off btw partial coverage & distribution mismatch
- $C^\star(\sigma)$: non-increasing in σ; $C^\star(0) = C^\star$

Provable benefits of hybrid RL

Theorem 11 (Li, Zhan, Lee, Chi, Chen '23)

Suppose $K^{\text{online}} = K^{\text{offline}} = K/2$ *(for simplicity), and suppose* ε *is small enough. For any* $\sigma \in [0,1]$*, using an order of*

$$\max\left\{\frac{H^3 SA \min\{H\sigma, 1\}}{\varepsilon^2} + \frac{H^3 S C^\star(\sigma)}{\varepsilon^2}\right\} \text{ episodes,}$$

our algorithm can find a policy $\widehat{\pi}$ *obeying*

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) \le \varepsilon$$

Provable benefits of hybrid RL

Theorem 11 (Li, Zhan, Lee, Chi, Chen '23)

Suppose $K^{\text{online}} = K^{\text{offline}} = K/2$ *(for simplicity), and suppose* ε *is small enough. For any* $\sigma \in [0,1]$, *using an order of*

$$\max\left\{\frac{H^3 SA \min\{H\sigma, 1\}}{\varepsilon^2} + \frac{H^3 SC^\star(\sigma)}{\varepsilon^2}\right\} \textit{ episodes,}$$

our algorithm can find a policy $\widehat{\pi}$ *obeying*

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) \le \varepsilon$$

- taking $\sigma = 0$ gives $\frac{H^3 SC^\star}{\varepsilon^2}$ (pure offline)

Provable benefits of hybrid RL

Theorem 11 (Li, Zhan, Lee, Chi, Chen '23)

Suppose $K^{\text{online}} = K^{\text{offline}} = K/2$ (for simplicity), and suppose ε is small enough. For any $\sigma \in [0,1]$, using an order of

$$\max\left\{\frac{H^3SA\min\{H\sigma,1\}}{\varepsilon^2} + \frac{H^3SC^\star(\sigma)}{\varepsilon^2}\right\} \text{ episodes,}$$

our algorithm can find a policy $\widehat{\pi}$ obeying

$$V_1^\star(\rho) - V_1^{\widehat{\pi}}(\rho) \le \varepsilon$$

- taking $\sigma = 0$ gives $\frac{H^3SC^\star}{\varepsilon^2}$ (pure offline)
- taking $\sigma = 1$ gives $\frac{H^3SA}{\varepsilon^2}$ (pure online)

Provable benefits of hybrid RL

Theorem 11 (Li, Zhan, Lee, Chi, Chen '23)

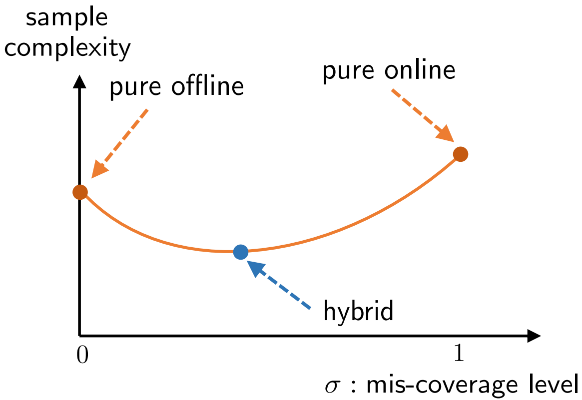
Suppose $K^{\text{online}} = K^{\text{offline}} = K/2$ (for simplicity), and suppose ε is small enough. For any $\sigma \in [0,1]$, using an order of

$$\max\left\{\frac{H^3 SA \min\{H\sigma, 1\}}{\varepsilon^2} + \frac{H^3 SC^\star(\sigma)}{\varepsilon^2}\right\} \text{ episodes,}$$

our algorithm can find a policy $\widehat{\pi}$ obeying

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

- taking $\sigma = 0$ gives $\frac{H^3 SC^\star}{\varepsilon^2}$ (pure offline)
- taking $\sigma = 1$ gives $\frac{H^3 SA}{\varepsilon^2}$ (pure online)
- strict sample size saving over both pure offline & pure online!



- our algorithm automatically finds the best σ (without knowing it)
- algorithm design: inspired by reward-agnostic exploration

Reference I

- "*A stochastic approximation method,*" H. Robbins, S. Monro, *Annals of mathematical statistics*, 1951
- "*Q-learning,*" C. Watkins, P. Dayan, *Machine learning*, 1992
- "*Is Q-Learning minimax optimal? A tight sample complexity analysis,*" G. Li, Y. Wei, Y. Chi, Y. Chen, *Operations Research*, 2023+.
- "*Accelerating stochastic gradient descent using predictive variance reduction*," R. Johnson, T. Zhang, *NeurIPS*, 2013
- "*Asymptotically efficient adaptive allocation rules*," T. L. Lai, H. Robbins, *Advances in applied mathematics*, vol. 6, no. 1, 1985.
- "*Finite-time analysis of the multiarmed bandit problem*," P. Auer, N. Cesa-Bianchi, P. Fischer, *Machine learning*, vol. 47, pp. 235-256, 2002.
- "*Minimax regret bounds for reinforcement learning*," M. G. Azar, I. Osband, R. Munos, *ICML* 2017.
- "*Is Q-learning provably efficient?*" C. Jin, Z. Allen-Zhu, S. Bubeck, and M. Jordan, *NeurIPS* 2018.

Reference II

- "*Episodic reinforcement learning in finite MDPs: Minimax lower bounds revisited*" O. D. Domingues, P. Menard, E. Kaufmann, M. Valko, *Algorithmic Learning Theory*, 2021.
- "*Almost optimal model-free reinforcement learning via reference-advantage decomposition,*" Z. Zhang, Y. Zhou, X. Ji, *NeurIPS* 2020.
- "*Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon,*" Z. Zhang, X. Ji, and S. Du, *COLT* 2021.
- "*Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning,*" G. Li, L. Shi, Y. Chen, Y. Chi, *Information and Inference: A Journal of the IMA*, 2023.
- "*Settling the sample complexity of online reinforcement learning,*" Z. Zhang, Y. Chen, J. D. Lee, S. S. Du, arXiv:2307.13586, 2023.
- "*Bridging offline reinforcement learning and imitation learning: A tale of pessimism,*" P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, S. Russell, *NeurIPS* 2021.

Reference III

- "*Is pessimism provably efficient for offline RL?*" Y. Jin, Z. Yang, Z. Wang, *ICML* 2021.
- "*Settling the sample complexity of model-based offline reinforcement learning,*" G. Li, L. Shi, Y. Chen, Y. Chi, Y. Wei, arXiv:2204.05275, 2022.
- "*Pessimistic Q-learning for offline reinforcement learning: Towards optimal sample complexity*," L. Shi, G. Li, Y. Wei, Y. Chen, Y. Chi, *ICML* 2022.
- "*The efficacy of pessimism in asynchronous Q-learning*," Y. Yan, G. Li, Y. Chen, J. Fan, *IEEE Transactions on Information Theory*, 2023.
- "*Reward-free exploration for reinforcement learning*," C. Jin, A. Krishnamurthy, M. Simchowitz, T. Yu, *ICML* 2020.
- "*Minimax-optimal reward-agnostic exploration in reinforcement learning*," G. Li, Y. Yan, Y. Chen, J. Fan, arXiv:2304.07278, 2023.
- "*Policy finetuning: Bridging sample-efficient offline and online reinforcement learning*" T. Xie, N. Jiang, H. Wang, C. Xiong, Y. Bai, *NeurIPS* 2021.

Reference IV

- "*Reward-agnostic fine-tuning: provable statistical benefits of hybrid reinforcement learning*," G. Li, W. Zhan, J. D. Lee, Y. Chi, Y. Chen, arXiv:2305.10282, 2023.

Statistical and Algorithmic Foundations of Reinforcement Learning (Part 3)

Yuejie Chi

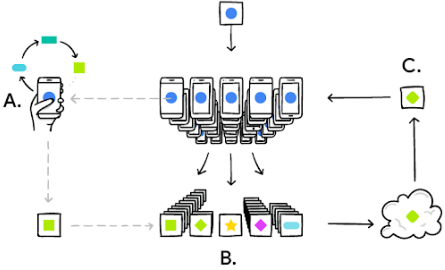
Carnegie Mellon University

JSM Short Course
August 2023

Federated and robust RL

1. Federated RL
2. Robust RL

Can we harness the power of federated learning?




FORBES > INNOVATION > AI

IBM Federated Learning Research – Extracting Machine Learning Models From Multiple Data Pools

Kevin Krewell Contributor
Tirias Research Contributor Group

Follow

Oct 15, 2021, 02:51pm EDT


How Apple personalizes Siri without hoovering up your data

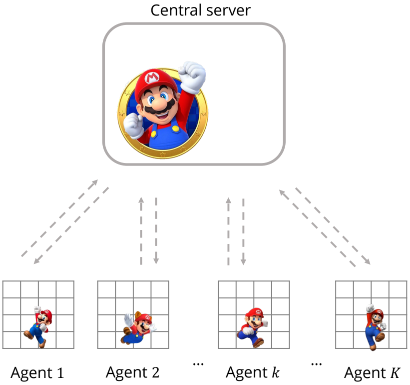
The tech giant is using privacy-preserving machine learning to improve its voice assistant while keeping your data on your phone.

By Karen Hao

December 11, 2019

Federated supervised learning is deployed nowadays by companies in many areas, e.g., on-device inference.

RL meets federated learning



Federated reinforcement learning: enables multiple agents to collaboratively learn a global policy without sharing datasets.

This talk

Understand the sample complexity of Q-Learning in federated settings.

Linear speedup:

Can we achieve linear speedup when learning with multiple agents?

Communication efficiency:

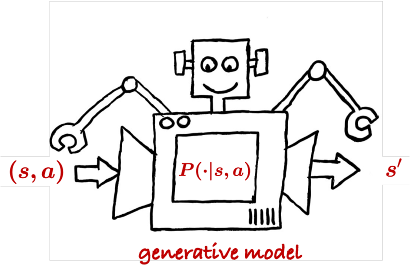
Can we perform multiple local updates to save communication?

Taming heterogeneity:

How to combine heterogeneous local updates to accelerate learning?

How to federate synchronous Q-learning?

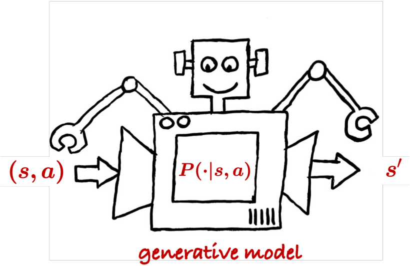
Synchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$

$$\underbrace{Q_{t+1}(s,a) = (1-\eta)Q_t(s,a) + \eta\mathcal{T}_t(Q_t)(s,a),}_{\text{draw the transition } (s,a,s') \text{ for all } (s,a)} \quad t \geq 0$$

Synchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$

$$\underbrace{Q_{t+1}(s,a) = (1-\eta)Q_t(s,a) + \eta \mathcal{T}_t(Q_t)(s,a)}_{\text{draw the transition } (s,a,s') \text{ for all } (s,a)}, \quad t \geq 0$$

$$\mathcal{T}_t(Q)(s,a) = r(s,a) + \gamma \max_{a'} Q(s', a')$$

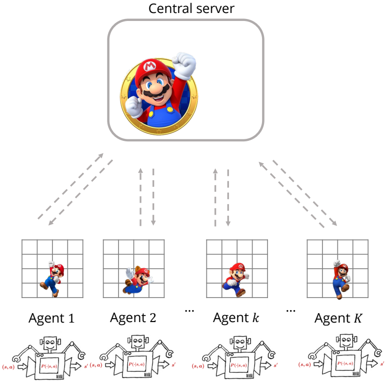
$$\mathcal{T}(Q)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{a'} Q(s', a') \right]$$

Federated synchronous Q-learning with local updates

- **The agent** k performs τ rounds of local Q-learning updates:

$$Q_{t+1}^k \leftarrow (1-\eta)Q_t^k + \eta\mathcal{T}_t(Q_t^k)$$

and sends it to the server.



Federated synchronous Q-learning with local updates

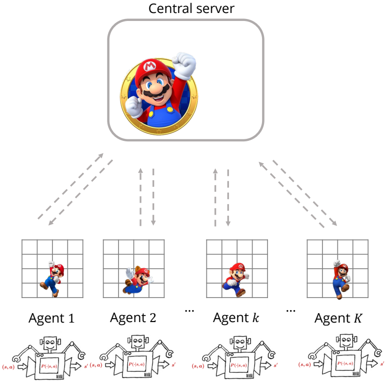
- **The agent** k performs τ rounds of local Q-learning updates:

$$Q_{t+1}^k \leftarrow (1-\eta)Q_t^k + \eta \mathcal{T}_t(Q_t^k)$$

and sends it to the server.

- **The server** averages the local updates and communicates it back to agents:

$$Q_t = \frac{1}{K}\sum_{k=1}^{K} Q_t^k$$



Federated synchronous Q-learning with local updates

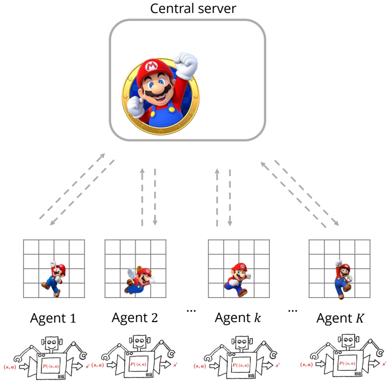
- **The agent** k performs τ rounds of local Q-learning updates:

$$Q_{t+1}^k \leftarrow (1-\eta)Q_t^k + \eta \mathcal{T}_t(Q_t^k)$$

and sends it to the server.

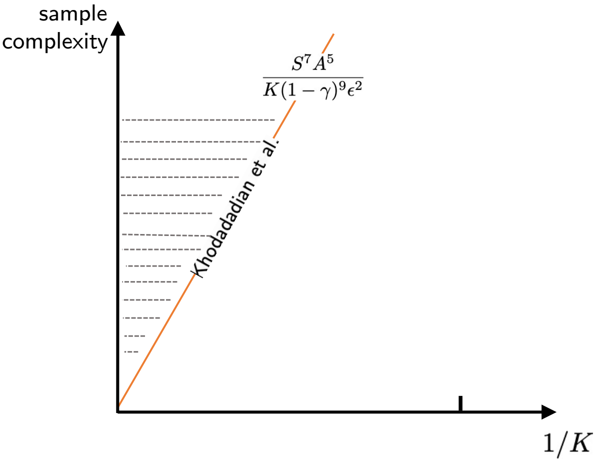
- **The server** averages the local updates and communicates it back to agents:

$$Q_t = \frac{1}{K}\sum_{k=1}^{K} Q_t^k$$



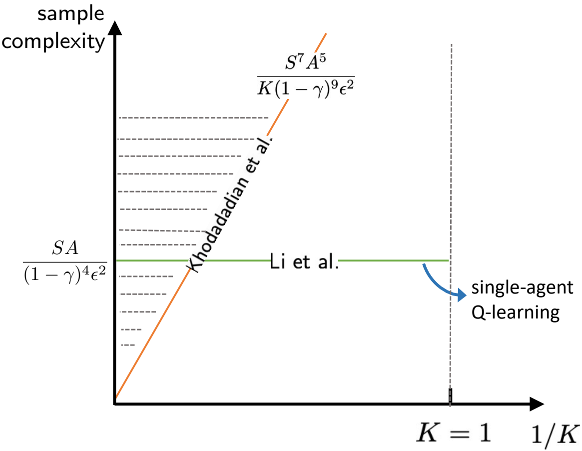
Can we achieve faster convergence, i.e. linear speedup, with low communication complexity?

