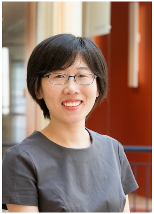
Reinforcement Learning: Fundamentals, Algorithms, and Theory

Yuting Wei
UPenn

Yuxin Chen
UPenn



Yuejie Chi
CMU

ICASSP Tutorial, May 2022

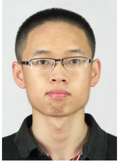
Reinforcement Learning: Fundamentals, Algorithms, and Theory (Part 1)

Yuting Wei

Statistics & Data Science, Wharton
University of Pennsylvania

ICASSP, May 2022

Our wonderful collaborators

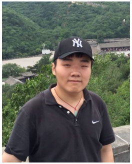


Gen Li
UPenn

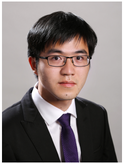
Shicong Cen
CMU

Chen Cheng
Stanford

Laixi Shi
CMU

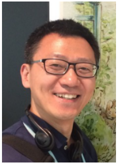


Yuling Yan
Princeton

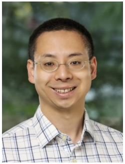


Changxiao Cai
UPenn

Wenhao Zhan
Princeton



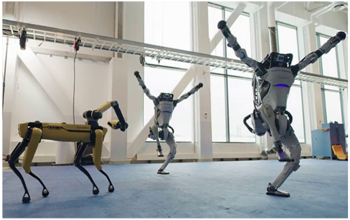
Yuantao Gu
Tsinghua



Jason Lee
Princeton

Jianqing Fan
Princeton

Successes of reinforcement learning (RL)



Recap: Supervised learning

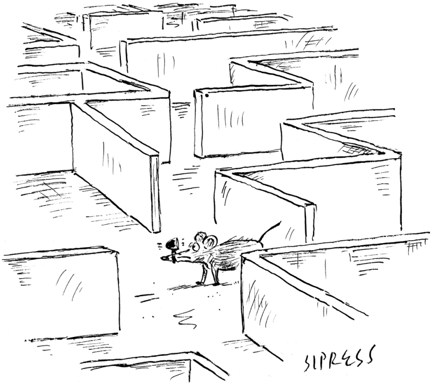
Given i.i.d training data, the goal is to make prediction on unseen data:

— *pic from internet*

Reinforcement learning (RL)

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

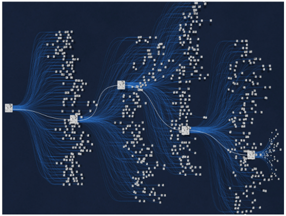
- no training data
- maximize total rewards
- trial-and-error
- sequential and online



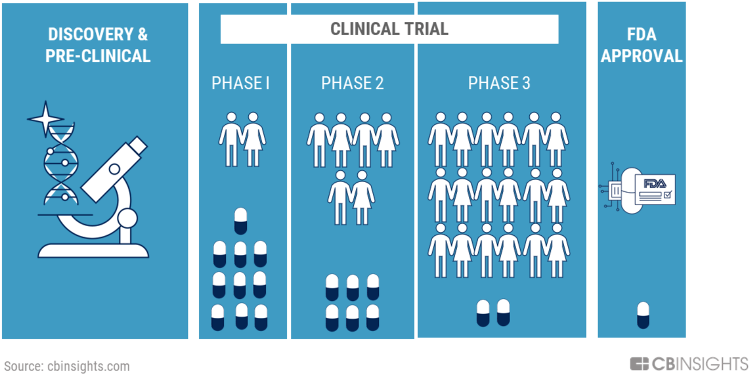
"Recalculating ... recalculating ..."

Challenges of RL

- explore or exploit: unknown or changing environments
- credit assignment problem: delayed rewards or feedback
- enormous state and action space
- nonconvex optimization

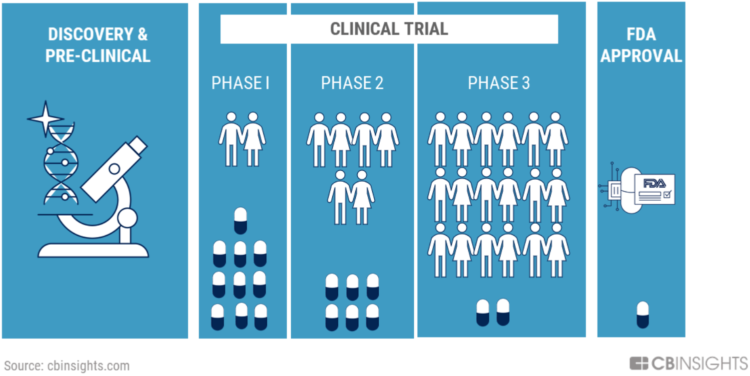


Sample efficiency



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

Sample efficiency



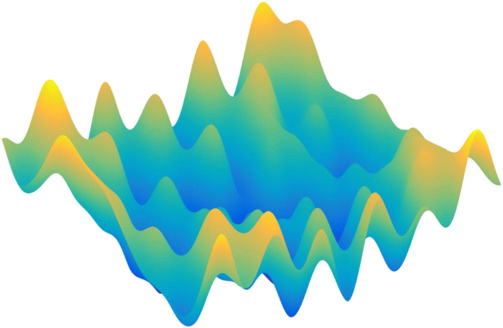
- prohibitively large state & action space
- collecting data samples can be expensive or time-consuming

Challenge: design sample-efficient RL algorithms

Computational efficiency

Running RL algorithms might take a long time . . .

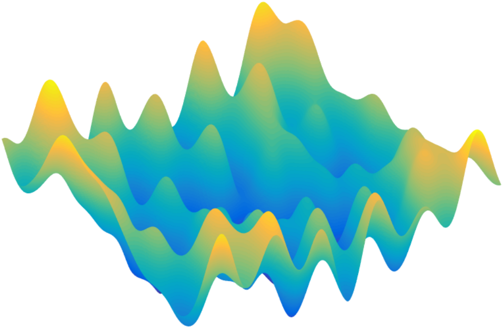
- enormous state-action space
- nonconvexity



Computational efficiency

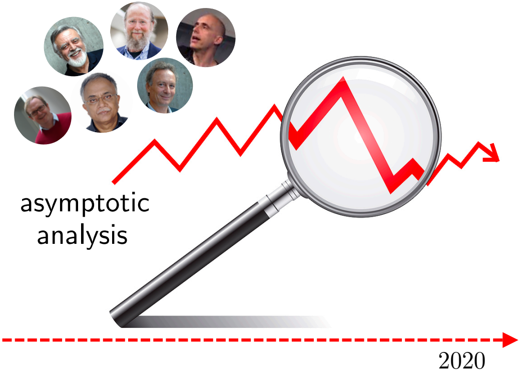
Running RL algorithms might take a long time . . .

- enormous state-action space
- nonconvexity

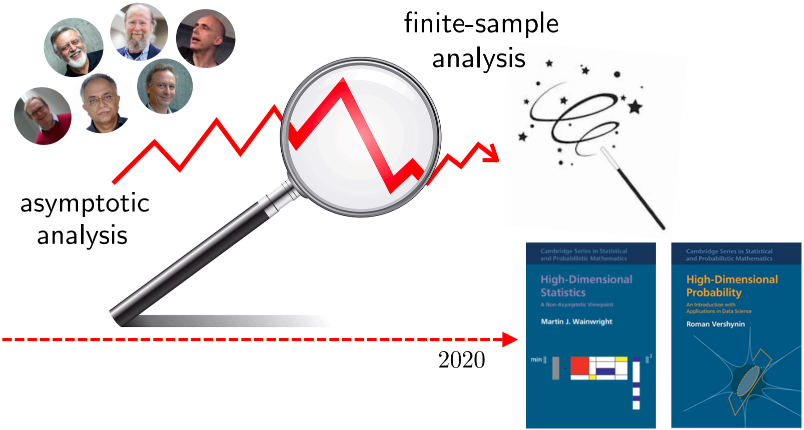


Challenge: design computationally efficient RL algorithms

Theoretical foundation of RL

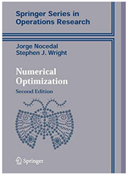


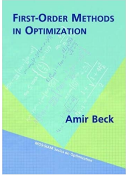
Theoretical foundation of RL

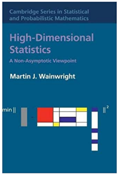


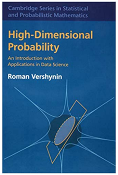
Understanding sample efficiency of RL requires a modern suite of non-asymptotic analysis tools

This tutorial





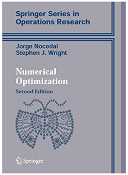


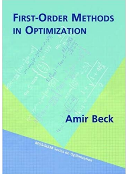
(large-scale) optimization

(high-dimensional) statistics

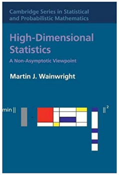
Demystify sample- and computational efficiency of RL algorithms

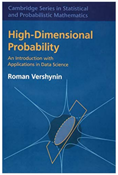
This tutorial





(large-scale) optimization





(high-dimensional) statistics

Demystify sample- and computational efficiency of RL algorithms

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

Part 2. **model-free RL**

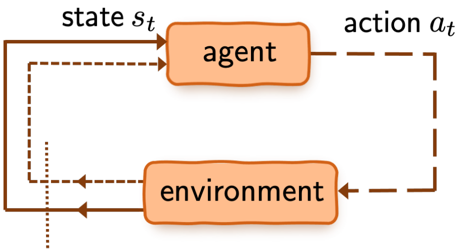
Part 3. **policy optimization**

Outline (Part 1)