Prior art



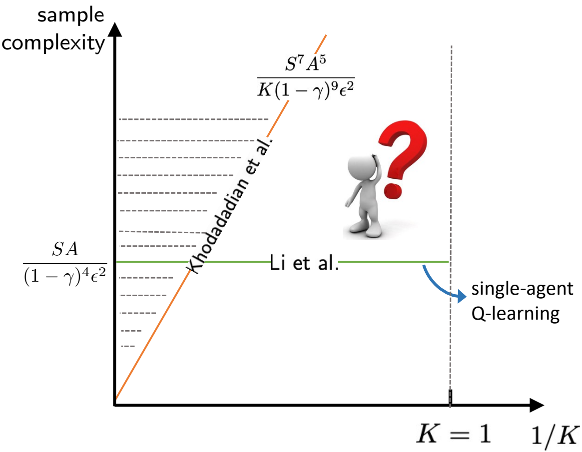
sample complexity
$\frac{S^7 A^5}{K(1-\gamma)^9 \epsilon^2}$
Khodadadian et al.
$1/K$

Prior art



The benefit of linear speedup only becomes effective $K \gg \frac{S^6 A^4}{(1-\gamma)^5}$

Prior art



Can we improve the dependency on the salient parameters?

Our theorem

Theorem (Woo, Joshi, Chi, ICML 2023)

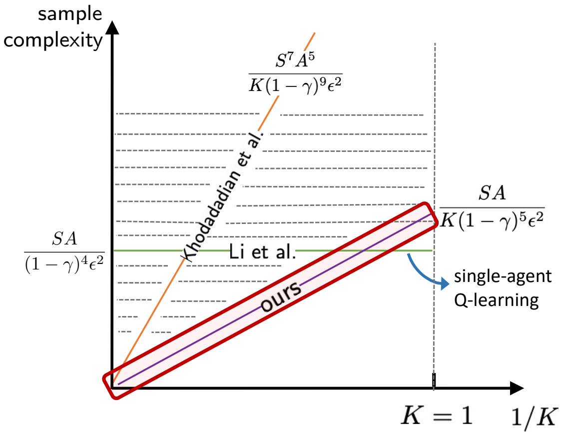
For any $0 < \epsilon \leq \frac{1}{1-\gamma}$, *federated synchronous Q-learning yields* $\|\widehat{Q} - Q^{\star}\|_{\infty} \leq \epsilon$ *with sample complexity at most*

$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{K(1-\gamma)^5\epsilon^2}\right)$$

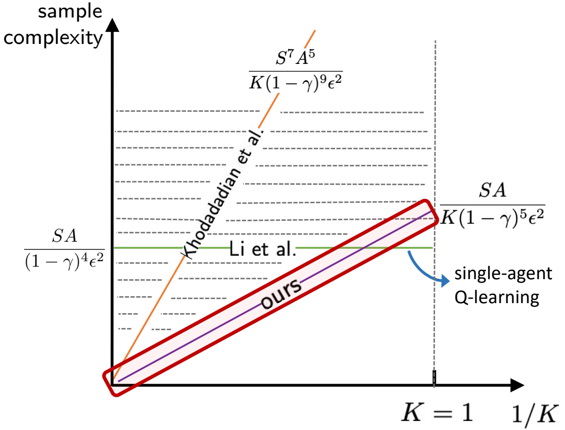
as long as $\tau - 1 \leq \frac{1}{\eta}\min\left\{\frac{1-\gamma}{8\gamma}, \frac{1}{K}\right\}$ *and* $\eta = \widetilde{O}(K(1-\gamma)^4\epsilon^2)$.

- **Communication efficiency:** when $K \gtrsim \frac{1}{1-\gamma}$ and $\epsilon \lesssim \frac{1}{K(1-\gamma)^2}$, choosing $\tau \asymp \frac{1}{K^2(1-\gamma)^4\epsilon^2}$ leads to ϵ-independent communication complexity $T/\tau = \widetilde{O}\left(\frac{K}{1-\gamma}\right)$.

Comparison with prior art

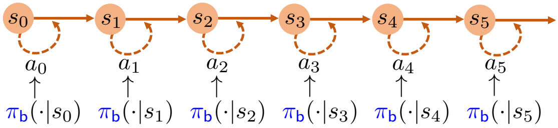


Comparison with prior art



Linear speedup with near-optimal parameter dependencies!

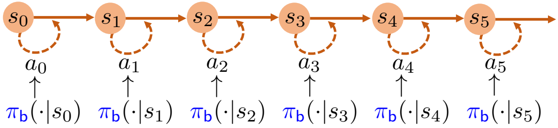
Asynchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$ using samples collected from a behavior policy π_{b}:

$$\underbrace{Q_{t+1}(s_t,a_t) = (1-\eta)Q_t(s_t,a_t) + \eta \mathcal{T}_t(Q_t)(s_t,a_t)}_{\textit{only} \text{ update } (s_t,a_t)\text{-th entry}}, \quad t \geq 0$$

Asynchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$ using samples collected from a behavior policy π_b:

$$\underbrace{Q_{t+1}(s_t, a_t) = (1-\eta)Q_t(s_t, a_t) + \eta \mathcal{T}_t(Q_t)(s_t, a_t)}_{\textit{only} \text{ update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

$$\mathcal{T}_t(Q)(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

$$\mathcal{T}(Q)(s, a) = r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \big[\max_{a'} Q(s', a')\big]$$

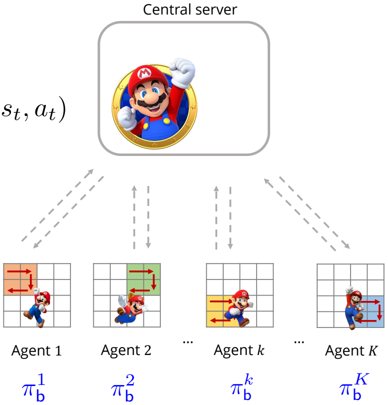
How to federate asynchronous Q-learning?

Federated asynchronous Q-learning with local updates

- **The agent** k performs τ rounds of local Q-learning updates:

$$Q^k_{t+1}(s_t, a_t) \leftarrow (1-\eta) Q^k_t(s_t, a_t) + \eta \mathcal{T}_t(Q^k_t)(s_t, a_t)$$

and sends it to the server.



Federated asynchronous Q-learning with local updates

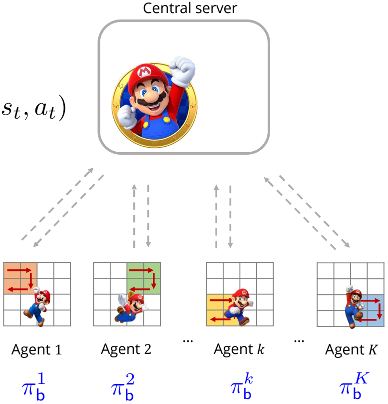
- **The agent** k performs τ rounds of local Q-learning updates:

$$Q_{t+1}^k(s_t, a_t) \leftarrow (1-\eta)Q_t^k(s_t, a_t) + \eta \mathcal{T}_t(Q_t^k)(s_t, a_t)$$

  and sends it to the server.

- **The server** averages the local updates and communicates it back to agents:

$$Q_t = \frac{1}{K}\sum_{k=1}^{K} Q_t^k$$



Federated asynchronous Q-learning with local updates

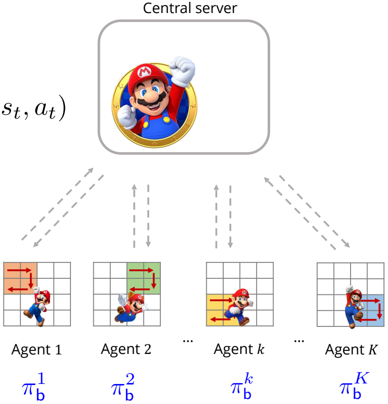
- **The agent** k performs τ rounds of local Q-learning updates:

$$Q_{t+1}^k(s_t, a_t) \leftarrow (1-\eta)Q_t^k(s_t, a_t) + \eta \mathcal{T}_t(Q_t^k)(s_t, a_t)$$

and sends it to the server.

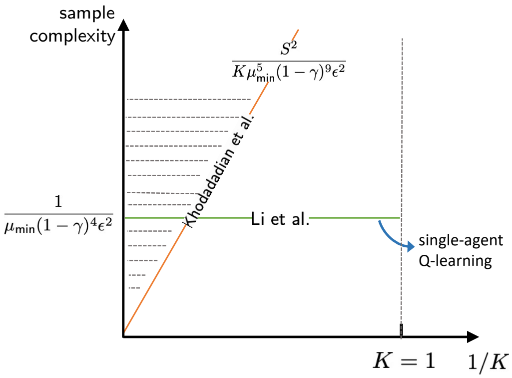
- **The server** averages the local updates and communicates it back to agents:

$$Q_t = \frac{1}{K}\sum_{k=1}^{K} Q_t^k$$



Can we achieve faster convergence with heterogeneous local behavior policies with low communication complexity?

Prior art

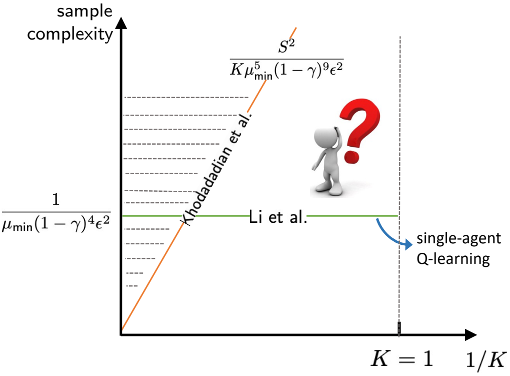


Key quantity: minimum state-action occupancy probability

$$\mu_{\min} := \min_{i,s,a} \underbrace{\mu_{\pi_b^i}(s,a)}_{\text{stationary distribution}}$$

The benefit of linear speedup only becomes effective $K \gg \frac{S^2}{\mu_{\min}^4(1-\gamma)^5}$

Prior art



Key quantity: minimum state-action occupancy probability

$$\mu_{\text{min}} := \min_{i,s,a} \underbrace{\mu_{\pi_b^i}(s,a)}_{\text{stationary distribution}}$$

Can we improve the dependency on the salient parameters?

Our theorem

Theorem (Woo, Joshi, Chi, ICML 2023)

For sufficiently small $\epsilon > 0$, federated asynchronous Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \le \epsilon$ with sample complexity at most

$$\widetilde{O}\left(\frac{C_{\mathsf{het}}}{K\mu_{\mathsf{min}}(1-\gamma)^5\epsilon^2}\right)$$

ignoring the burn-in cost that depends on the mixing times, where

$$C_{\mathsf{het}} = K \max_{k,s,a} \frac{\mu_{\mathsf{b}}^k(s,a)}{\sum_{k=1}^K \mu_{\mathsf{b}}^k(s,a)}.$$

Our theorem

Theorem (Woo, Joshi, Chi, ICML 2023)

For sufficiently small $\epsilon > 0$, federated asynchronous Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \leq \epsilon$ with sample complexity at most

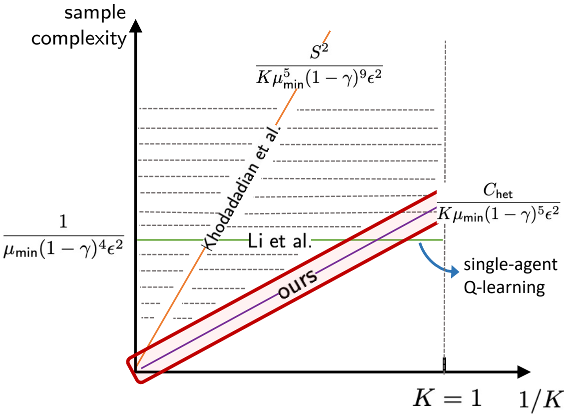
$$\widetilde{O}\left(\frac{C_{\mathsf{het}}}{K\mu_{\mathsf{min}}(1-\gamma)^5\epsilon^2}\right)$$

ignoring the burn-in cost that depends on the mixing times, where

$$C_{\mathsf{het}} = K \max_{k,s,a} \frac{\mu_{\mathsf{b}}^k(s,a)}{\sum_{k=1}^K \mu_{\mathsf{b}}^k(s,a)}.$$

- $1 \leq C_{\mathsf{het}} \leq \frac{1}{\mu_{\mathsf{min}}}$ measures the heterogeneity of local behavior policies.
- Near-optimal linear speedup when the local behavior policies are similar, $C_{\mathsf{het}} \approx 1$.

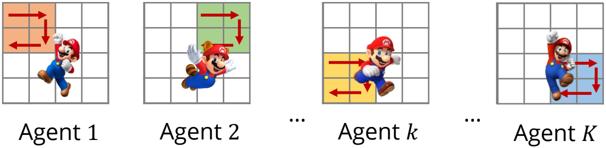
Comparison with prior art



Linear speedup with near-optimal parameter dependencies!

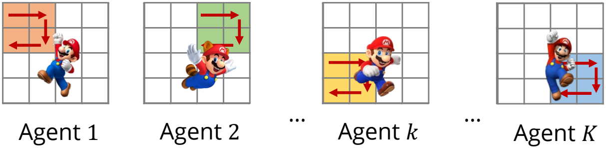
Benefit of heterogeneity?

- **Curse of heterogeneity?** performance degenerates when local behavior policies are heterogeneous (i.e. $C_{\text{het}} \gg 1$).
- **Full coverage:** require full coverage of every agent over the entire state-action space (i.e. $\mu_{\text{min}} > 0$).



Benefit of heterogeneity?

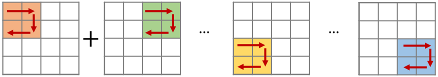
- **Curse of heterogeneity?** performance degenerates when local behavior policies are heterogeneous (i.e. $C_{\mathsf{het}} \gg 1$).
- **Full coverage:** require full coverage of every agent over the entire state-action space (i.e. $\mu_{\mathsf{min}} > 0$).



Is it possible to alleviate these requirements?

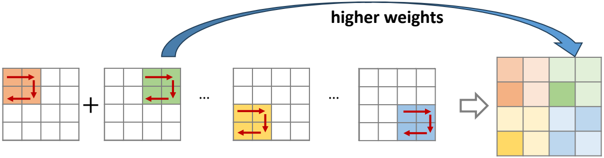
Importance averaging

Key observation: not all updates are of same quality due to limited visits induced by the behavior policy.



Importance averaging

Key observation: not all updates are of same quality due to limited visits induced by the behavior policy.