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

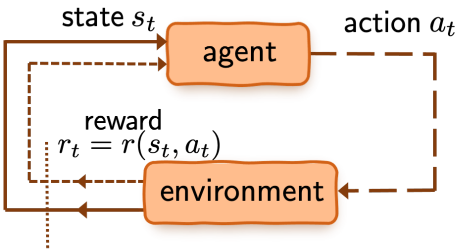
Basics: Markov decision processes

Markov decision process (MDP)



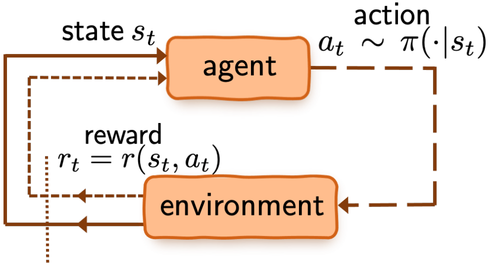
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space

Markov decision process (MDP)



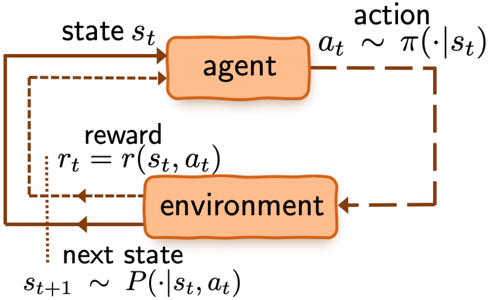
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward

Markov decision process (MDP)



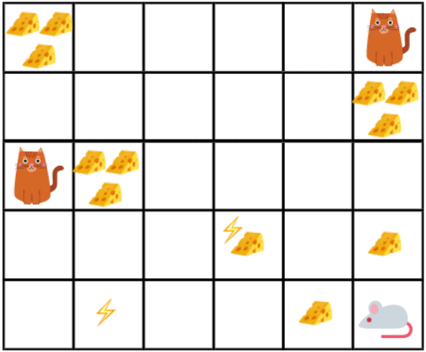
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s,a) \in [0,1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)

Markov decision process (MDP)

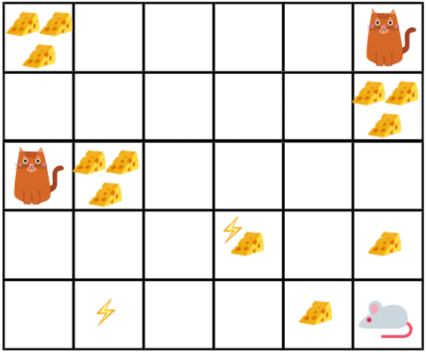


- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s,a) \in [0,1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)
- $P(\cdot|s,a)$: unknown transition probabilities

Help the mouse!

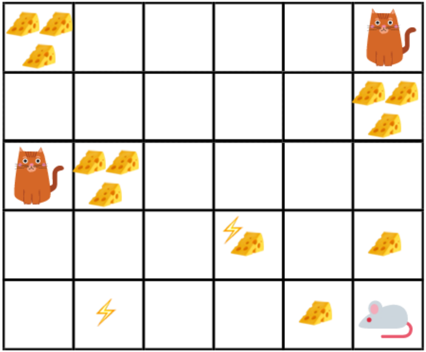


Help the mouse!



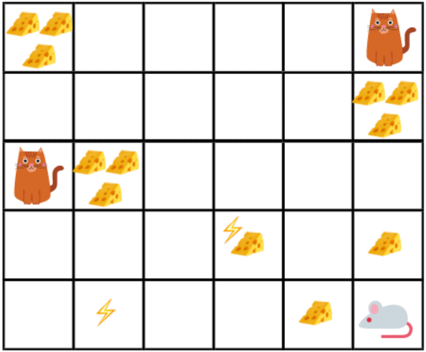
- state space $\mathcal{S}$: positions in the maze

Help the mouse!



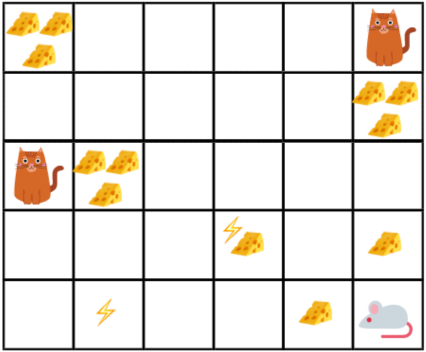
- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right

Help the mouse!



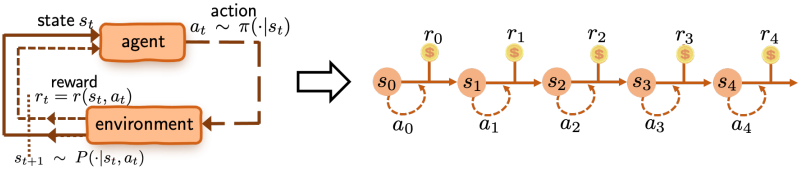
- state space $\mathcal{S}$: positions in the maze
- action space $\mathcal{A}$: up, down, left, right
- immediate reward r: cheese, electricity shocks, cats

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

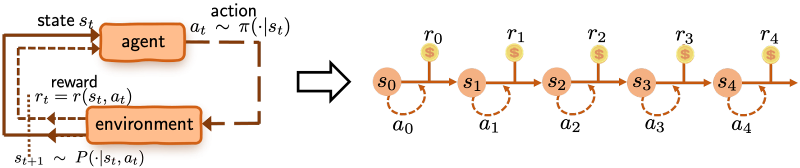
Value function



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

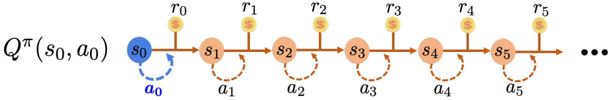


Value of policy π: cumulative discounted reward

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

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

Q-function (action-value function)

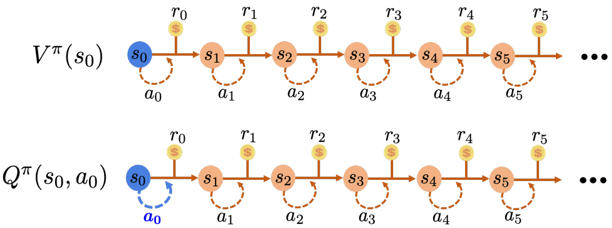


Q-function of policy π:

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

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

Q-function (action-value function)

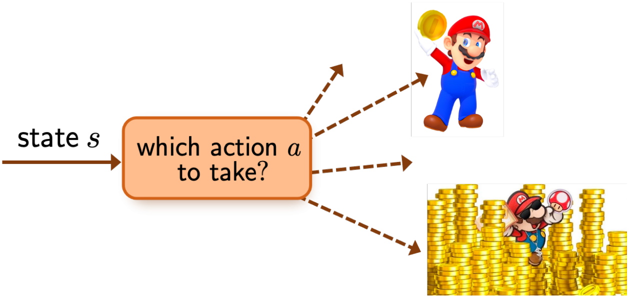


Q-function of policy π:

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

- $(\not{a_0}, s_1, a_1, s_2, a_2, \cdots)$: induced by policy π

Optimal policy and optimal value



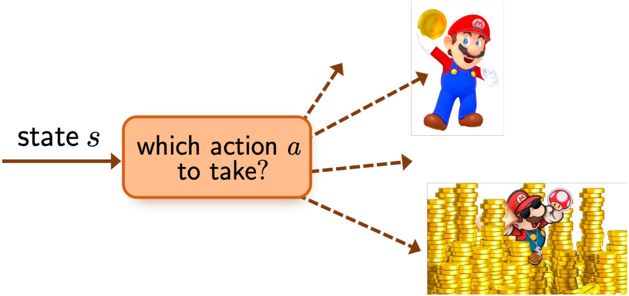
optimal policy $\pi^\star$: maximizing value function $\max_\pi V^\pi$

Proposition (Puterman'94)

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

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

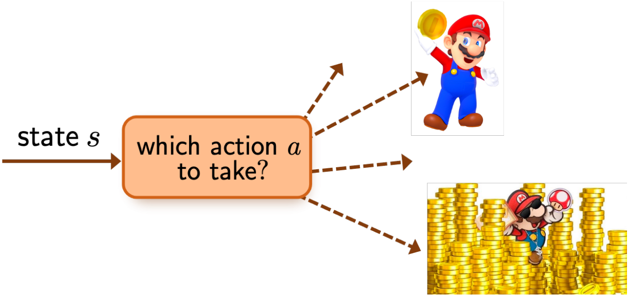
Optimal policy and optimal value



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 $V^{\pi}, \ \forall s$?)

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 $V^\pi, \ \forall s$?)

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

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

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

Richard Bellman

Policy evaluation: Bellman's consistency equation

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

Bellman's consistency equation

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

- 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

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

- one-step look-ahead

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

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

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

γ**-contraction of Bellman operator:**

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

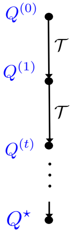
Richard Bellman

Two dynamic programming algorithms

Value iteration (VI)

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

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

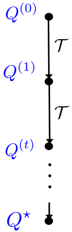


Two dynamic programming algorithms

Value iteration (VI)

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

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

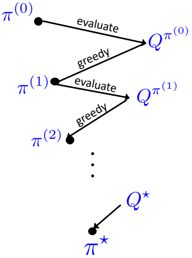


Policy iteration (PI)

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

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

policy improvement: $\pi^{(t+1)}(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^{(t)}(s, a)$



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)

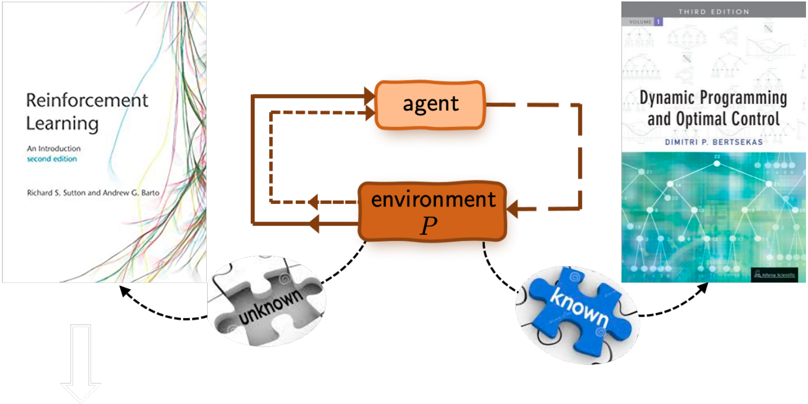
$$\big\|Q^{(t)} - Q^{\star}\big\|_{\infty} \leq \gamma^{t}\big\|Q^{(0)} - Q^{\star}\big\|_{\infty}$$

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

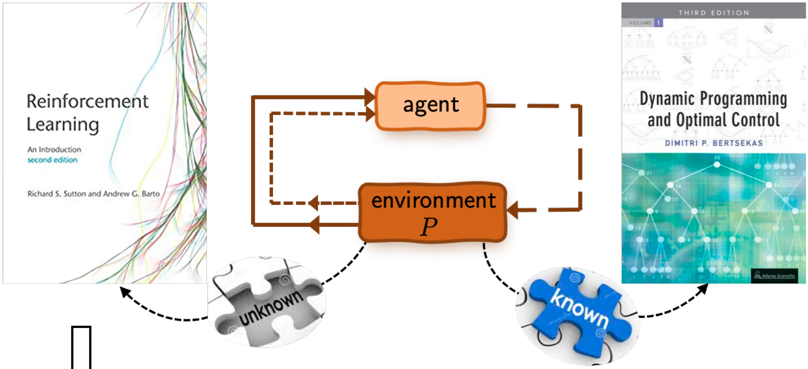
$$\frac{1}{1-\gamma}\log\left(\frac{\|Q^{(0)} - Q^{\star}\|_{\infty}}{\varepsilon}\right) \quad \text{iterations}$$

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

When the model is unknown . . .



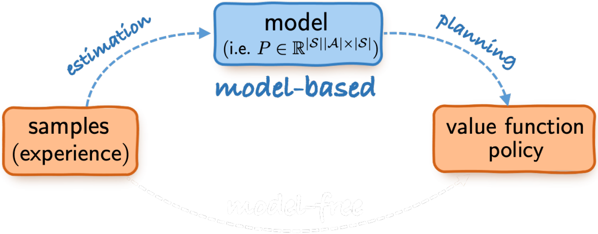
When the model is unknown ...



⇓

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

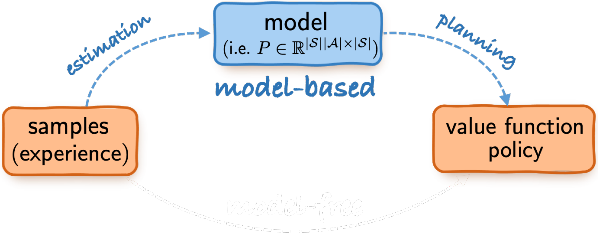
Three approaches



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 $\widehat{P}$

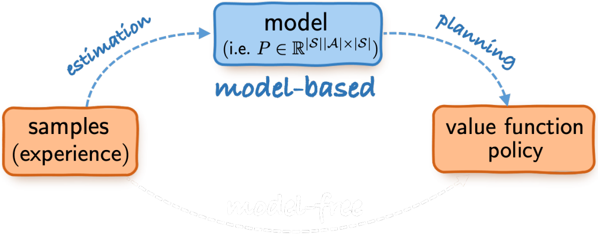
Tutorial Part 2: Model-free approach

— learning w/o estimating the model explicitly

Tutorial Part 3: 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 $\widehat{P}$

Tutorial Part 2: Model-free approach

— learning w/o estimating the model explicitly

Tutorial Part 3: Policy based approach

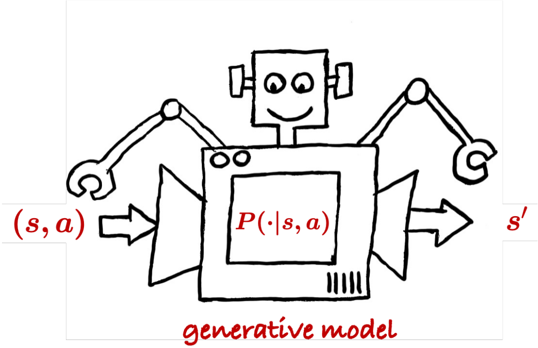
— optimization in the space of policies

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

1. Sampling from a generative model (simulator)
2. Offline RL / batch RL

A generative model / simulator

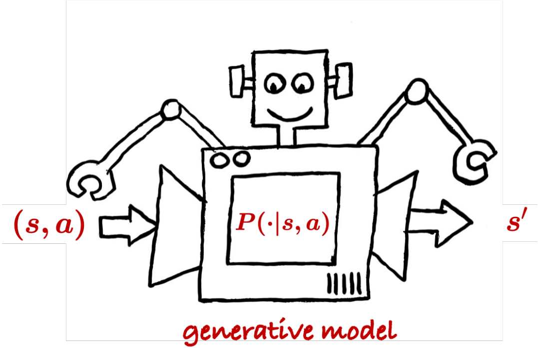
— [Kearns and Singh, 1999]



- **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*]



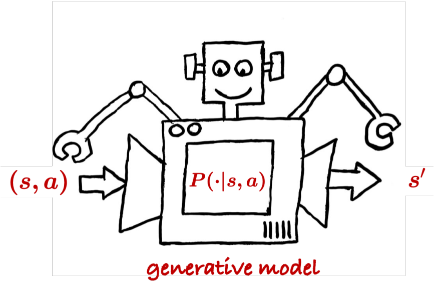
- **sampling:** for each (s, a), collect N samples $\{(s, a, s'_{(i)})\}_{1 \leq i \leq N}$
- construct $\widehat{\pi}$ based on samples (in total $|\mathcal{S}||\mathcal{A}| \times N$)

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

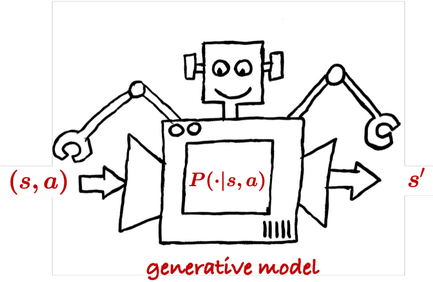
- [Kearns and Singh, 1999]
- [Kakade, 2003]
- [Kearns et al., 2002]
- [Azar et al., 2012]
- **[Azar et al., 2013]**
- [Sidford et al., 2018a]
- [Sidford et al., 2018b]
- [Wang, 2019]
- **[Agarwal et al., 2019]**
- [Wainwright, 2019a]
- [Wainwright, 2019b]
- [Pananjady and Wainwright, 2019]
- [Yang and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

Model estimation



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

Model estimation



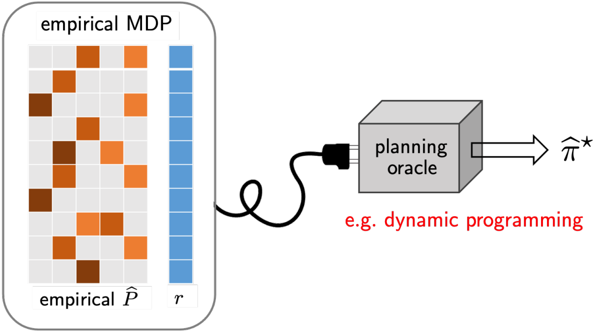
Sampling: for each (s,a), collect N ind. samples $\{(s,a,s'_{(i)})\}_{1\le i\le N}$

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

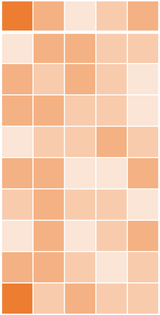
— [Azar et al., 2013, Agarwal et al., 2019]



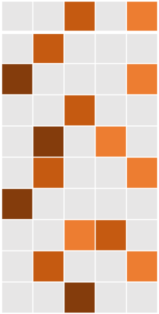
Find policy based on the empirical MDP (*empirical maximizer*)

using, e.g., policy iteration — $(\widehat{P}, r)$

Challenges in the sample-starved regime



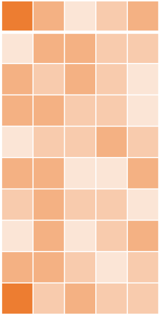
truth: $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$



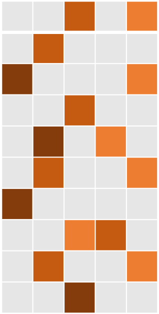
empirical estimate: $\widehat{P}$

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

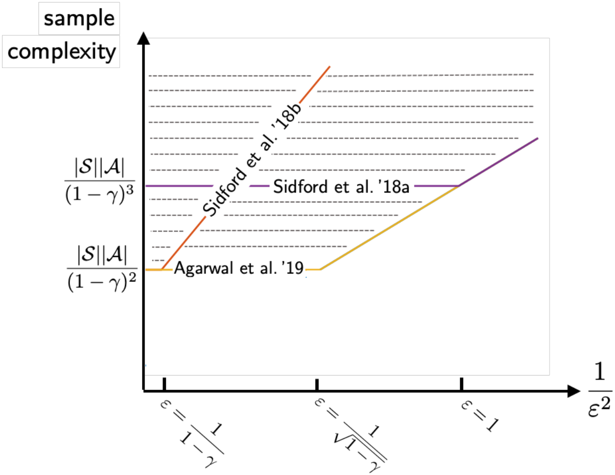
For any $0 < \varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$*, the optimal policy* $\widehat{\pi}^\star$ *of empirical MDP achieves*