Importance averaging: the server averages the local updates based on importance via

$$Q_t(s,a) = \frac{1}{K}\sum_{k=1}^{K} \alpha_t^k(s,a) Q_t^k(s,a),$$

where

$$\alpha_t^k = \frac{(1-\eta)^{-N_{t-\tau,t}^k(s,a)}}{\sum_{k=1}^K (1-\eta)^{-N_{t-\tau,t}^k(s,a)}}, \quad N_{t-\tau,t}^k(s,a) = \text{number of visits in the sync period}.$$

Our theorem

Theorem (Jiin, Joshi, Chi, ICML 2023)

For sufficiently small $\epsilon > 0$, federated asynchronous Q-learning with importance averaging yields $\|\widehat{Q} - Q^\star\|_\infty \leq \epsilon$ with sample complexity at most

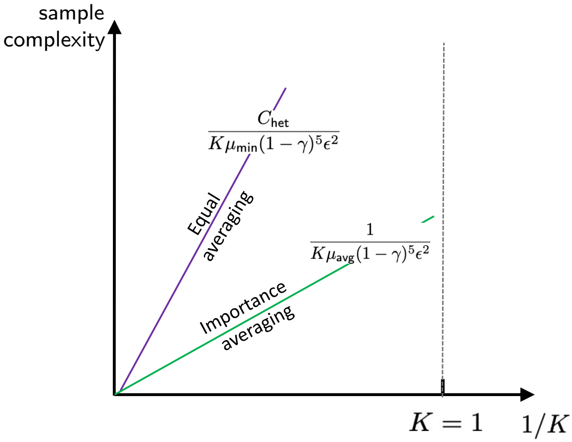
$$\widetilde{O}\left(\frac{1}{K \mu_{\mathsf{avg}} (1-\gamma)^5 \epsilon^2}\right)$$

ignoring the burn-in cost that depends on the mixing times, where

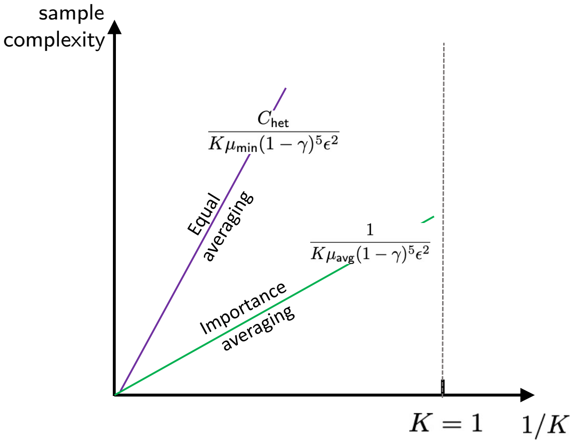
$$\mu_{\mathsf{avg}} = \min_{s,a} \frac{1}{K} \sum_{k=1}^{K} \mu_{\mathsf{b}}^k(s,a) \geq \mu_{\mathsf{min}}.$$

- Linear speedup without requiring local behavior policies to cover the entire state-action space, as long as they collectively cover the entire state-action space.

Equal averaging versus importance averaging



Equal averaging versus importance averaging



Importance averaging does not require full coverage of individual agents!

Federated and robust RL

1. Federated RL
2. Robust RL

Safety and robustness in RL

(Zhou et al., 2021; Panaganti and Kalathil, 2022; Yang et al., 2022;)

Training environment

$\neq$

Test environment

Safety and robustness in RL

(Zhou et al., 2021; Panaganti and Kalathil, 2022; Yang et al., 2022;)

Training environment

$\neq$

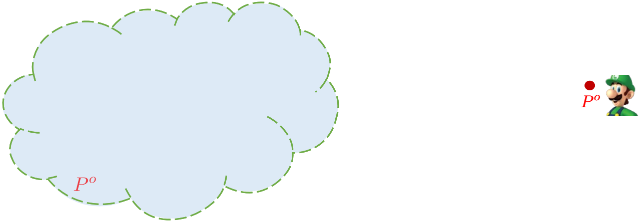
Test environment

Sim2Real Gap: Can we learn optimal policies that are robust to model perturbations?

Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

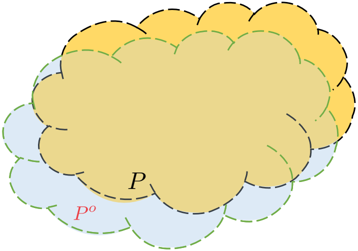
$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \leq \sigma\}$$

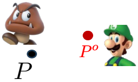


Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \le \sigma\}$$

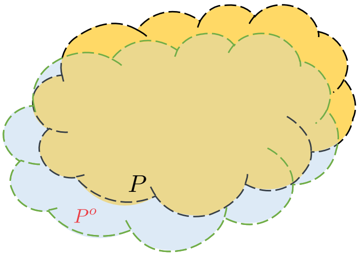


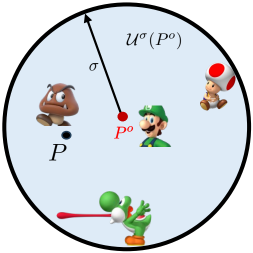


Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \le \sigma\}$$

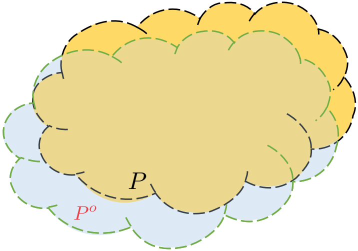


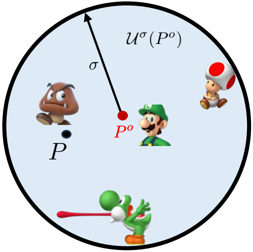


Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

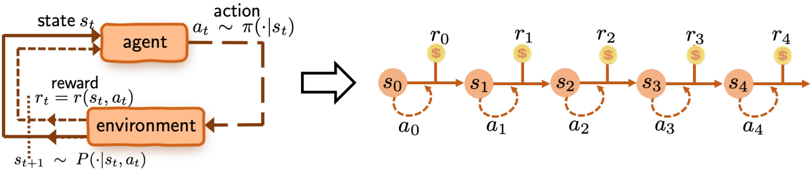
$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \leq \sigma\}$$





- Examples of ρ: f-divergence (TV, χ^2, KL...)

Robust value/Q function



Robust value/Q function of policy π:

$$\forall s \in \mathcal{S}: \qquad V^{\pi,\sigma}(s) := \inf_{P \in \mathcal{U}^\sigma(P^o)} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right]$$

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad Q^{\pi,\sigma}(s,a) := \inf_{P \in \mathcal{U}^\sigma(P^o)} \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a\right]$$

Measures the worst-case performance of the policy in the uncertainty set.

Distributionally robust MDP

Robust MDP

Find the policy $\pi^\star$ *that maximizes* $V^{\pi,\sigma}$

(Iyengar. '05, Nilim and El Ghaoui. '05)

Distributionally robust MDP

Robust MDP

Find the policy $\pi^\star$ *that maximizes* $V^{\pi,\sigma}$

(Iyengar. '05, Nilim and El Ghaoui. '05)

Robust Bellman's optimality equation: the optimal robust policy $\pi^\star$ and optimal robust value $V^{\star,\sigma} := V^{\pi^\star,\sigma}$ satisfy

$$Q^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{P_{s,a} \in \mathcal{U}^\sigma(P^o_{s,a})} \langle P_{s,a}, V^{\star,\sigma} \rangle ,$$

$$V^{\star,\sigma}(s) = \max_a Q^{\star,\sigma}(s,a)$$

Distributionally robust MDP

Robust MDP

Find the policy $\pi^\star$ *that maximizes* $V^{\pi,\sigma}$

(Iyengar. '05, Nilim and El Ghaoui. '05)

Robust Bellman's optimality equation: the optimal robust policy $\pi^\star$ and optimal robust value $V^{\star,\sigma} := V^{\pi^\star,\sigma}$ satisfy

$$Q^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{P_{s,a} \in \mathcal{U}^\sigma(P^o_{s,a})} \left\langle P_{s,a}, V^{\star,\sigma} \right\rangle,$$

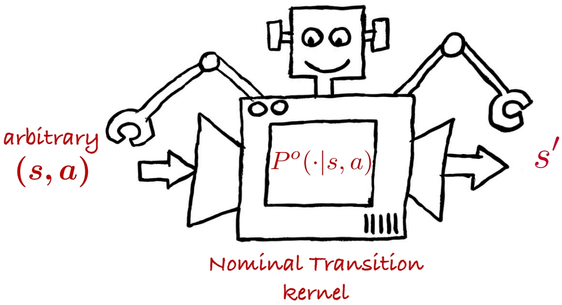
$$V^{\star,\sigma}(s) = \max_a Q^{\star,\sigma}(s,a)$$

Distributionally robust value iteration (DRVI):

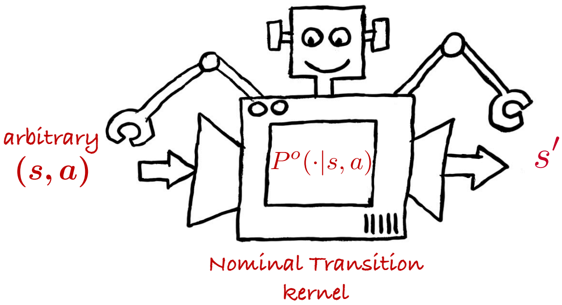
$$Q(s,a) \leftarrow r(s,a) + \gamma \inf_{P_{s,a} \in \mathcal{U}^\sigma(P^o_{s,a})} \langle P_{s,a}, V \rangle,$$

where $V(s) = \max_a Q(s,a)$.

Learning distributionally robust MDPs



Learning distributionally robust MDPs

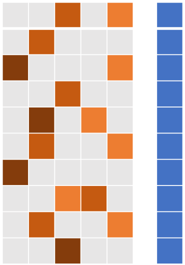


Goal of robust RL: given $\mathcal{D} := \{(s_i, a_i, s_i')\}_{i=1}^N$ from the *nominal* environment P^0, find an ϵ-optimal robust policy $\widehat{\pi}$ obeying

$$V^{\star,\sigma} - V^{\widehat{\pi},\sigma} \leq \epsilon$$

— *in a sample-efficient manner*

A curious question

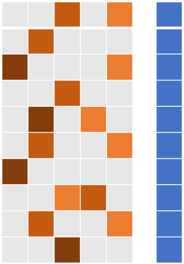


empirical MDP

Learn the optimal policy of the nominal MDP?

Learn the **robust** policy around the nominal MDP?

A curious question



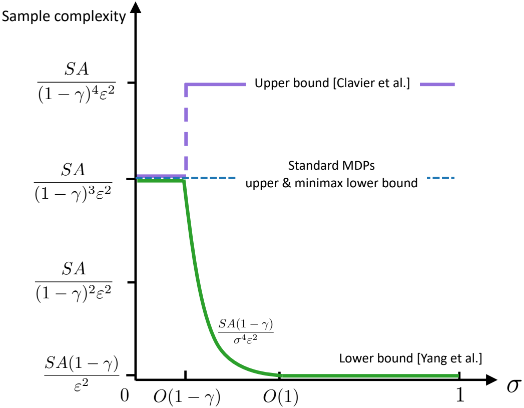
empirical MDP

Learn the optimal policy of the nominal MDP?

Learn the **robust** policy around the nominal MDP?

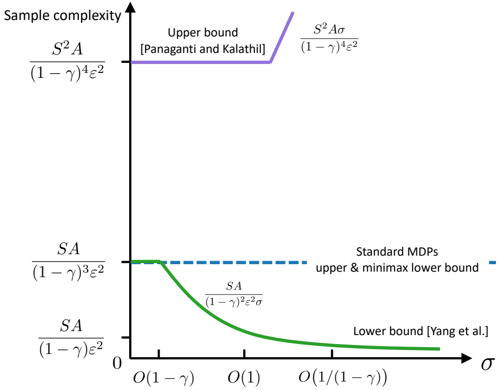
Robustness-statistical trade-off? Is there a statistical premium that one needs to pay in quest of additional robustness?

Prior art: TV uncertainty



- Large gaps between existing upper and lower bounds
- Unclear benchmarking with standard MDP

Prior art: χ^2 uncertainty



- Large gaps between existing upper and lower bounds
- Unclear benchmarking with standard MDP

Our theorem under TV uncertainty

Theorem (Shi et al., 2023)

Assume the uncertainty set is measured via the TV distance with radius $\sigma \in [0, 1)$. For sufficiently small $\epsilon > 0$, DRVI outputs a policy $\widehat{\pi}$ that satisfies $V^{\star,\sigma} - V^{\widehat{\pi},\sigma} \leq \epsilon$ with sample complexity at most

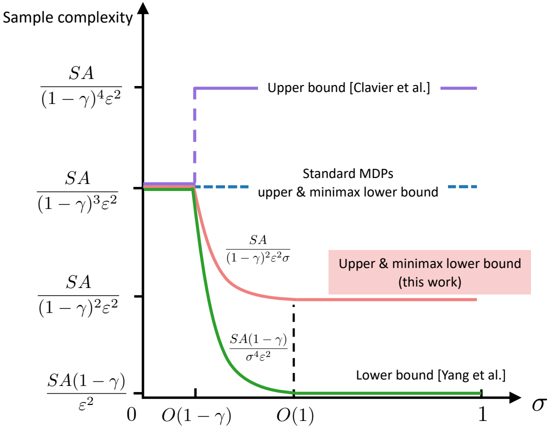
$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^2 \max\{1-\gamma, \sigma\}\epsilon^2}\right)$$

ignoring logarithmic factors. In addition, no algorithm can succeed if the sample size is below

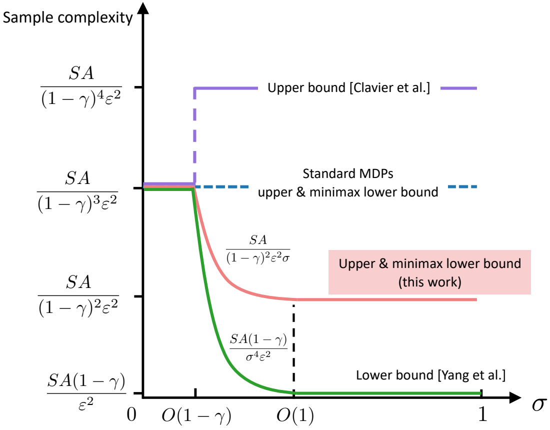
$$\widetilde{\Omega}\left(\frac{SA}{(1-\gamma)^2 \max\{1-\gamma, \sigma\}\epsilon^2}\right).$$

- Establish the minimax optimality of DRVI for RMDP under the TV uncertainty set over the full range of σ.

When the uncertainty set is TV



When the uncertainty set is TV



RMDPs are **easier** to learn than standard MDPs.

Our theorem under χ^2 uncertainty

Theorem (Upper bound, Shi et al., 2023)

Assume the uncertainty set is measured via the χ^2 *divergence with radius* $\sigma \in [0, \infty)$. *For sufficiently small* $\epsilon > 0$, *DRVI outputs a policy* $\widehat{\pi}$ *that satisfies* $V^{\star,\sigma} - V^{\widehat{\pi},\sigma} \leq \epsilon$ *with sample complexity at most*

$$\widetilde{O}\left(\frac{SA(1+\sigma)}{(1-\gamma)^4\epsilon^2}\right)$$

ignoring logarithmic factors.