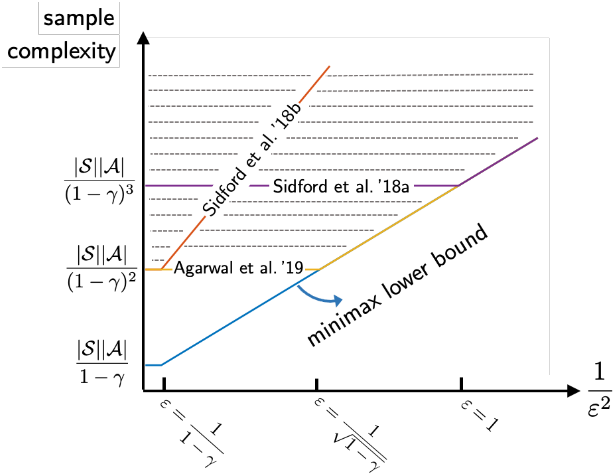
$$\|V^{\widehat{\pi}^\star} - V^\star\|_\infty \leq \varepsilon$$

with high prob., with sample complexity at most

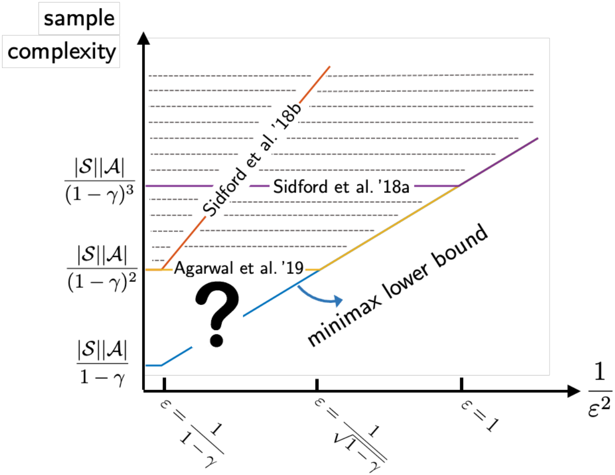
$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$$

- 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

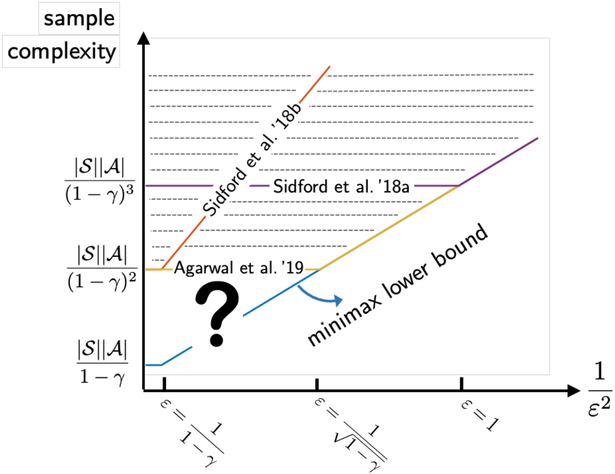

sample
complexity
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3}$
$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$
Sidford et al. '18b
Sidford et al. '18a
Agarwal et al. '19
$\frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = \frac{1}{\sqrt{1-\gamma}}$
$\varepsilon = 1$



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 $\gtrsim \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$

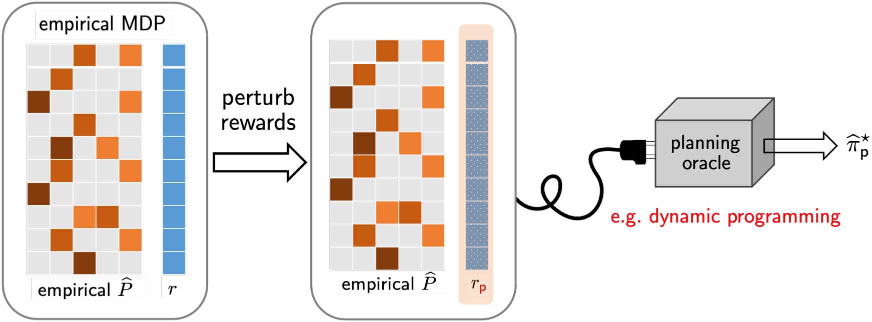


[Agarwal et al., 2019] still requires a burn-in sample size $\gtrsim \frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^2}$

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, Gu, 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, Gu, Chen '20)

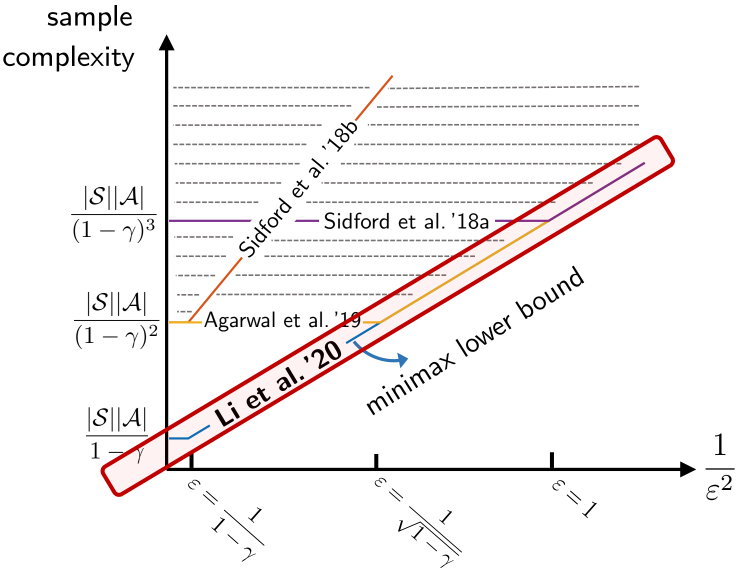
For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *the optimal policy* $\widehat{\pi}_{\rm p}^\star$ *of perturbed empirical MDP achieves*

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

with high prob., with sample complexity at most

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

- 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
$\frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = \frac{1}{\sqrt{1-\gamma}}$
$\varepsilon = 1$

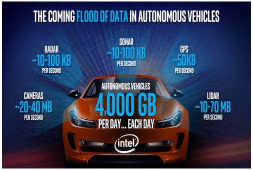
Model-based RL (a "plug-in" approach)

1. Sampling from a generative model (simulator)
2. Offline RL / batch RL

Offline RL / Batch RL

- Collecting new data might be expensive or time-consuming
- But we have already stored tons of historical data

medical records



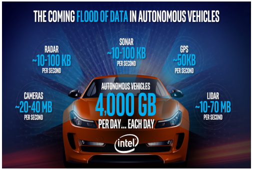
data of self-driving

clicking times of ads

Offline RL / Batch RL

- Collecting new data might be expensive 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 design algorithms based solely on historical data?

Offline RL / batch RL

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

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

for some state distribution ρ^{b} and behavior policy π^{b}

Offline RL / batch RL

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

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

for some state distribution ρ^{b} and behavior policy π^{b}

Goal: given some test distribution ρ and accuracy level ε, find an ε-optimal policy $\widehat{\pi}$ based on $\mathcal{D}$ obeying

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

— *in a sample-efficient manner*

Challenges of offline RL

- **Distribution shift**:

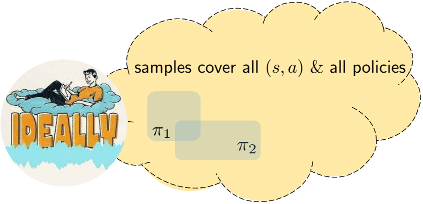
$$\text{distribution}(\mathcal{D}) \; \neq \; \text{target distribution under } \pi^\star$$

Challenges of offline RL

- **Distribution shift**:

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

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



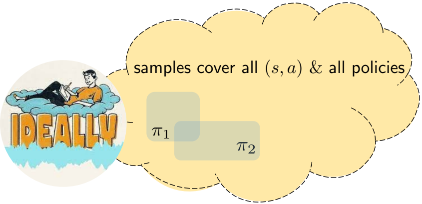
uniform coverage over entire space
(sufficiently explored)

Challenges of offline RL

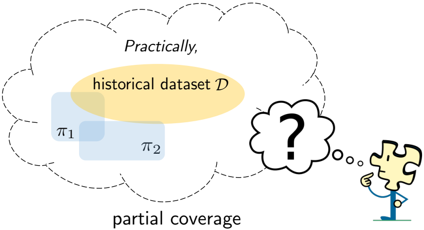
- **Distribution shift**:

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

- **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

$$C^{\star} := \max_{s,a} \frac{d^{\pi^{\star}}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)}$$

where $d^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}\big((s^t,a^t)=(s,a)\,|\,\pi\big)$

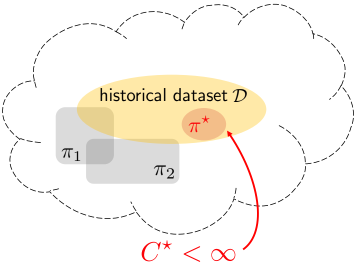
How to quantify quality of historical dataset $\mathcal{D}$ (induced by π^{b})?

Single-policy concentrability coefficient

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

where $d^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}\big((s^t,a^t) = (s,a) \,|\, \pi\big)$

- captures distributional shift
- allows for partial coverage



A model-based offline algorithm: VI-LCB

Pessimism in the face of uncertainty: penalize value estimate of those (s, a) pairs that were poorly visited [Jin et al., 2021, Rashidinejad et al., 2021]

A model-based offline algorithm: VI-LCB

Pessimism in the face of uncertainty: penalize value estimate of those (s,a) pairs that were poorly visited [Jin et al., 2021, Rashidinejad et al., 2021]

Algorithm: value iteration w/ lower confidence bounds

- compute empirical estimate $\widehat{P}$ of P
- initialize $\widehat{Q} = 0$, and repeat

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

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

Minimax optimality of model-based offline RL

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

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

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

with high prob., with sample complexity at most

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

Minimax optimality of model-based offline RL

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

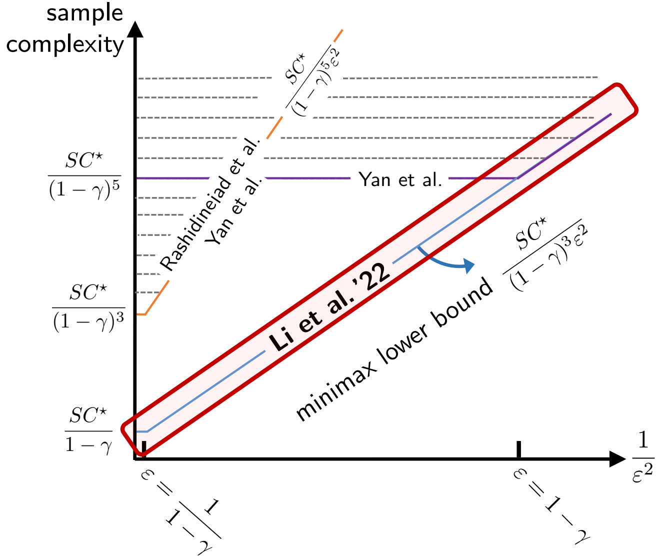
For any $0 < \varepsilon \le \frac{1}{1-\gamma}$, the policy $\widehat{\pi}$ returned by VI-LCB achieves

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

with high prob., with sample complexity at most

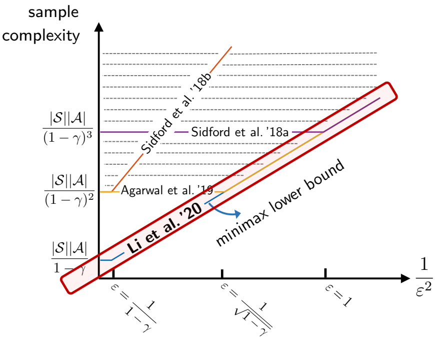
$$\widetilde{O}\left(\frac{SC^{\star}}{(1-\gamma)^3\varepsilon^2}\right)$$

- matches minimax lower bound: $\widetilde{\Omega}\left(\frac{SC^{\star}}{(1-\gamma)^3\varepsilon^2}\right)$ [Rashidinejad et al., 2021]
- depends on distribution shift (as reflected by $C^{\star}$)
- full ε-range (no burn-in cost)

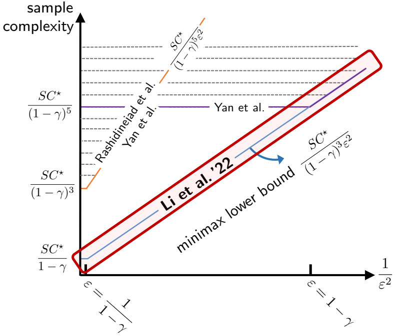


sample complexity
$\frac{SC^\star}{(1-\gamma)^5}$
$\frac{SC^\star}{(1-\gamma)^3}$
$\frac{SC^\star}{1-\gamma}$
$\frac{SC^\star}{(1-\gamma)^5\varepsilon^2}$
Rashidinejad et al.
Yan et al.
Yan et al.
Li et al. '22
$\frac{SC^\star}{(1-\gamma)^3\varepsilon^2}$
minimax lower bound
$\frac{1}{\varepsilon^2}$
$\varepsilon = \frac{1}{1-\gamma}$
$\varepsilon = 1-\gamma$

Summary of this part



generative model



offline/batch RL

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

Reference I

- "*Reinforcement Learning: Theory and Algorithms*," A. Agarwal, N. Jiang, S. Kakade, W. Sun, in preparation.
- "*Dynamic programming and optimal control (4th edition)*," D. Bertsekas, 2017.
- "*Finite-sample convergence rates for Q-learning and indirect algorithms*," M. Kearns, S. Singh *NeurIPS*, 1998.
- "*Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model*," M. Azar, R. Munos, H. J. Kappen, *Machine Learning*, vol. 91, no. 3, 2013.
- "*Near-optimal time and sample complexities for solving Markov decision processes with a generative model*," A. Sidford, M. Wang, X. Wu, L. Yang, Y. Ye, *NeurIPS*, 2018.
- "*Model-based reinforcement learning with a generative model is minimax optimal*," A. Agarwal, S. Kakade, L. F. Yang, *COLT*, 2020.

Reference II

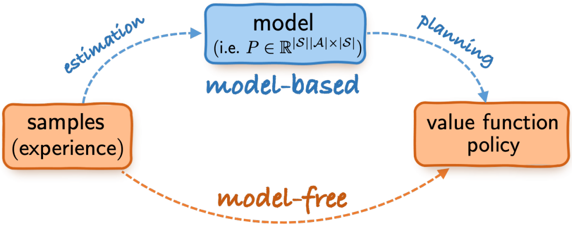
- "*Breaking the sample size barrier in model-based reinforcement learning with a generative model*," G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, *NeurIPS*, 2020.
- "*Offline reinforcement learning: Tutorial, review, and perspectives on open problems*," S. Levine, A. Kumar, G. Tucker, J. Fu, arXiv:2005.01643, 2020.
- "*Is pessimism provably efficient for offline RL?*" Y. Jin, Z. Yang, Z. Wang, ICML, 2021
- "*Bridging offline reinforcement learning and imitation learning: A tale of pessimism*," P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, S. Russell, *NeurIPS*, 2021.
- "*Policy finetuning: Bridging sample-efficient offline and online reinforcement learning*," T. Xie, N. Jiang, H. Wang, C. Xiong, Y. Bai, *NeurIPS*, 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.

Reinforcement Learning: Fundamentals, Algorithms, and Theory (Part 2)

Yuxin Chen

Wharton Statistics & Data Science, ICASSP 2022

Model-based vs. model-free RL



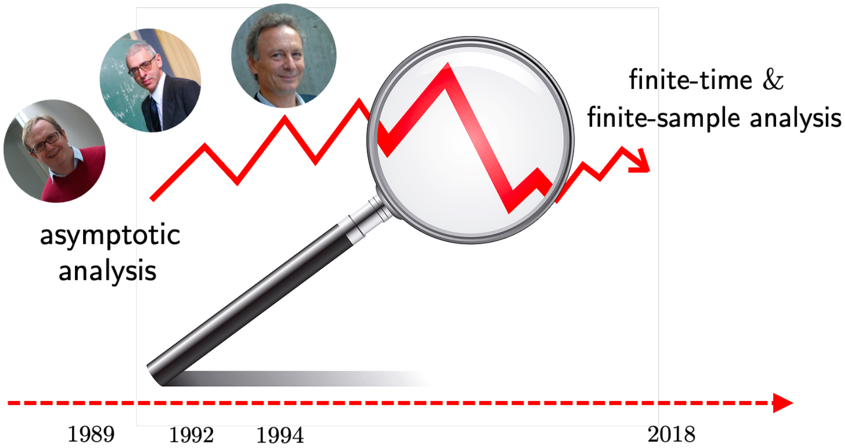
Model-based approach ("plug-in")

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

Model-free approach

— learning w/o modeling & estimating environment explicitly

— memory-efficient, online, . . .



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

Model-free RL

1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)
4. Q-learning with lower confidence bounds (offline RL)
5. Q-learning with upper confidence bounds (online RL)

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

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

- one-step look-ahead

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

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

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

Richard Bellman

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

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

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

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

Q-learning: a stochastic approximation algorithm

Chris Watkins

Peter Dayan

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

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

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

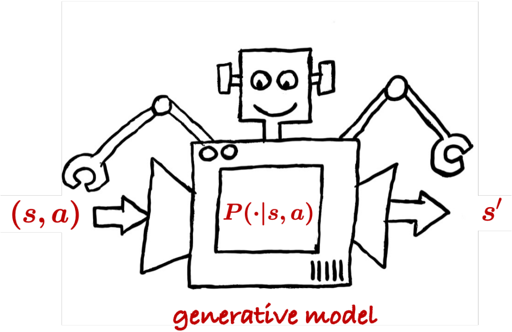
$$\mathcal{T}(Q)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{a'} Q(s',a') \right]$$

Model-free RL

1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)
4. Q-learning with lower confidence bounds (offline RL)
5. Q-learning with upper confidence bounds (online RL)

A generative model / simulator

— *Kearns, Singh '99*



In each iteration, collect an independent sample (s, a, s') for each (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

Sample complexity of synchronous Q-learning

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

For any $0 < \varepsilon \leq 1$, *synchronous Q-learning yields* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *with high prob., with sample complexity (i.e.,* $T|\mathcal{S}||\mathcal{A}|$*) at most*