Our theorem under χ^2 uncertainty

Theorem (Upper bound, Shi et al., 2023)

Assume the uncertainty set is measured via the χ^2 *divergence with radius* $\sigma \in [0, \infty)$*. For sufficiently small* $\epsilon > 0$*, DRVI outputs a policy* $\widehat{\pi}$ *that satisfies* $V^{\star,\sigma} - V^{\widehat{\pi},\sigma} \leq \epsilon$ *with sample complexity at most*

$$\widetilde{O}\left(\frac{SA(1+\sigma)}{(1-\gamma)^4\epsilon^2}\right)$$

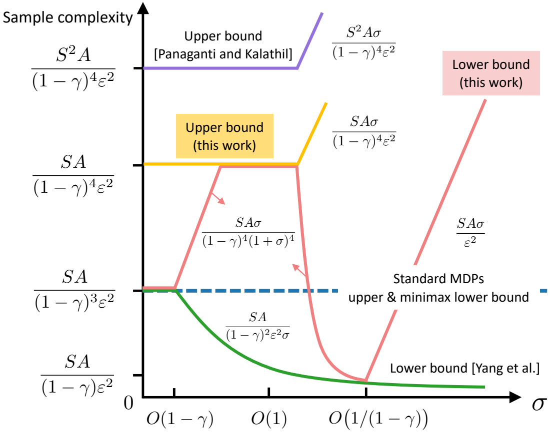
ignoring logarithmic factors.

Theorem (Lower bound, Shi et al., 2023)

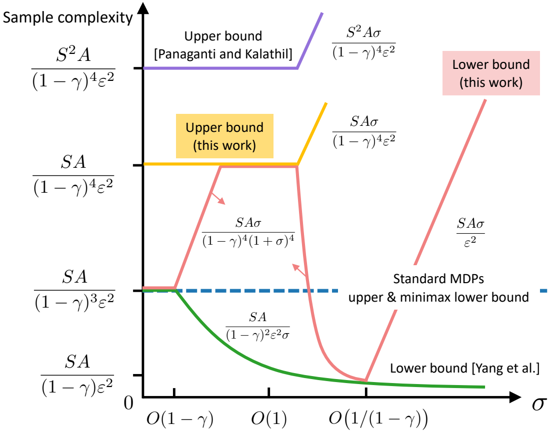
In addition, no algorithm succeeds when the sample size is below

$$\begin{cases} \widetilde{\Omega}\left(\frac{SA}{(1-\gamma)^3\epsilon^2}\right) & \text{if } \sigma \lesssim 1-\gamma \\ \widetilde{\Omega}\left(\frac{\sigma SA}{\min\{1,(1-\gamma)^4(1+\sigma)^4\}\epsilon^2}\right) & \text{otherwise} \end{cases}$$

When the uncertainty set is χ^2 divergence



When the uncertainty set is χ^2 divergence



RMDPs can be **harder** to learn than standard MDPs.

Reference I

- "*The Blessing of Heterogeneity in Federated Q-learning: Linear Speedup and Beyond*", J. Woo, G. Joshi, Y. Chi, ICML 2023.
- "*Robust dynamic programming*", Iyengar, *Mathematics of Operations Research*, 2005.
- "*Robust control of Markov decision processes with uncertain transition matrices*", Nilim and El Ghaoui, *Operations Research*, 2005.
- "*The Curious Price of Distributional Robustness in Reinforcement Learning with a Generative Model,*", L. Shi, G. Li, Y. Wei, Y. Chen, M. Geist, Y. Chi, arXiv preprint arXiv:2305.16589.

Statistical and Algorithmic Foundations of Reinforcement Learning (Part 4)

Yuejie Chi

Carnegie Mellon University

JSM
August 2023

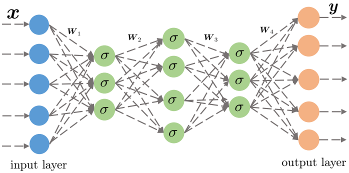
Policy optimization and Markov game

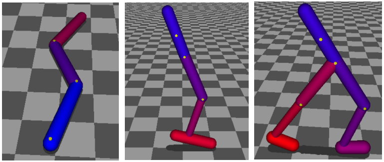
1. Policy optimization
2. Markov game

Policy optimization in practice

$$\text{maximize}_\theta \quad \text{value}(\text{policy}(\theta))$$

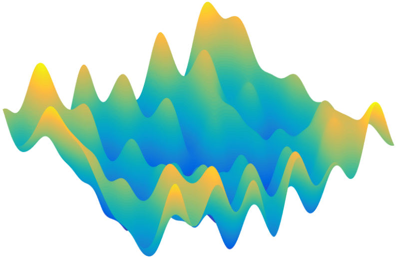
- directly optimize the policy, which is the quantity of interest;
- allow flexible differentiable parameterizations of the policy;
- work with both continuous and discrete problems.





Theoretical challenges: non-concavity

Little understanding on the global convergence of policy gradient methods until very recently, e.g. (Fazel et al., 2018; Bhandari and Russo, 2019; Agarwal et al., 2019; Mei et al. 2020), and many more.

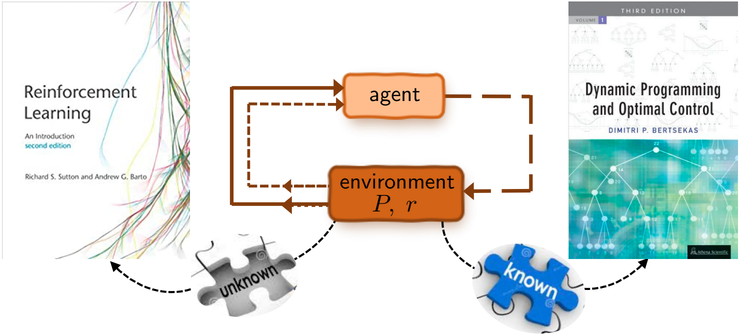


Our goal:

- understand finite-time convergence rates of popular heuristics;
- design fast-convergent algorithms that scale for finding policies with desirable properties.

Backgrounds: policy optimization in tabular Markov decision processes

Searching for the optimal policy



Goal: find the optimal policy $\pi^\star$ that maximize $V^\pi(s)$

- optimal value / Q function: $V^\star := V^{\pi^\star}$, $Q^\star := Q^{\pi^\star}$

Policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_\pi \quad V^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]$$

Policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_{\pi} \quad V^{\pi}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi}(s)\right]$$

$$\Downarrow$$

Parameterization:

$$\pi := \pi_{\theta}$$

Policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_\pi \quad V^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]$$

$$\Downarrow$$

Parameterization:

$$\pi := \pi_\theta$$

$$\text{maximize}_\theta \quad V^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi_\theta}(s)\right]$$

Policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_{\pi} \quad V^{\pi}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi}(s)\right]$$

$$\Downarrow$$

Parameterization:

$$\pi := \pi_\theta$$

$$\text{maximize}_{\theta} \quad V^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi_\theta}(s)\right]$$

Policy gradient method (Sutton et al., 2000)

For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate.*

The policy gradient theorem

Theorem (Policy gradient theorem, Sutton et al., 2000)

The policy gradient can be evaluated via

$$\begin{aligned}\nabla_\theta V^{\pi_\theta}(\rho) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a|s)\Big] \\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a|s)\Big],\end{aligned}$$

where

- $d_\rho^{\pi_\theta}$ *is the discounted state visitation distribution,*
- $\psi_\theta(s,a) := \nabla\log\pi_\theta(a|s)$ *is the score function, and*
- $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ *is the advantage function.*

Provides a general scheme for policy gradient evaluation (e.g., REINFORCE).

Softmax policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_\pi \quad V^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]$$

$$\Downarrow$$

softmax parameterization:
$$\pi_\theta(a|s) \propto \exp(\theta(s,a))$$

$$\text{maximize}_\theta \quad V^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi_\theta}(s)\right]$$

Policy gradient method (Sutton et al., 2000)

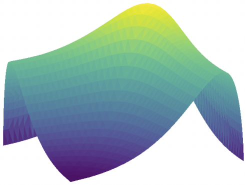
For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate.*

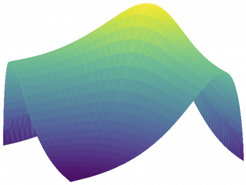
Finite-time global convergence guarantees

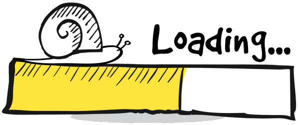
Global convergence of the PG method?



- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.

Global convergence of the PG method?

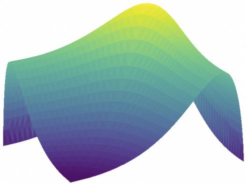


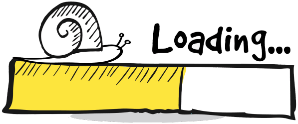


- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.
- (Mei et al., 2020) Softmax PG converges to global opt in

$O\left(\frac{1}{\epsilon}\right)$ iterations

Global convergence of the PG method?

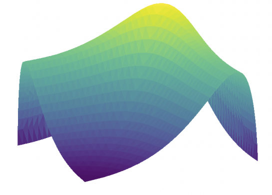


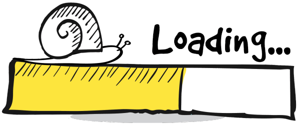


- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.
- (Mei et al., 2020) Softmax PG converges to global opt in

$$c\left(|\mathcal{S}|, |\mathcal{A}|, \frac{1}{1-\gamma}, \cdots\right) O\left(\frac{1}{\epsilon}\right) \text{ iterations}$$

Global convergence of the PG method?





- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.
- (Mei et al., 2020) Softmax PG converges to global opt in

$$c\Big(|\mathcal{S}|, |\mathcal{A}|, \frac{1}{1-\gamma}, \cdots\Big)\, O\Big(\frac{1}{\epsilon}\Big) \text{ iterations}$$

Is the rate of PG good, bad or ugly?

A negative message

Theorem (Li, Wei, Chi, Chen, 2021)

There exists an MDP s.t. it takes softmax PG at least

$$\frac{1}{\eta}\,|\mathcal{S}|^{2^{\Theta\left(\frac{1}{1-\gamma}\right)}} \text{ iterations}$$

to achieve $\|V^{(t)} - V^{\star}\|_{\infty} \leq 0.15.$

A negative message

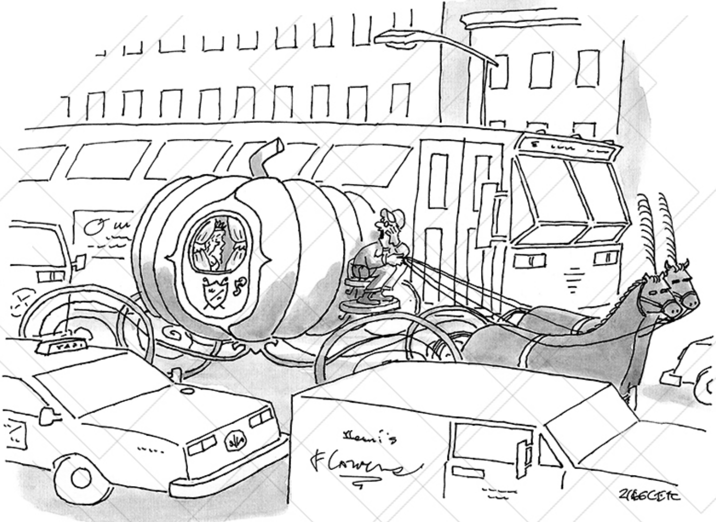
Theorem (Li, Wei, Chi, Chen, 2021)

There exists an MDP s.t. it takes softmax PG at least

$$\frac{1}{\eta}\,|\mathcal{S}|^{2^{\Theta(\frac{1}{1-\gamma})}} \text{ iterations}$$

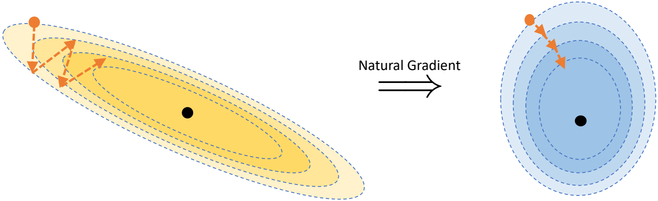
to achieve $\|V^{(t)} - V^{\star}\|_{\infty} \leq 0.15$.

- Softmax PG can take (super)-exponential time to converge (in problems w/ large state space & long effective horizon)!
- Also hold for average sub-opt gap $\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\big[V^{(t)}(s) - V^{\star}(s)\big]$.



"Seriously, lady, at this hour you'd make a lot better time taking the subway."

Booster #1: natural policy gradient



Natural policy gradient (NPG) method (Kakade, 2002)

For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate and* $\mathcal{F}_\rho^\theta$ *is the Fisher information matrix:*

$$\mathcal{F}_\rho^\theta := \mathbb{E}\left[\left(\nabla_\theta \log \pi_\theta(a|s)\right)\left(\nabla_\theta \log \pi_\theta(a|s)\right)^\top\right].$$

Connection with TRPO/PPO

TRPO/PPO (Schulman et al., 2015; 2017) are popular heuristics in training RL algorithms, with **KL regularization**

$$\mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \approx \frac{1}{2}(\theta - \theta^{(t)})^\top \mathcal{F}_\rho^\theta (\theta - \theta^{(t)})$$

via constrained or proximal terms:

$$\begin{aligned} \theta^{(t+1)} &= \operatorname*{argmax}_\theta V^{\pi_\theta^{(t)}}(\rho) + (\theta - \theta^{(t)})^\top \nabla_\theta V^{\pi_\theta^{(t)}}(\rho) - \eta \mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \\ &\approx \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta^{(t)}}(\rho), \end{aligned}$$

leading to exactly NPG!

Connection with TRPO/PPO

TRPO/PPO (Schulman et al., 2015; 2017) are popular heuristics in training RL algorithms, with **KL regularization**

$$\mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \approx \frac{1}{2}(\theta - \theta^{(t)})^\top \mathcal{F}_\rho^\theta (\theta - \theta^{(t)})$$

via constrained or proximal terms:

$$\begin{aligned} \theta^{(t+1)} &= \operatorname*{argmax}_\theta V^{\pi_\theta^{(t)}}(\rho) + (\theta - \theta^{(t)})^\top \nabla_\theta V^{\pi_\theta^{(t)}}(\rho) - \eta \mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \\ &\approx \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta^{(t)}}(\rho), \end{aligned}$$

leading to exactly NPG!

NPG $\approx$ TRPO/PPO!

NPG in the tabular setting

Natural policy gradient (NPG) method (Tabular setting)

For $t = 0, 1, \cdots$, NPG updates the policy via

$$\pi^{(t+1)}(\cdot|s) \propto \underbrace{\pi^{(t)}(\cdot|s)}_{\text{current policy}} \underbrace{\exp\left(\frac{\eta Q^{(t)}(s, \cdot)}{1-\gamma}\right)}_{\text{soft greedy}}$$

where $Q^{(t)} := Q^{\pi^{(t)}}$ is the Q-function of $\pi^{(t)}$, and $\eta > 0$.

- invariant with the choice of ρ
- Reduces to policy iteration (PI) when $\eta = \infty$.