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

other papers	sample complexity
Even-Dar & Mansour '03	$2^{\frac{1}{1-\gamma}} \frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^4\varepsilon^2}$
Beck & Srikant '12	$\frac{\|\mathcal{S}\|^2\|\mathcal{A}\|^2}{(1-\gamma)^5\varepsilon^2}$
Wainwright '19	$\frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^5\varepsilon^2}$
Chen et al. '20	$\frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^5\varepsilon^2}$

Sample complexity of synchronous Q-learning

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

For any $0 < \varepsilon \leq 1$, *synchronous Q-learning yields* $\|\widehat{Q} - Q^{\star}\|_{\infty} \leq \varepsilon$ *with high prob., with sample complexity (i.e.,* $T|\mathcal{S}||\mathcal{A}|$*) at most*

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

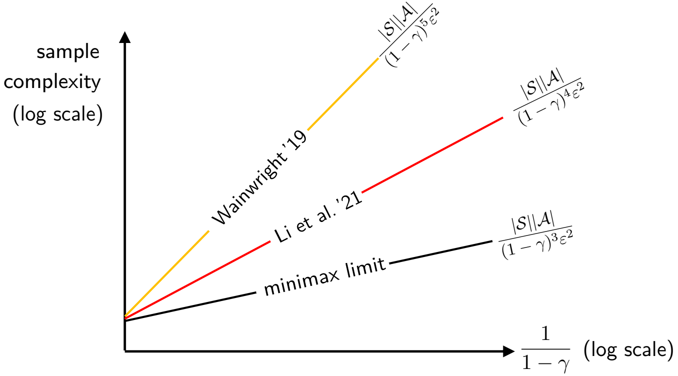
- Covers both *constant* and *rescaled linear* learning rates:

$$\eta_t \equiv \frac{1}{1 + \frac{c_1(1-\gamma)T}{\log^2 T}}$$

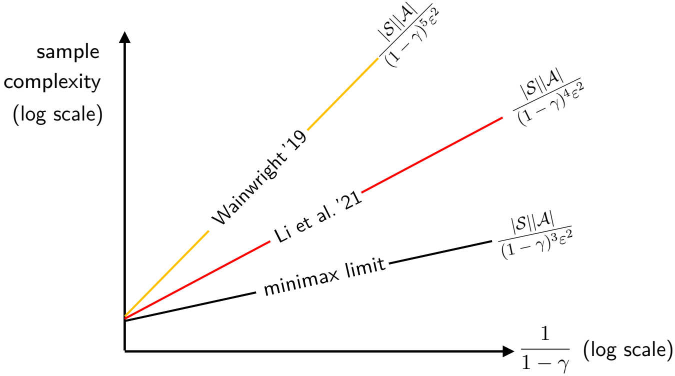
or $$\eta_t = \frac{1}{1 + \frac{c_2(1-\gamma)t}{\log^2 T}}$$

other papers	sample complexity
Even-Dar & Mansour '03	$2^{\frac{1}{1-\gamma}} \frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^4\varepsilon^2}$
Beck & Srikant '12	$\frac{\|\mathcal{S}\|^2\|\mathcal{A}\|^2}{(1-\gamma)^5\varepsilon^2}$
Wainwright '19	$\frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^5\varepsilon^2}$
Chen et al. '20	$\frac{\|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^5\varepsilon^2}$

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



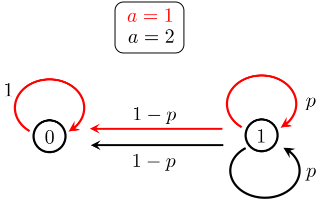
All this requires sample size at least $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ ...



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

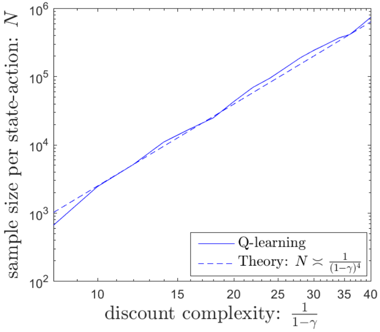
A numerical example: $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ samples seem necessary . . .

— *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 2 (Li, Cai, Chen, Gu, Wei, Chi, 2021)

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

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

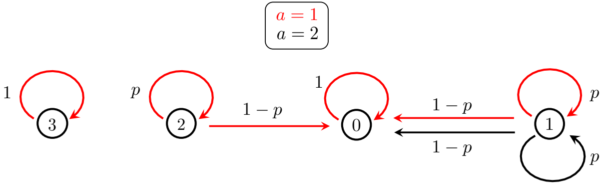
Q-learning is NOT minimax optimal

Theorem 2 (Li, Cai, Chen, Gu, Wei, Chi, 2021)

For any $0 < \varepsilon \leq 1$, *there exist an MDP 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

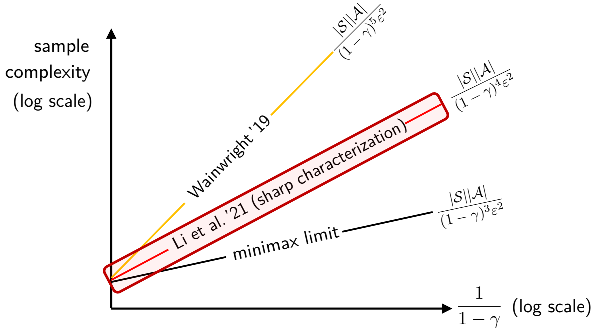


Q-learning is NOT minimax optimal

Theorem 2 (Li, Cai, Chen, Gu, Wei, Chi, 2021)

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

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



Why is Q-learning sub-optimal?

Over-estimation of Q-functions (Thrun & Schwartz '93; Hasselt '10)

- $\max_{a \in \mathcal{A}} \mathbb{E}[X(a)]$ tends to be over-estimated (high positive bias) when $\mathbb{E}[X(a)]$ is replaced by its empirical estimates using a small sample size
- often gets worse with a large number of actions (Hasselt, Guez, Silver '15)

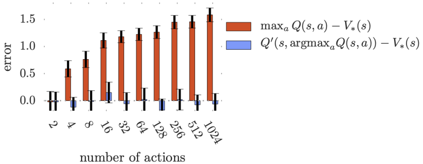


Figure 1: The orange bars show the bias in a single Q-learning update when the action values are $Q(s,a) = V_*(s) + \epsilon_a$ and the errors $\{\epsilon_a\}_{a=1}^m$ are independent standard normal random variables. The second set of action values Q', used for the blue bars, was generated identically and independently. All bars are the average of 100 repetitions.

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)*

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

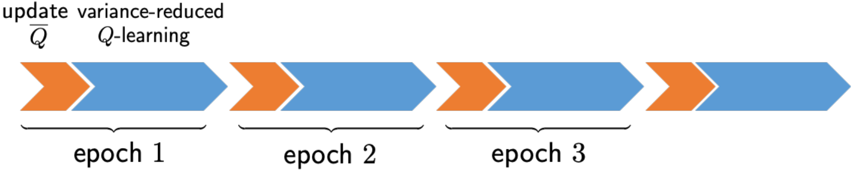
- $\overline{Q}$: some reference Q-estimate
- $\widetilde{\mathcal{T}}$: empirical Bellman operator (using a *batch* of samples)

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

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

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 3 (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*

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

- allows for more aggressive learning rates

Sample complexity of variance-reduced Q-learning

Theorem 3 (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*

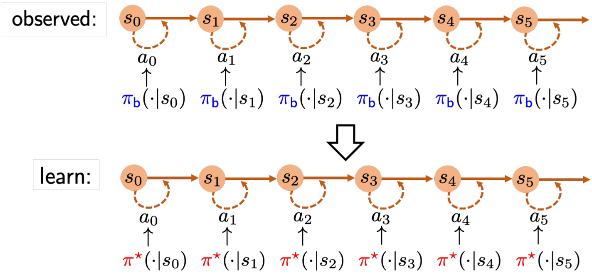
$$\widetilde{O}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^{3}\varepsilon^{2}}\right)$$

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

Model-free RL

1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)
4. Q-learning with lower confidence bounds (offline RL)
5. Q-learning with upper confidence bounds (online RL)

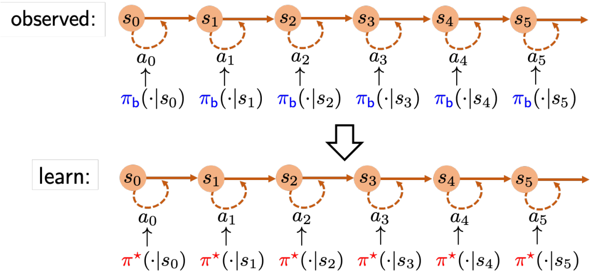
Markovian samples and behavior policy



Observed: $\underbrace{\{s_t, a_t, r_t\}_{t\geq 0}}_{\text{stationary Markovian trajectory}}$ generated 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 (uniform coverage)