Global convergence of NPG

Theorem (Agarwal et al., 2019)

Set $\pi^{(0)}$ *as a uniform policy. For all* $t \geq 0$*, we have*

$$V^{(t)}(\rho) \geq V^{\star}(\rho) - \left(\frac{\log |\mathcal{A}|}{\eta} + \frac{1}{(1-\gamma)^2} \right) \frac{1}{t}.$$

Global convergence of NPG

Theorem (Agarwal et al., 2019)

Set $\pi^{(0)}$ *as a uniform policy. For all* $t \geq 0$, *we have*

$$V^{(t)}(\rho) \geq V^{\star}(\rho) - \left(\frac{\log |\mathcal{A}|}{\eta} + \frac{1}{(1-\gamma)^2} \right) \frac{1}{t}.$$

Implication: set $\eta \geq (1-\gamma)^2 \log |\mathcal{A}|$, we find an ϵ-optimal policy within at most

$$\frac{2}{(1-\gamma)^2 \epsilon} \text{ iterations.}$$

Global convergence of NPG

Theorem (Agarwal et al., 2019)

Set $\pi^{(0)}$ as a uniform policy. For all $t \geq 0$, we have

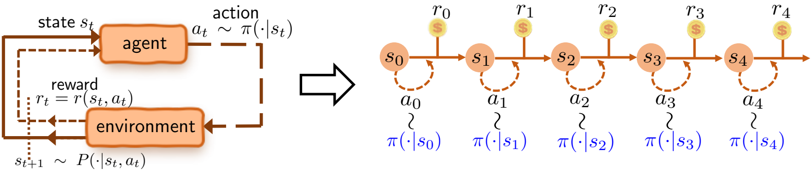
$$V^{(t)}(\rho) \geq V^{\star}(\rho) - \left(\frac{\log |\mathcal{A}|}{\eta} + \frac{1}{(1-\gamma)^2} \right) \frac{1}{t}.$$

Implication: set $\eta \geq (1-\gamma)^2 \log |\mathcal{A}|$, we find an ϵ-optimal policy within at most

$$\frac{2}{(1-\gamma)^2 \epsilon} \text{ iterations.}$$

Global convergence at a sublinear rate independent of $|\mathcal{S}|$, $|\mathcal{A}|$!

Booster #2: entropy regularization

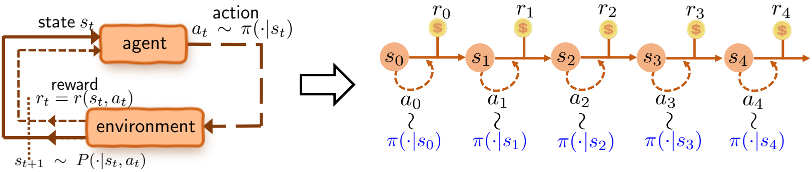


To encourage exploration, promote the stochasticity of the policy using the **"soft"** value function (Williams and Peng, 1991):

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \tau \mathcal{H}(\pi(\cdot|s_t))\Big) \,\Big|\, s_0 = s\right]$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

Booster #2: entropy regularization



To encourage exploration, promote the stochasticity of the policy using the **"soft"** value function (Williams and Peng, 1991):

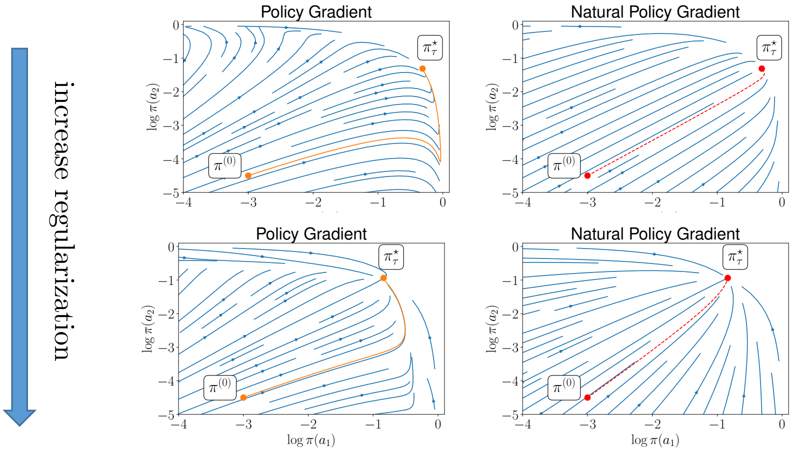
$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \big(r_t + \tau \mathcal{H}(\pi(\cdot|s_t))\big) \,\Big|\, s_0 = s\right]$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

$$\text{maximize}_\theta \quad V_\tau^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V_\tau^{\pi_\theta}(s)\right]$$

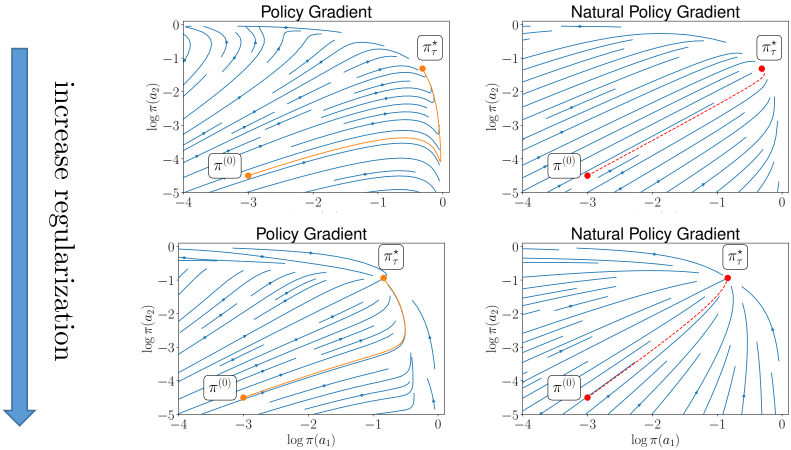
Entropy-regularized natural gradient helps!

Toy example: a bandit with 3 arms of rewards 1, 0.9 and 0.1.



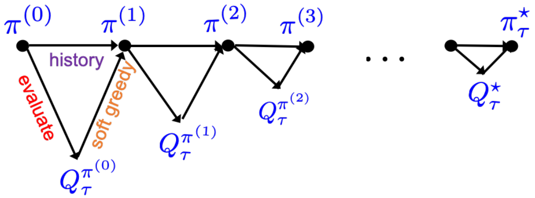
Entropy-regularized natural gradient helps!

Toy example: a bandit with 3 arms of rewards 1, 0.9 and 0.1.



Can we justify the efficacy of entropy-regularized NPG?

Entropy-regularized NPG in the tabular setting



Entropy-regularized NPG (Tabular setting)

For $t = 0, 1, \cdots$, the policy is updated via

$$\pi^{(t+1)}(\cdot|s) \propto \underbrace{\pi^{(t)}(\cdot|s)}_{\text{current policy}}{}^{1-\frac{\eta\tau}{1-\gamma}} \underbrace{\exp(Q_\tau^{(t)}(s,\cdot)/\tau)}_{\text{soft greedy}}{}^{\frac{\eta\tau}{1-\gamma}}$$

where $Q_\tau^{(t)} := Q_\tau^{\pi^{(t)}}$ is the soft Q-function of $\pi^{(t)}$, and $0 < \eta \leq \frac{1-\gamma}{\tau}$.

- invariant with the choice of ρ
- Reduces to soft policy iteration (SPI) when $\eta = \frac{1-\gamma}{\tau}$.

Linear convergence with exact gradient

Exact oracle: perfect evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$;

— *Read our paper for the inexact case!*

Linear convergence with exact gradient

Exact oracle: perfect evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$;

— *Read our paper for the inexact case!*

Theorem (Cen, Cheng, Chen, Wei, Chi, 2020)

For any learning rate $0 < \eta \leq (1-\gamma)/\tau$, the entropy-regularized NPG updates satisfy

- **Linear convergence of soft Q-functions:**

$$\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \leq C_1 \gamma \left(1 - \eta\tau\right)^t$$

for all $t \geq 0$, where $Q_\tau^\star$ is the optimal soft Q-function, and

$$C_1 = \|Q_\tau^\star - Q_\tau^{(0)}\|_\infty + 2\tau \left(1 - \frac{\eta\tau}{1-\gamma}\right) \|\log \pi_\tau^\star - \log \pi^{(0)}\|_\infty .$$

Implications

To reach $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \le \epsilon$, the iteration complexity is at most

- **General learning rates ($0 < \eta < \frac{1-\gamma}{\tau}$):**

$$\frac{1}{\eta\tau} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

- **Soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$):**

$$\frac{1}{1-\gamma} \log\left(\frac{\|Q_\tau^\star - Q_\tau^{(0)}\|_\infty \gamma}{\epsilon}\right)$$

Implications

To reach $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \le \epsilon$, the iteration complexity is at most

- **General learning rates ($0 < \eta < \frac{1-\gamma}{\tau}$):**

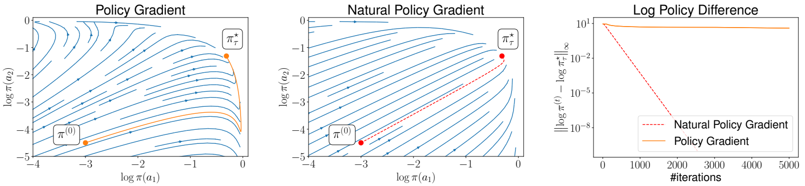
$$\frac{1}{\eta\tau} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

- **Soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$):**

$$\frac{1}{1-\gamma} \log\left(\frac{\|Q_\tau^\star - Q_\tau^{(0)}\|_\infty \gamma}{\epsilon}\right)$$

Global linear convergence of entropy-regularized NPG
at a rate independent of $|\mathcal{S}|$, $|\mathcal{A}|$!

Comparisons with entropy-regularized PG



(Mei et al., 2020) showed entropy-regularized PG achieves

$$V_\tau^\star(\rho) - V_\tau^{(t)}(\rho) \le \left(V_\tau^\star(\rho) - V_\tau^{(0)}(\rho)\right)$$

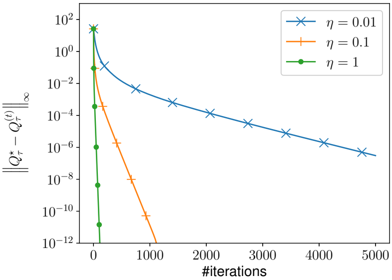
$$\cdot \exp\left(-\frac{(1-\gamma)^4 t}{(8/\tau + 4 + 8\log|\mathcal{A}|)|\mathcal{S}|}\left\|\frac{d_\rho^{\pi_\tau^\star}}{\rho}\right\|_\infty^{-1} \min_s \rho(s) \underbrace{\left(\inf_{0\le k\le t-1}\min_{s,a}\pi^{(k)}(a|s)\right)^2}_{\text{can be exponential in } |\mathcal{S}| \text{ and } \frac{1}{1-\gamma}}\right)$$

Much faster convergence of entropy-regularized NPG
at a **dimension-free** rate!

Comparison with unregularized NPG

Regularized NPG

$\tau = 0.001$

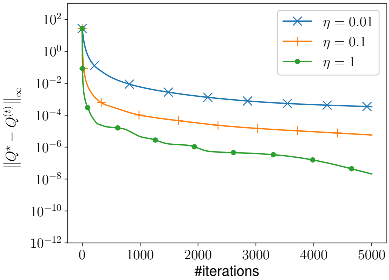


Linear rate: $\frac{1}{\eta\tau}\log\left(\frac{1}{\epsilon}\right)$

Ours

Vanilla NPG

$\tau = 0$

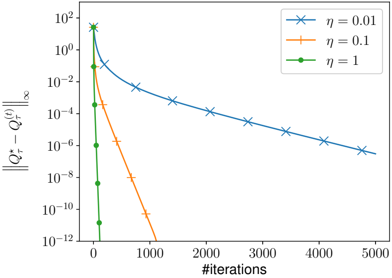


Sublinear rate: $\frac{1}{\min\{\eta,(1-\gamma)^2\}\epsilon}$

(Agarwal et al. 2019)

Regularized NPG

$\tau = 0.001$

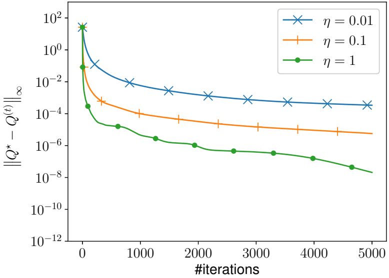


Linear rate: $\frac{1}{\eta\tau}\log\left(\frac{1}{\epsilon}\right)$

Ours

Vanilla NPG

$\tau = 0$



Sublinear rate: $\frac{1}{\min\{\eta,(1-\gamma)^2\}\epsilon}$

(Agarwal et al. 2019)

Entropy regularization enables fast convergence!

A key operator: soft Bellman operator

Soft Bellman operator

$$\mathcal{T}_\tau(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{\pi(\cdot|s')} \mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} \Big[\underbrace{Q(s',a')}_{\text{next state's value}} - \underbrace{\tau \log \pi(a'|s')}_{\text{entropy}} \Big] \right],$$

A key operator: soft Bellman operator

Soft Bellman operator

$$\mathcal{T}_\tau(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{\pi(\cdot|s')} \mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} \Big[\underbrace{Q(s',a')}_{\text{next state's value}} - \underbrace{\tau \log \pi(a'|s')}_{\text{entropy}} \Big] \right],$$

Soft Bellman equation: $Q_\tau^\star$ is *unique* solution to

$$\mathcal{T}_\tau(Q_\tau^\star) = Q_\tau^\star$$

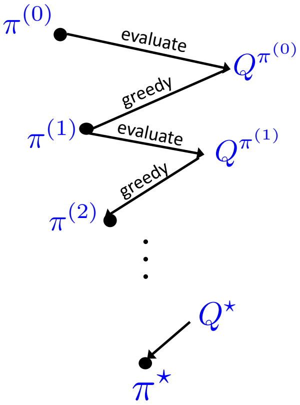
γ**-contraction of soft Bellman operator:**

$$\|\mathcal{T}_\tau(Q_1) - \mathcal{T}_\tau(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

Richard Bellman

Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

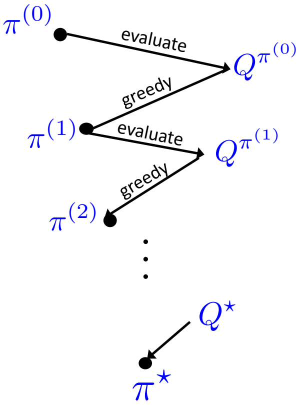
Policy iteration



Bellman operator

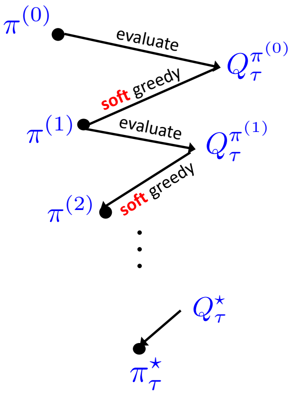
Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

Policy iteration



Bellman operator

Soft policy iteration



Soft Bellman operator

Beyond entropy regularization

Leverage regularization to promote structural properties of the learned policy.

cost-sensitive RL

weighted 1-norm

sparse exploration

Tsallis entropy

constrained and safe RL

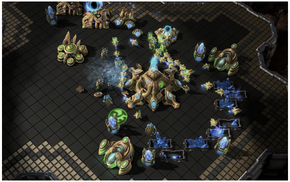
log-barrier

For further details, see: (Lan, PMD 2021) and (Zhan et al, GPMD 2021)

Policy optimization and Markov game

1. Policy optimization
2. Markov game

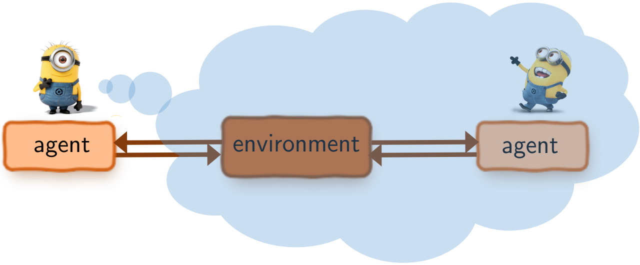
Multi-agent reinforcement learning (MARL)



To collaborate or to compete, that is the question.