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

- mixing time: t_{mix}

Q-learning on Markovian samples

Chris Watkins

Peter Dayan

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

Q-learning on Markovian samples

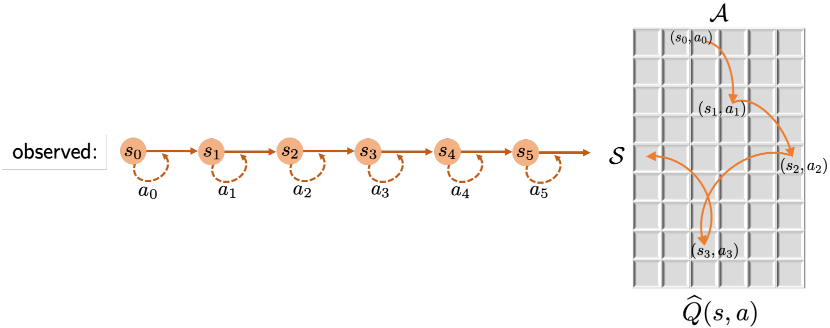
Chris Watkins

Peter Dayan

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

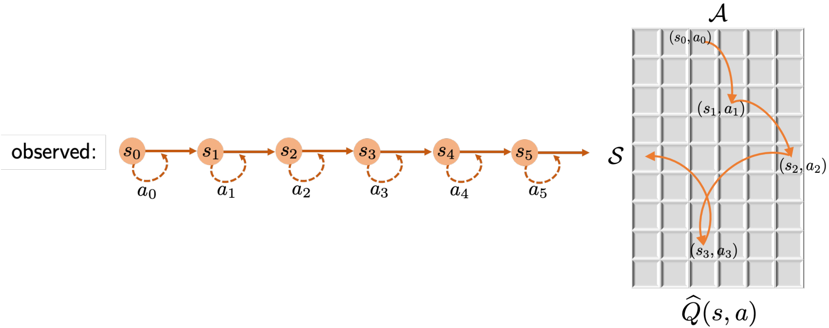
$$\mathcal{T}_t(Q)(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

Q-learning on Markovian samples



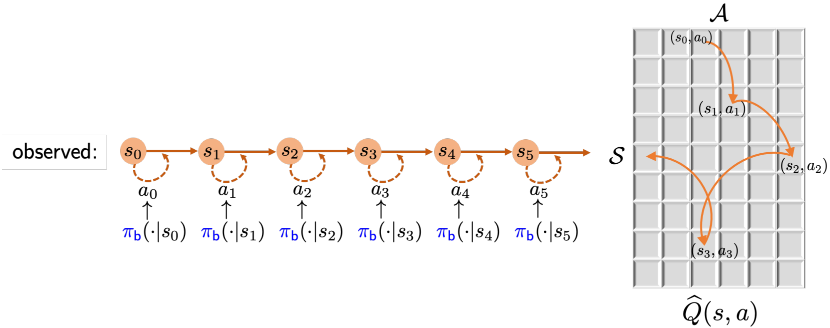
- **asynchronous:** only a single entry is updated each iteration

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

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, Gu, Wei, Chi '21
- Chen, Maguluri, Shakkottai, Shanmugam '21
- ...

Sample complexity of asynchronous Q-learning

Theorem 4 (Li, Cai, Chen, Gu, Wei, Chi '21)

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 log factor)*

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

Sample complexity of asynchronous Q-learning

Theorem 4 (Li, Cai, Chen, Gu, Wei, Chi '21)

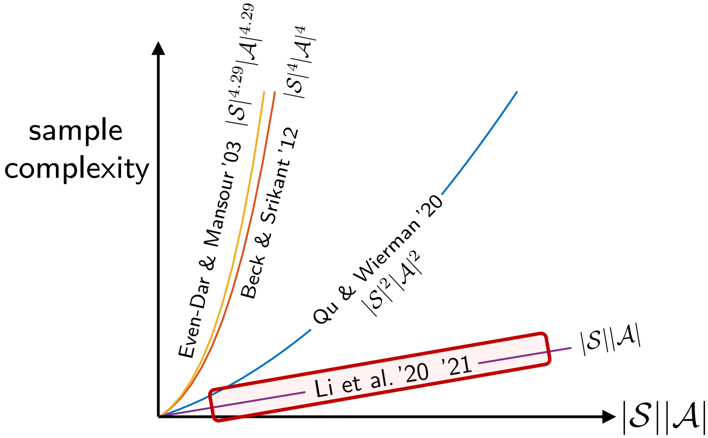
For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, *sample complexity of async Q-learning to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *is at most (up to log factor)*

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

- **learning rates:** constant & rescaled linear

other papers	sample complexity
Even-Dar et al. '03	$\frac{(t_{\mathsf{cover}})^{\frac{1}{1-\gamma}}}{(1-\gamma)^4\varepsilon^2}$
Even-Dar et al. '03	$\left(\frac{t_{\mathsf{cover}}^{1+3\omega}}{(1-\gamma)^4\varepsilon^2}\right)^{\frac{1}{\omega}} + \left(\frac{t_{\mathsf{cover}}}{1-\gamma}\right)^{\frac{1}{1-\omega}}$, $\omega \in (\frac{1}{2}, 1)$
Beck & Srikant '12	$\frac{t_{\mathsf{cover}}^3 \|\mathcal{S}\|\|\mathcal{A}\|}{(1-\gamma)^5\varepsilon^2}$
Qu & Wierman '20	$\frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}^2(1-\gamma)^5\varepsilon^2}$
Li et al. '20	$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^5\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)}$
Chen et al. '21	$\frac{1}{\mu_{\mathsf{min}}^3(1-\gamma)^5\varepsilon^2}$ + other-term(t_{mix})

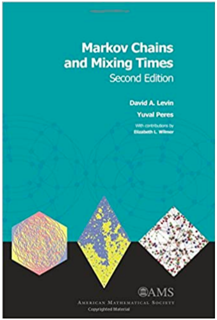
Linear dependency on $1/\mu_{\text{min}}$



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

Effect of mixing time on sample complexity

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



- reflects cost taken to reach steady state
- one-time expense (almost independent of ε)

  — it becomes amortized as algorithm runs
- can be improved with the aid of variance reduction (Li et al. '20)

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

Model-free RL

1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)
4. Q-learning with lower confidence bounds (offline RL)
5. Q-learning with upper confidence bounds (online RL)

Recap: offline RL / batch RL

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

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

for some state distribution ρ^{b} and behavior policy π^{b}

Recap: offline RL / batch RL

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

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

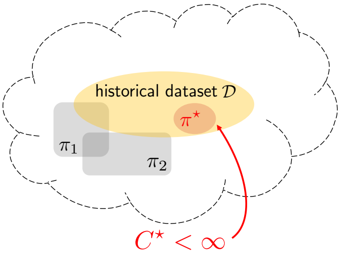
for some state distribution ρ^{b} and behavior policy π^{b}

Single-policy concentrability

$$C^\star := \max_{s,a} \frac{d^{\pi^\star}(s,a)}{d^{\pi^{\mathsf{b}}}(s,a)} \geq 1$$

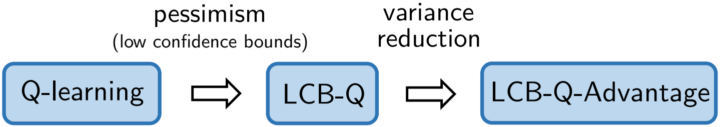
where d^π: occupancy distribution under π

- captures distributional shift
- allows for partial coverage



How to design offline model-free algorithms with optimal sample efficiency?

How 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: $\widetilde{O}\left(\frac{SC^\star}{(1-\gamma)^5\varepsilon^2}\right) \quad \Longrightarrow \quad$ 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*

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\eta_t) Q_t(s_t, a_t) - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$

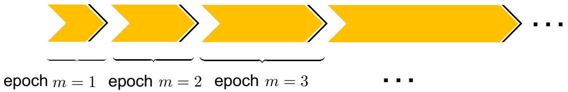
$$+ \eta_t \Big(\underbrace{\mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q})}_{\text{reference}} \Big)(s_t, a_t)$$

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



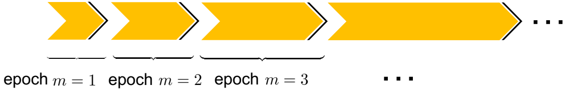
Q-learning with LCB and variance reduction

— *Shi et al. '22, Yan et al. '22*

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\eta_t) Q_t(s_t, a_t) - \eta_t \underbrace{b_t(s_t, a_t)}_{\text{LCB penalty}}$$

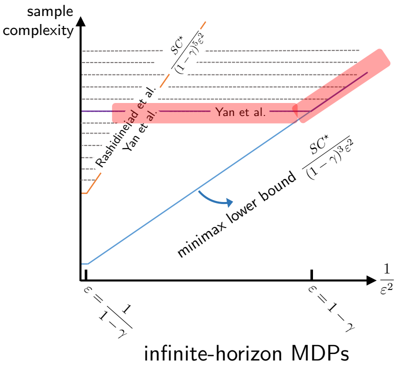
$$+ \eta_t \Big(\underbrace{\mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q})}_{\text{advantage}} + \underbrace{\widehat{\mathcal{T}}(\overline{Q})}_{\text{reference}} \Big)(s_t, a_t)$$

- incorporates variance reduction into LCB-Q



Theorem 5 (Yan, Li, Chen, Fan '22, Shi, Li, Wei, Chen, Chi '22)

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



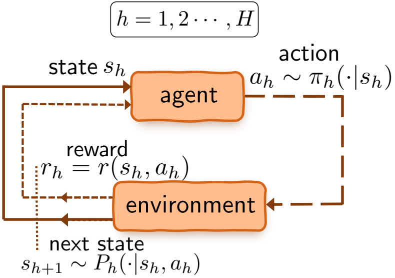
Model-free offline RL attains sample optimality too!