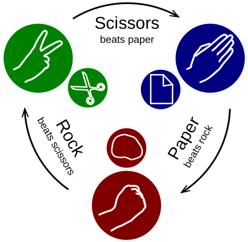
Challenges in MARL: nonstationarity

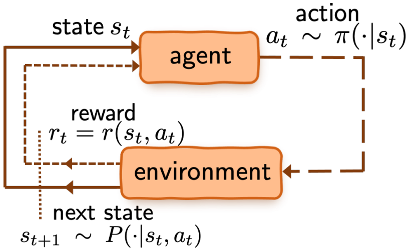
Challenges in MARL: nonstationarity



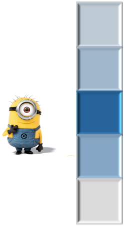
From a single-agent perspective:
the environment is **time-varying** and **nonstationary**!

MARL = Game theory + RL

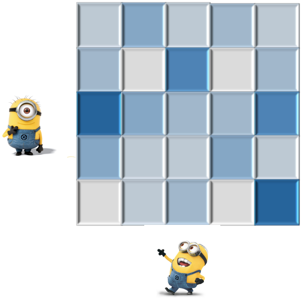




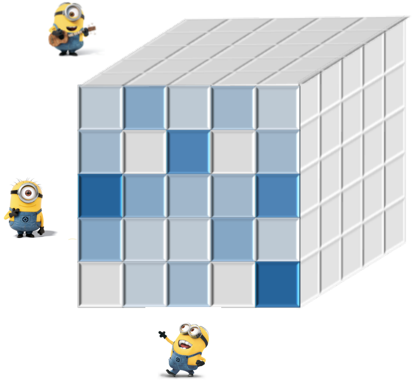
Challenges in MARL: curse of multiple agents



Challenges in MARL: curse of multiple agents



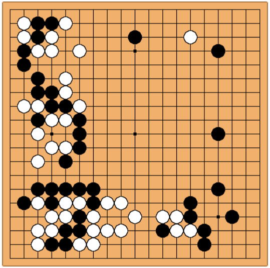
Challenges in MARL: curse of multiple agents



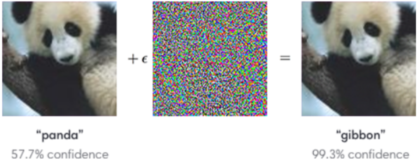
The explosion of choices:
The joint action space grows **exponentially** with the agents!

Backgrounds: two-player zero-sum Markov games

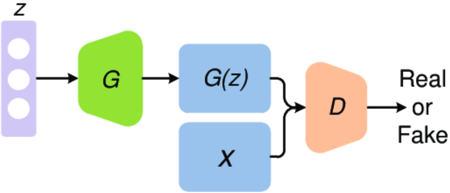
Competitive games



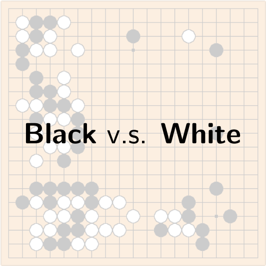
Go



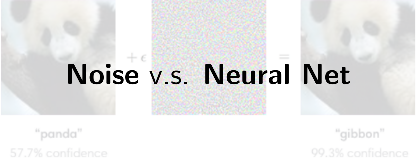
Adversarial Training



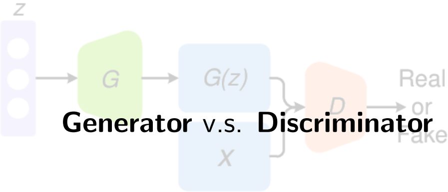
Generative Adversarial Networks



Black v.s. **White**

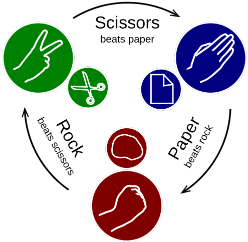


Noise v.s. **Neural Net**



Generator v.s. **Discriminator**

Zero-sum two-player matrix game



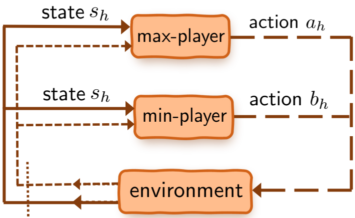
			
	0	-1	1
	1	0	-1
	-1	1	0

Zero-sum two-player matrix game

$$\max_{\mu \in \Delta(\mathcal{A})} \min_{\nu \in \Delta(\mathcal{B})} \mu^\top A \nu$$

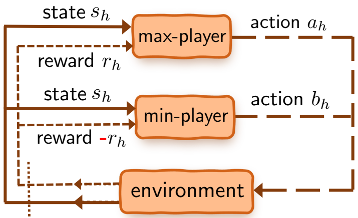
- $\mathcal{A}$, $\mathcal{B}$: action space of the two players;
- $\mu \in \Delta(\mathcal{A})$, $\nu \in \Delta(\mathcal{B})$: policies of the two players;
- $\Delta(\mathcal{A})$, $\Delta(\mathcal{B})$: set of probability distribution over $\mathcal{A}$, $\mathcal{B}$;
- $A \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{B}|}$: payoff matrix.

Two-player zero-sum Markov games (finite-horizon)



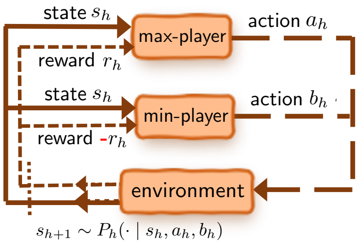
- $\mathcal{S}$: shared state space
- H: horizon
- $\mathcal{A}$: action space of max-player
- $\mathcal{B}$: action space of min-player

Two-player zero-sum Markov games (finite-horizon)



- $\mathcal{S}$: shared state space
- H: horizon
- $\mathcal{A}$: action space of max-player
- $\mathcal{B}$: action space of min-player
- immediate reward: max-player $r_h(s, a, b) \in [0, 1]$
  min-player $-r_h(s, a, b)$

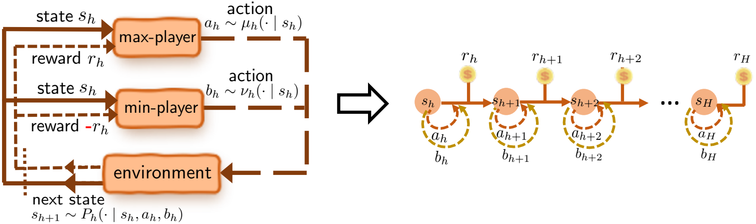
Two-player zero-sum Markov games (finite-horizon)



- $\mathcal{S}$: shared state space
- H: horizon
- $\mathcal{A}$: action space of max-player
- $\mathcal{B}$: action space of min-player
- immediate reward: max-player $r_h(s, a, b) \in [0, 1]$
  min-player $-r_h(s, a, b)$
- $P_h(\cdot \mid s, a, b)$: unknown transition probabilities

Value function of policy pair

μ: policy of max-player; ν: policy of min-player

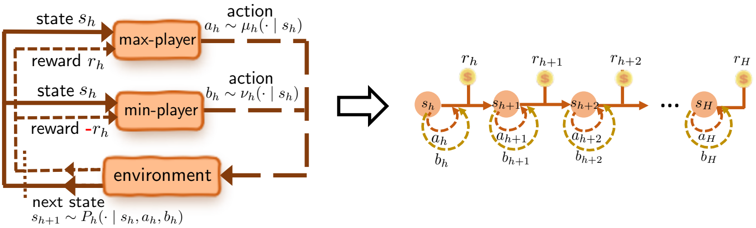


Value function of policy pair (μ, ν):

$$V^{\mu,\nu}(s) := \mathbb{E}\left[\sum_{t=1}^{H} r_t(s_t, a_t, b_t) \,\middle|\, s_1 = s\right]$$

Value function of policy pair

μ: policy of max-player; ν: policy of min-player

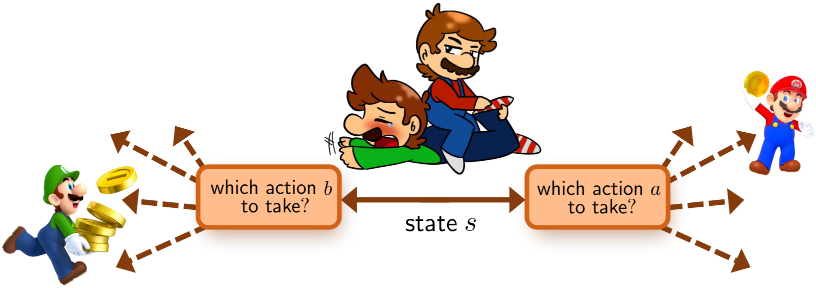


Value function of policy pair (μ, ν):

$$V^{\mu,\nu}(s) := \mathbb{E}\left[\sum_{t=1}^{H} r_t(s_t, a_t, b_t) \,\middle|\, s_1 = s\right]$$

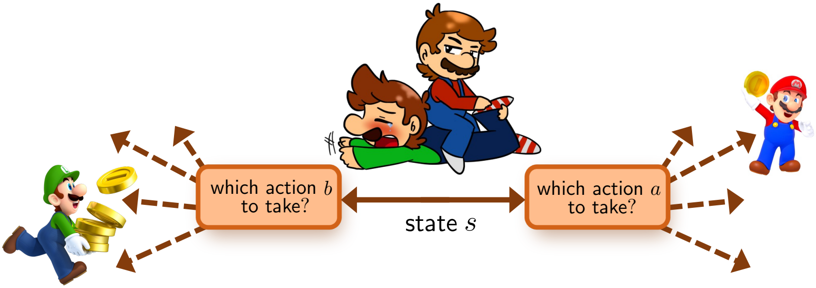
- $\{(a_t, b_t, s_{t+1})\}$: generated when max-player and min-player execute policies μ and ν *independently (i.e. no coordination)*

Target policy



- Each agent seeks **optimal policy** maximizing her own interest
- But two agents have conflicting goals ...

Target policy



- Each agent seeks **optimal policy** maximizing her own interest
- But two agents have conflicting goals . . .

Zero-sum two-player Markov game

$$\max_{\mu \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \min_{\nu \in \Delta(\mathcal{B})^{|\mathcal{S}|}} V^{\mu,\nu}(s)$$

Nash equilibrium (NE)

John von Neumann

John Nash

An NE policy pair $(\mu^\star, \nu^\star)$ obeys

$$\max_{\mu} V^{\mu,\nu^\star} = V^{\mu^\star,\nu^\star} = \min_{\nu} V^{\mu^\star,\nu}$$

Nash equilibrium (NE)

John von Neumann

John Nash

An NE policy pair $(\mu^\star, \nu^\star)$ obeys

$$\max_{\mu} V^{\mu,\nu^\star} = V^{\mu^\star,\nu^\star} = \min_{\nu} V^{\mu^\star,\nu}$$

- no unilateral deviation is beneficial

Nash equilibrium (NE)

John von Neumann

John Nash

An NE policy pair $(\mu^\star, \nu^\star)$ obeys

$$\max_{\mu} V^{\mu,\nu^\star} = V^{\mu^\star,\nu^\star} = \min_{\nu} V^{\mu^\star,\nu}$$

- no unilateral deviation is beneficial
- no coordination between two agents (they act *independently*)

Nash equilibrium (NE)

John von Neumann

John Nash

An ϵ-NE policy pair $(\widehat{\mu}, \widehat{\nu})$ obeys

$$\max_{\mu} V^{\mu,\widehat{\nu}} - \epsilon \leq V^{\widehat{\mu},\widehat{\nu}} \leq \min_{\nu} V^{\widehat{\mu},\nu} + \epsilon$$

- no unilateral deviation is beneficial
- no coordination between two agents (they act *independently*)

Nash value iteration (finite-horizon)

Nash value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \longleftarrow r_h(s,a,b) + \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu(s)} \min_{\nu(s)} \mu(s')^\top Q_{h+1}(s') \nu(s')}_{\text{matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

- The matrix game can be solved efficiently.
- Requires knowledge of the transition kernel $P_h(\cdot|s,a,b)$.

Nash value iteration (finite-horizon)

Nash value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \longleftarrow r_h(s,a,b) + \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu(s)} \min_{\nu(s)} \mu(s')^\top Q_{h+1}(s') \nu(s')}_{\text{matrix game}} \Bigg],$$

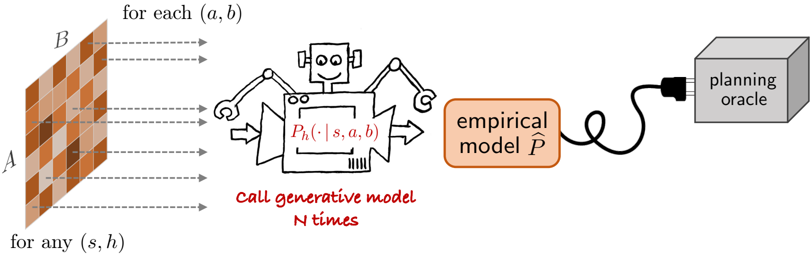
where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

- The matrix game can be solved efficiently.
- Requires knowledge of the transition kernel $P_h(\cdot|s,a,b)$.

How do we learn the NE without access to the model in a statistically efficient manner?

Model-based approach w/ non-adaptive sampling

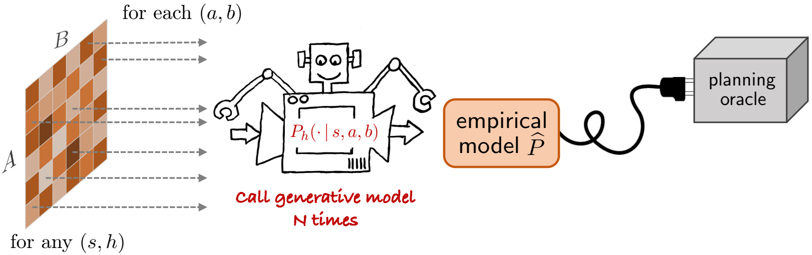
(Zhang et al., 2020)



1. for each (s, a, b, h), call generative models N times

Model-based approach w/ non-adaptive sampling

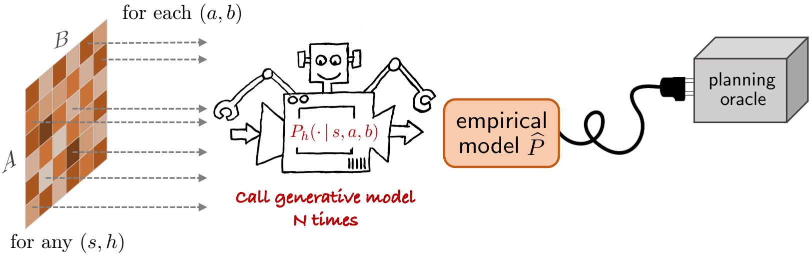
(Zhang et al., 2020)



1. for each (s, a, b, h), call generative models N times
2. build empirical model $\widehat{P}$

Model-based approach w/ non-adaptive sampling

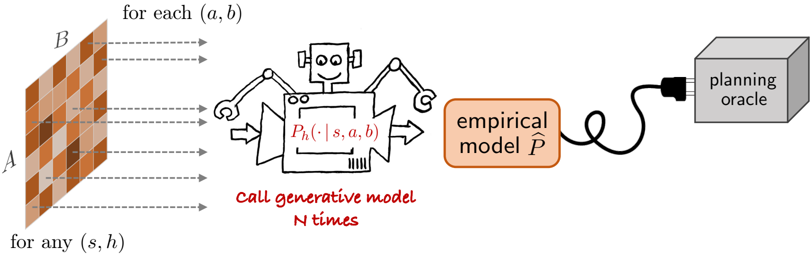
(Zhang et al., 2020)



1. for each (s, a, b, h), call generative models N times
2. build empirical model $\widehat{P}$, and run classical planning algorithms

Model-based approach w/ non-adaptive sampling

(Zhang et al., 2020)

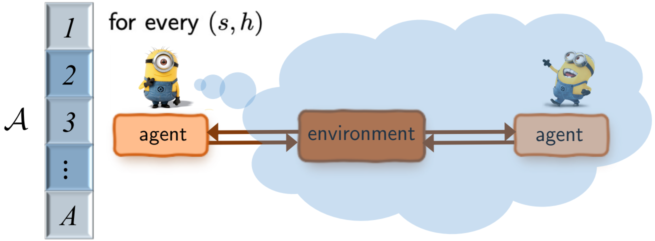


1. for each (s, a, b, h), call generative models N times
2. build empirical model $\widehat{P}$, and run classical planning algorithms

sample complexity: $\frac{H^4 SAB}{\epsilon^2}$

Breaking the curse of multi-agents?

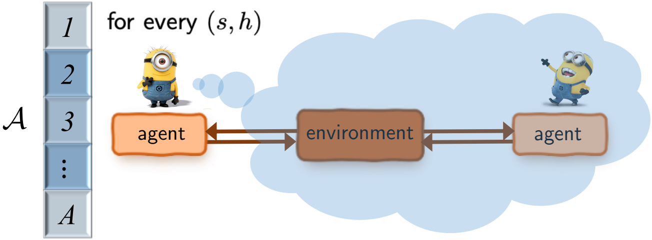
(Song, Mei, Bai, 2021; Jin et al., 2021; Basar et al., 2021)



V-learning (online setting): MARL meets adversarial learning:
for the max-player, for $h = 1, \dots, H$

Breaking the curse of multi-agents?

(Song, Mei, Bai, 2021; Jin et al., 2021; Basar et al., 2021)

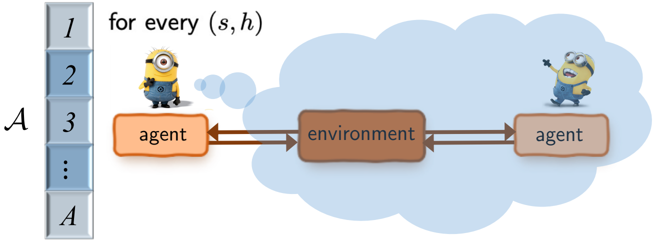


V-learning (online setting): MARL meets adversarial learning:
for the max-player, for $h = 1, \ldots, H$

1. *adaptive sampling:* sampling $\mathcal{A}$ based on $\mu_h(\cdot|s)$

Breaking the curse of multi-agents?

(Song, Mei, Bai, 2021; Jin et al., 2021; Basar et al., 2021)

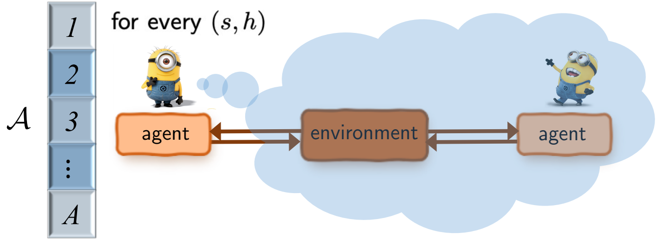


V-learning (online setting): MARL meets adversarial learning:
for the max-player, for $h = 1, \ldots, H$

1. *adaptive sampling:* sampling $\mathcal{A}$ based on $\mu_h(\cdot|s)$
2. estimate V-function only with *Hoeffding bonus* (of size S)

Breaking the curse of multi-agents?

(Song, Mei, Bai, 2021; Jin et al., 2021; Basar et al., 2021)

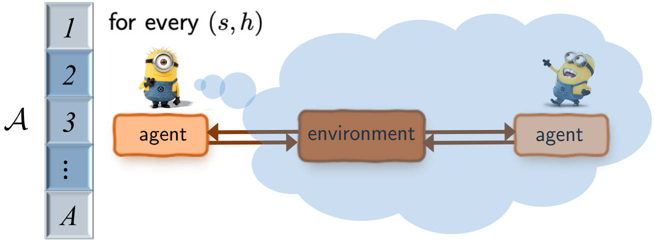


V-learning (online setting): MARL meets adversarial learning:
for the max-player, for $h = 1, \ldots, H$

1. *adaptive sampling:* sampling $\mathcal{A}$ based on $\mu_h(\cdot|s)$
2. estimate V-function only with *Hoeffding bonus* (of size S)
3. update policy via *adversarial learning subroutine*, e.g. FTRL

Breaking the curse of multi-agents?

(Song, Mei, Bai, 2021; Jin et al., 2021; Basar et al., 2021)

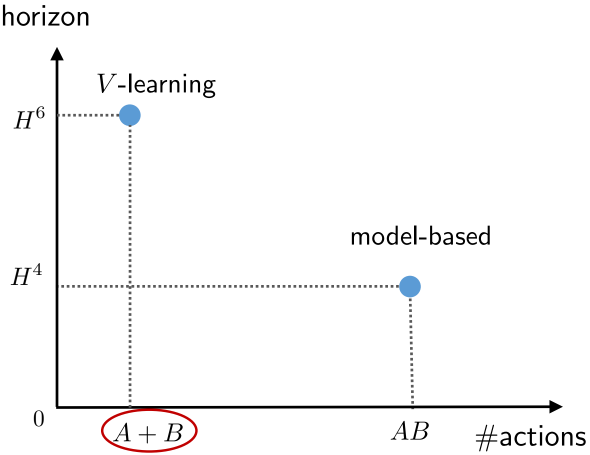


V-learning (online setting): MARL meets adversarial learning:
for the max-player, for $h = 1, \ldots, H$

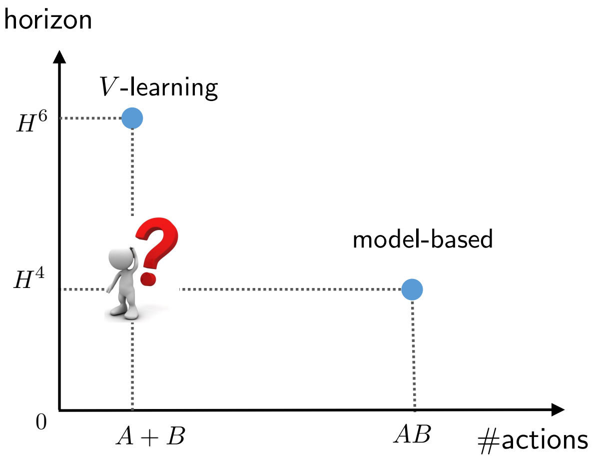
1. *adaptive sampling:* sampling $\mathcal{A}$ based on $\mu_h(\cdot|s)$
2. estimate V-function only with *Hoeffding bonus* (of size S)
3. update policy via *adversarial learning subroutine*, e.g. `FTRL`

sample complexity: $\frac{H^6 S(A+B)}{\epsilon^2}$

Summary of prior arts



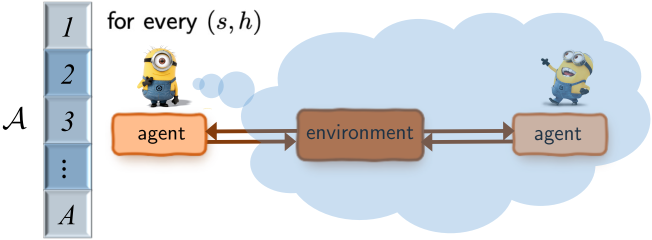
Summary of prior arts



Can we simultaneously overcome
curse of multi-agents & barrier of long horizon?

Our algorithm (with a generative model)

(Li et al., NeurIPS 2022)

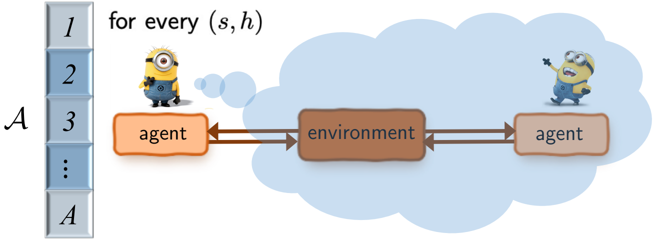


Nash-Q-FTRL (ours): for the max-player, for $h = H, \ldots, 1$

- collect $k = 1, \ldots, K$ samples:

Our algorithm (with a generative model)

(Li et al., NeurIPS 2022)

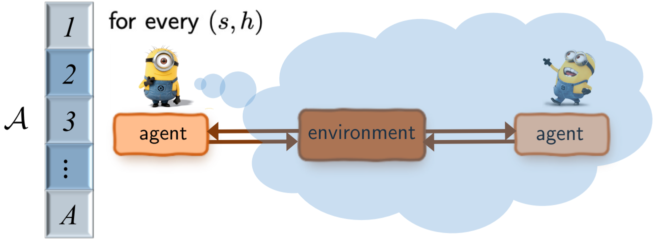


Nash-Q-FTRL (ours): for the max-player, for $h = H, \ldots, 1$

- collect $k = 1, \ldots, K$ samples:
    1. *adaptive sampling:* sample $\mathcal{A}$ based on $\mu_h^k(\cdot|s)$

Our algorithm (with a generative model)

(Li et al., NeurIPS 2022)

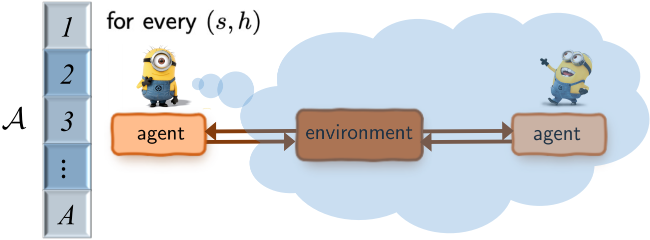


Nash-Q-FTRL (ours): for the max-player, for $h = H, \dots, 1$

- collect $k = 1, \dots, K$ samples:
  1. *adaptive sampling:* sample $\mathcal{A}$ based on $\mu_h^k(\cdot|s)$
  2. estimate single-agent Q-function $Q_h(s, \cdot)$ via Q-learning

Our algorithm (with a generative model)

(Li et al., NeurIPS 2022)

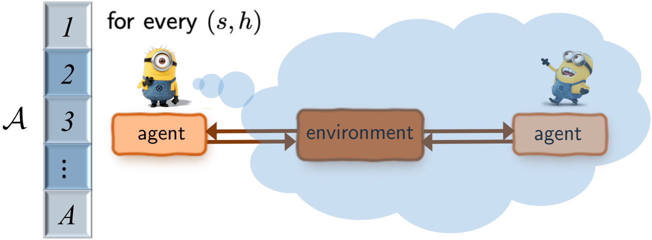


Nash-Q-FTRL (ours): for the max-player, for $h = H, \ldots, 1$

- collect $k = 1, \ldots, K$ samples:
  1. *adaptive sampling:* sample $\mathcal{A}$ based on $\mu_h^k(\cdot|s)$
  2. estimate single-agent Q-function $Q_h(s,\cdot)$ via Q-learning
  3. update policy $\mu_h^{k+1}(\cdot|s)$ via FTRL

Our algorithm (with a generative model)

(Li et al., NeurIPS 2022)



Nash-Q-FTRL (ours): for the max-player, for $h = H, \dots, 1$

- collect $k = 1, \dots, K$ samples:
  1. *adaptive sampling:* sample $\mathcal{A}$ based on $\mu_h^k(\cdot|s)$
  2. estimate single-agent Q-function $Q_h(s, \cdot)$ via Q-learning
  3. update policy $\mu_h^{k+1}(\cdot|s)$ via FTRL
- output a Markov policy μ_h and V_h with Bernstein bonuses

Main result: two-player zero-sum Markov games

Theorem (Li, Chi, Wei, Chen '22)

For any $0 < \epsilon \leq H$, the policy pair $(\widehat{\mu}, \widehat{\nu})$ returned by the proposed algorithm is ϵ-Nash, with sample complexity at most

$$\widetilde{O}\left(\frac{H^4 S(A+B)}{\epsilon^2}\right).$$

Main result: two-player zero-sum Markov games

Theorem (Li, Chi, Wei, Chen '22)

For any $0 < \epsilon \leq H$*, the policy pair* $(\widehat{\mu}, \widehat{\nu})$ *returned by the proposed algorithm is* ϵ*-Nash, with sample complexity at most*

$$\widetilde{O}\left(\frac{H^4 S(A+B)}{\epsilon^2}\right).$$

- **minimax lower bound:** $\widetilde{\Omega}\left(\frac{H^4 S(A+B)}{\epsilon^2}\right)$
- breaks curse of multi-agents & long-horizon barrier at once!

Main result: two-player zero-sum Markov games

Theorem (Li, Chi, Wei, Chen '22)

For any $0 < \epsilon \leq H$*, the policy pair* $(\widehat{\mu}, \widehat{\nu})$ *returned by the proposed algorithm is* ϵ*-Nash, with sample complexity at most*

$$\widetilde{O}\bigg(\frac{H^4 S(A+B)}{\epsilon^2}\bigg).$$

- **minimax lower bound:** $\widetilde{\Omega}\big(\frac{H^4 S(A+B)}{\epsilon^2}\big)$
- breaks curse of multi-agents & long-horizon barrier at once!
- full ϵ-range (no burn-in cost)

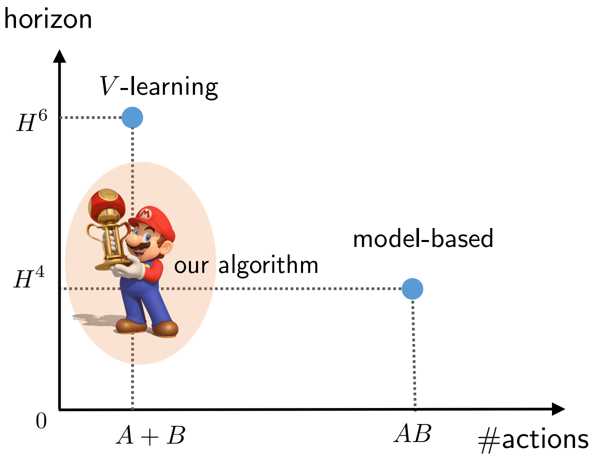
Main result: two-player zero-sum Markov games

Theorem (Li, Chi, Wei, Chen '22)

For any $0 < \epsilon \leq H$, *the policy pair* $(\widehat{\mu}, \widehat{\nu})$ *returned by the proposed algorithm is* ϵ*-Nash, with sample complexity at most*

$$\widetilde{O}\left(\frac{H^4 S(A+B)}{\epsilon^2}\right).$$

- **minimax lower bound:** $\widetilde{\Omega}\left(\frac{H^4 S(A+B)}{\epsilon^2}\right)$
- breaks curse of multi-agents & long-horizon barrier at once!
- full ϵ-range (no burn-in cost)
- other features: Markov policy, decentralized, . . .



Our algorithm breaks curses of multi-agents and long-horizon barrier simultaneously!

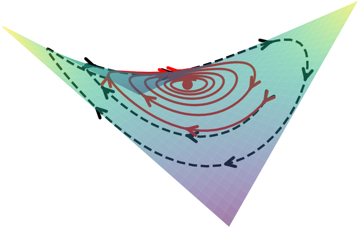
Policy optimization for games

Policy optimization: saddle-point optimization

Zero-sum two-player Markov game

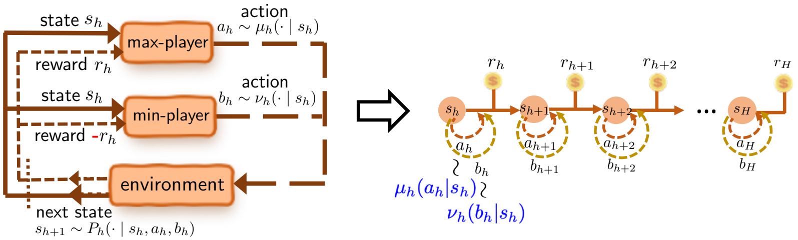
Given an initial state distribution $s \sim \rho$, *find policy* π *such that*

$$\max_{\mu \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \min_{\nu \in \Delta(\mathcal{B})^{|\mathcal{S}|}} V^{\mu,\nu}(\rho) := \mathbb{E}_{s\sim\rho}[V^{\mu,\nu}(s)]$$



Can we design a policy optimization method that guarantees fast *last-iterate* convergence?

Entropy regularization in MARL

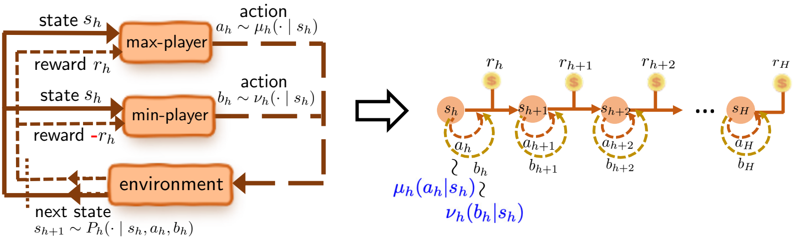


Promote the stochasticity of the policy pair using the **"soft"** value function (Williams and Peng, 1991; Cen et al., 2020):

$$V_\tau^{\mu,\nu}(s) := \mathbb{E}\left[\sum_{h=1}^{H}\left(r_t + \tau\mathcal{H}(\mu_t(\cdot|s_t)) - \tau\mathcal{H}(\nu_t(\cdot|s_t))\right) \,\middle|\, s_0 = s\right],$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

Entropy regularization in MARL



Promote the stochasticity of the policy pair using the **"soft"** value function (Williams and Peng, 1991; Cen et al., 2020):

$$V_\tau^{\mu,\nu}(s) := \mathbb{E}\left[\sum_{h=1}^{H}\left(r_t + \tau\mathcal{H}(\mu_t(\cdot|s_t) - \tau\mathcal{H}(\nu_t(\cdot|s_t))\right) \middle| \, s_0 = s\right],$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

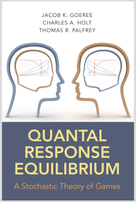
$$\max_{\mu \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \min_{\nu \in \Delta(\mathcal{B})^{|\mathcal{S}|}} V_\tau^{\mu,\nu}(\rho)$$

Quantal response equilibrium (QRE)

Quantal response equilibrium (McKelvey and Palfrey, 1995)

The quantal response equilibrium (QRE) is the policy pair $(\mu_\tau^\star, \nu_\tau^\star)$ *that is the unique solution to*

$$\max_{\mu \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \min_{\nu \in \Delta(\mathcal{B})^{|\mathcal{S}|}} V_\tau^{\mu,\nu}(\rho).$$



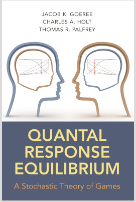
- Unlike NE, QRE assumes bounded rationality: action probability follows the logit function.

Quantal response equilibrium (QRE)

Quantal response equilibrium (McKelvey and Palfrey, 1995)

The quantal response equilibrium (QRE) is the policy pair $(\mu_\tau^\star, \nu_\tau^\star)$ *that is the unique solution to*

$$\max_{\mu \in \Delta(\mathcal{A})^{|\mathcal{S}|}} \min_{\nu \in \Delta(\mathcal{B})^{|\mathcal{S}|}} V_\tau^{\mu,\nu}(\rho).$$



- Unlike NE, QRE assumes bounded rationality: action probability follows the logit function.

Translating to an ϵ-NE: setting $\tau \asymp \widetilde{O}\left(\epsilon/H\right)$.

Soft value iteration

Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Soft value iteration

Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Soft value iteration

Soft value iteration: for $h = H, \dots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Entropy-regularized matrix game

$$\max_{\mu \in \Delta(\mathcal{A})} \min_{\nu \in \Delta(\mathcal{B})} \mu^\top A \nu + \tau \mathcal{H}(\mu) - \tau \mathcal{H}(\nu)$$

A prelude: entropy-regularized matrix game

Optimistic multiplicative weights update (OMWU) method
(Related to OMD, Rakhlin and Sridharan, 2013): for $t = 0, 1, \cdots,$

$$\text{predict}: \quad \begin{cases} \bar{\mu}^{(t+1)} \propto [\mu^{(t)}]^{1-\eta\tau} \exp\left([A\bar{\nu}^{(t)}]/\tau\right)^{\eta\tau} \\ \bar{\nu}^{(t+1)} \propto [\nu^{(t)}]^{1-\eta\tau} \exp\left(-[A^\top \bar{\mu}^{(t)}]/\tau\right)^{\eta\tau} \end{cases}$$

$$\text{update}: \quad \begin{cases} \mu^{(t+1)} \propto [\mu^{(t)}]^{1-\eta\tau} \exp\left([A\bar{\nu}^{(t+1)}]/\tau\right)^{\eta\tau} \\ \nu^{(t+1)} \propto [\nu^{(t)}]^{1-\eta\tau} \exp\left(-[A^\top \bar{\mu}^{(t+1)}]/\tau\right)^{\eta\tau} \end{cases}$$

Optimistic multiplicative weights update (OMWU) method (Related to OMD, Rakhlin and Sridharan, 2013): for $t = 0, 1, \cdots,$

$$\text{predict}: \quad \begin{cases} \bar{\mu}^{(t+1)} \propto [\mu^{(t)}]^{1-\eta\tau} \exp\left([A\bar{\nu}^{(t)}]/\tau\right)^{\eta\tau} \\ \bar{\nu}^{(t+1)} \propto [\nu^{(t)}]^{1-\eta\tau} \exp\left(-[A^\top \bar{\mu}^{(t)}]/\tau\right)^{\eta\tau} \end{cases}$$

$$\text{update}: \quad \begin{cases} \mu^{(t+1)} \propto [\mu^{(t)}]^{1-\eta\tau} \exp\left([A\bar{\nu}^{(t+1)}]/\tau\right)^{\eta\tau} \\ \nu^{(t+1)} \propto [\nu^{(t)}]^{1-\eta\tau} \exp\left(-[A^\top \bar{\mu}^{(t+1)}]/\tau\right)^{\eta\tau} \end{cases}$$

Theorem (Cen, Wei, Chi, 2021)

Suppose that $\eta \leq \min\left\{\frac{1}{2\tau+2\|A\|_\infty}, \frac{1}{4\|A\|_\infty}\right\}$, *then for all* $t \geq 0$, *the last-iterate converges to* ϵ*-QRE within* $\widetilde{O}\left(\frac{1}{\eta\tau}\log\frac{1}{\epsilon}\right)$ *iterations.*

Linear, last-iterate convergence to the QRE!

Soft value iteration via nested-loop OMWU

Soft value iteration: for $h = H, \dots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Soft value iteration via nested-loop OMWU

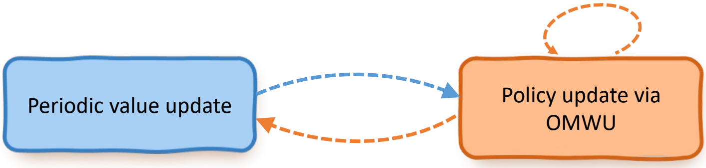
Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Nested-loop approach:

$(\mu_h^{(t)}, \nu_h^{(t)}) \leftarrow \texttt{OMWU}(Q_h)$



$Q_h \leftarrow \texttt{SVI}(Q_{h+1})$

Soft value iteration via nested-loop OMWU

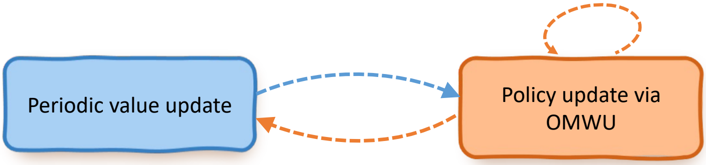
Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Big[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s')\nu(s') + \tau\mathcal{H}(\mu(s')) - \tau\mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Big],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A\times B}$.

Nested-loop approach:

$(\mu_h^{(t)}, \nu_h^{(t)}) \leftarrow \texttt{OMWU}(Q_h)$



$Q_h \leftarrow \texttt{SVI}(Q_{h+1})$

However, not easy to use in online settings...

A two-timescale single-loop approach?

Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Bigg[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^{\top} Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Bigg],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

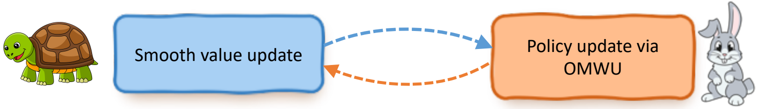
A two-timescale single-loop approach?

Soft value iteration: for $h = H, \ldots, 1$

$$Q_h(s,a,b) \leftarrow r_h(s,a,b) + \cdot \mathop{\mathbb{E}}_{s' \sim P_h(\cdot|s,a,b)} \Big[\underbrace{\max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s'))}_{\text{Entropy-regularized matrix game}} \Big],$$

where $Q_h(s) = [Q_h(s,\cdot,\cdot)] \in \mathbb{R}^{A \times B}$.

Single-loop, two-timescale approach:



$Q^{(t+1)} \leftarrow (1-\alpha) Q^{(t)} + \alpha \cdot \texttt{lookahead}$ $\qquad$ $(\mu^{(t+1)}, \nu^{(t+1)}) \leftarrow \texttt{OMWU}(Q^{(t)})$

Main result: episodic setting

Theorem (Cen, Chi, Du, Xiao, 2022)

The last-iterate of the two-timescale single-loop algorithm finds an ϵ-QRE in

$$\widetilde{O}\left(\frac{H^2}{\tau}\log\frac{1}{\epsilon}\right)$$

iterations, corresponding to $\widetilde{O}\left(\frac{H^3}{\epsilon}\right)$ iterations for finding an ϵ-NE.

- First last-iterate convergence result for the episodic setting.
- **Almost dimension-free:** independent of the size of the state-action space.

Main result: discounted setting

Theorem (Cen, Chi, Du, Xiao, 2022)

For the infinite-horizon γ-discounted setting, the last-iterate of the single-loop algorithm finds an ϵ-QRE in

$$\widetilde{O}\left(\frac{S}{(1-\gamma)^4\tau}\log\frac{1}{\epsilon}\right)$$

iterations, and in $\widetilde{O}\left(\frac{S}{(1-\gamma)^5\epsilon}\right)$ iterations for finding an ϵ-NE.

- This significantly improves upon the prior art $\widetilde{O}\left(\frac{S^5(A+B)^{1/2}}{(1-\gamma)^{16}c^4\epsilon^2}\right)$ of (Wei et al., 2021) and $\widetilde{O}\left(\frac{S^2\|1/\rho\|^5}{(1-\gamma)^{14}c^4\epsilon^3}\right)$ of (Zeng et al., 2022) in *all* parameter dependencies.

Reference I

- "*Simple statistical gradient-following algorithms for connectionist reinforcement learning.*" Williams, Machine Learning, 1992.
- "*Policy gradient methods for reinforcement learning with function approximation.*", Sutton, McAllester, Singh, and Mansour. NeurIPS 1999.
- "*A natural policy gradient.*" Kakade, NeurIPS 2001.
- "*Global convergence of policy gradient methods for the linear quadratic regulator.*" Fazel, Ge, Kakade, and Mesbahi, ICML 2018.
- "*On the theory of policy gradient methods: Optimality, approximation, and distribution shift.*" Agarwal, Kakade, Lee, and Mahajan, Journal of Machine Learning Research, 2021.
- "*On the global convergence rates of softmax policy gradient methods.*" Mei, Xiao, Szepesvári, and Schuurmans, ICML 2020.
- "*Softmax policy gradient methods can take exponential time to converge.*" Li, Wei, Chi, and Chen, Mathematical Programming, 2023.

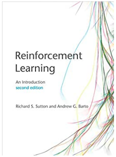
Reference II

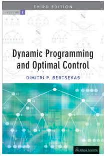
- "*Fast global convergence of natural policy gradient methods with entropy regularization.*" Cen, Cheng, Chen, Wei, and Chi, Operations Research, 2022.
- "*Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence.*" Zhan, Cen, Huang, Chen, Lee, and Chi, SIAM Journal on Optimization, 2023.
- "*Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes.*" Lan, Mathematical Programming, 2021.
- "*Stochastic games,*" L. S. Shapley, *PNAS*, 1953
- "*Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity,*" K. Zhang, S. Kakade, T. Basar, L. Yang, *NeurIPS* 2020
- "*When can we learn general-sum Markov games with a large number of players sample-efficiently?*" Z. Song, S. Mei, Y. Bai, *ICLR* 2022
- "*V-learning: A simple, efficient, decentralized algorithm for multiagent RL,*" C. Jin, Q. Liu, Y. Wang, T. Yu, 2021

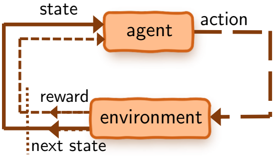
Reference III

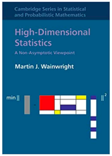
- *"Minimax-optimal multi-agent RL in markov games with a generative model,"* G. Li, Y. Chi, Y. Wei, Y. Chen, *NeurIPS* 2022
- *"The complexity of Markov equilibrium in stochastic games,"* C. Daskalakis, N. Golowich, K. Zhang, *COLT* 2023
- *"Fast policy extragradient methods for competitive games with entropy regularization,"* S. Cen, Y. Wei, Y. Chi, *NeurIPS* 2021.

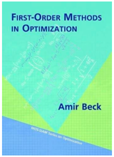
Concluding Remarks











Designing RL algorithms and understanding their non-asymptotic performances are fruitful!

Promising directions:

- function approximation
- multi-agent/federated RL
- safe RL
- many more...

Thanks!

https://users.ece.cmu.edu/~yuejiec/