— *with some burn-in cost though . . .*

Model-free RL

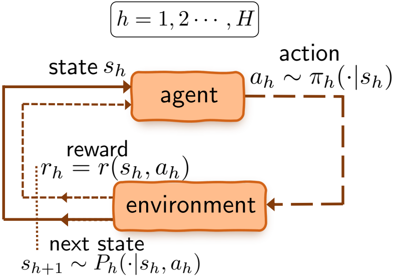
1. Basics of Q-learning
2. Synchronous Q-learning and variance reduction (simulator)
3. Asynchronous Q-learning (Markovian data)
4. Q-learning with lower confidence bounds (offline RL)
5. Q-learning with upper confidence bounds (online RL)

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



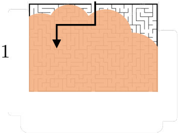
value function: $V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^{H} r_h(s_h, a_h) \,\middle|\, s_h = s\right]$

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

Online RL: interacting with real environments

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

episode 1



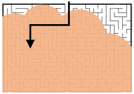
execute π^1

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

Online RL: interacting with real environments

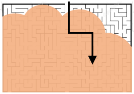
Sequentially execute MDP for K episodes, each consisting of H steps

episode 1



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

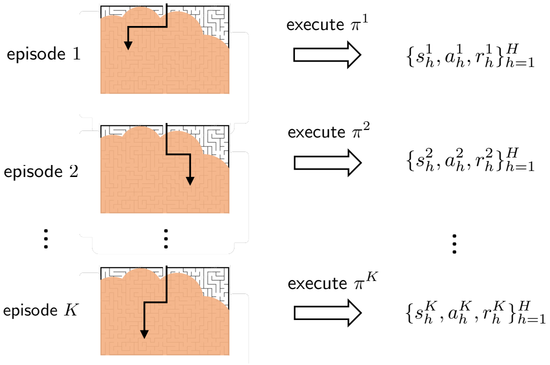
episode 2



execute π^2 $\Longrightarrow$ $\{s_h^2, a_h^2, r_h^2\}_{h=1}^H$

Online RL: interacting with real environments

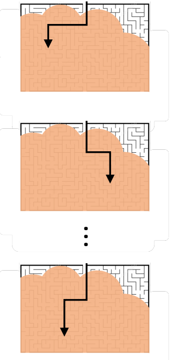
Sequentially execute MDP for K episodes, each consisting of H steps



Online RL: interacting with real environments

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

— *sample size:* $T = KH$



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

episode 2 — execute π^2 $\Longrightarrow$ $\{s_h^2, a_h^2, r_h^2\}_{h=1}^H$

$\vdots$

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

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

Regret: gap between learned policy & optimal policy

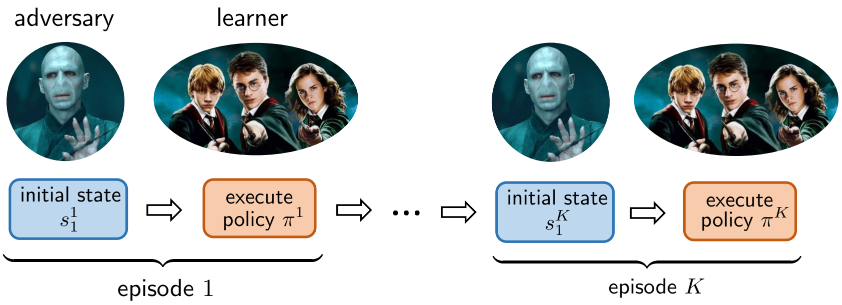
adversary

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

adversary learner

initial state s_1^1 ⇨ execute policy π^1 ⇨ ··· ⇨ initial state s_1^K ⇨ execute policy π^K

episode 1 … episode K

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

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

Lower bound

(Domingues et al. '21)

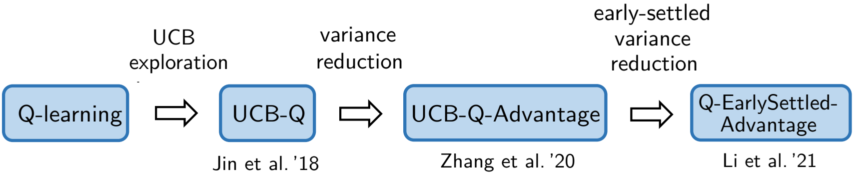
$$\text{Regret}(T) \gtrsim \sqrt{H^2SAT}$$

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**
- UCB-M-Q: Menard et al. '21
- **Q-EarlySettled-Advantage: Li et al. '21**

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)

$\text{Regret}(T) \lesssim \sqrt{H^3SAT}$ $\Longrightarrow$ 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; 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*

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 variance reduction into UCB-Q:

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

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates variance reduction into UCB-Q:

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

UCB-Q-Advantage is asymptotically regret-optimal

Q-learning with UCB and variance reduction

— *Zhang et al. '20*

Incorporates variance reduction into UCB-Q:

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

UCB-Q-Advantage is asymptotically regret-optimal

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

UCB-Q with variance reduction and early settlement

One additional key idea: early settlement of the reference as soon as it reaches a reasonable quality

UCB-Q with variance reduction and early settlement

One additional key idea: early settlement of the reference as soon as it reaches a reasonable quality

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

With high prob., Q-EarlySettled-Advantage *achieves*

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

UCB-Q with variance reduction and early settlement

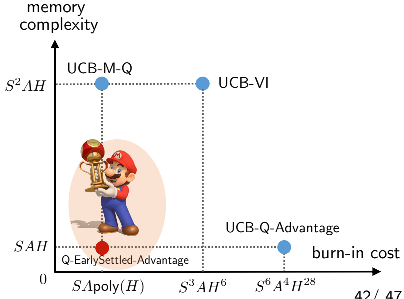
One additional key idea: early settlement of the reference as soon as it reaches a reasonable quality

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

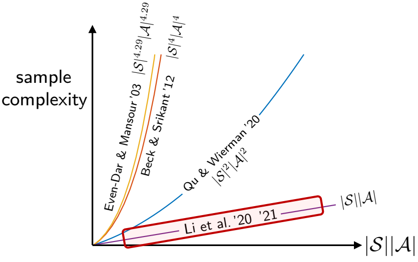
With high prob., Q-EarlySettled-Advantage *achieves*

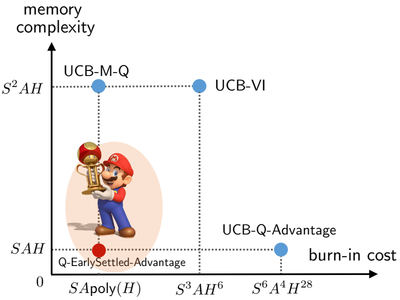
$$\mathsf{Regret}(T) \leq \widetilde{O}(\sqrt{H^2SAT} + H^6SA)$$

- regret-optimal w/ near-minimal burn-in cost in S and A
- memory-efficient $O(SAH)$
- computationally efficient: runtime $O(T)$



Summary of this part





Model-free RL can achieve memory efficiency, computational efficiency, and sample efficiency at once!

— *with some burn-in cost though*

Reference I

- "*A stochastic approximation method,*" H. Robbins, S. Monro, *Annals of mathematical statistics*, 1951
- "*Robust stochastic approximation approach to stochastic programming,*" A. Nemirovski, A. Juditsky, G. Lan, A. Shapiro, *SIAM Journal on optimization*, 2009
- "*Learning from delayed rewards,*" C. Watkins, 1989
- "*Q-learning,*" C. Watkins, P. Dayan, *Machine learning*, 1992
- "*Learning to predict by the methods of temporal differences,*" R. Sutton, *Machine learning*, 1988
- "*Analysis of temporal-diffference learning with function approximation,*" B. van Roy, J. Tsitsiklis, *IEEE transactions on automatic control*, 1997
- "*Learning Rates for Q-learning,*" E. Even-Dar, Y. Mansour, *Journal of machine learning Research*, 2003

Reference II

- "*The asymptotic convergence-rate of Q-learning,*" C. Szepesvari, *NeurIPS*, 1998
- "*Stochastic approximation with cone-contractive operators: Sharp* ℓ_∞ *bounds for Q-learning,*" M. Wainwright, arXiv:1905.06265, 2019
- "*Is Q-Learning minimax optimal? A tight sample complexity analysis,*" G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, arXiv:2102.06548, 2021
- "*Accelerating stochastic gradient descent using predictive variance reduction*," R. Johnson, T. Zhang, *NeurIPS*, 2013.
- "*Variance-reduced Q-learning is minimax optimal,*" M. Wainwright, arXiv:1906.04697, 2019
- "*Asynchronous stochastic approximation and Q-learning,*" J. Tsitsiklis, *Machine learning*, 1994

Reference III

- "*On the convergence of stochastic iterative dynamic programming algorithms,*" T. Jaakkola, M. Jordan, S. Singh, *Neural computation*, 1994
- "*Error bounds for constant step-size Q-learning,*" C. Beck, R. Srikant, *Systems and control letters*, 2012
- "*Sample complexity of asynchronous Q-learning: sharper analysis and variance reduction,*" G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, *NeurIPS* 2020
- "*Finite-time analysis of asynchronous stochastic approximation and Q-learning,*" G. Qu, A. Wierman, *COLT* 2020.
- "*Pessimistic Q-learning for offline reinforcement learning: Towards optimal sample complexity*," L. Shi, G. Li, Y. Wei, Y. Chen, Y. Chi, arXiv:2202.13890, 2022.

Reference IV

- "*The efficacy of pessimism in asynchronous Q-learning*," Y. Yan, G. Li, Y. Chen, J. Fan, arXiv:2203.07368, 2022.
- "*Asymptotically efficient adaptive allocation rules*," T. L. Lai, H. Robbins, *Advances in applied mathematics*, vol. 6, no. 1, 1985.
- "*Is Q-learning provably efficient?*" C. Jin, Z. Allen-Zhu, S. Bubeck, and M. Jordan, *NeurIPS* 2018.
- "*Almost optimal model-free reinforcement learning via reference-advantage decomposition,*" Z. Zhang, Y. Zhou, X. Ji, *NeurIPS* 2020.
- "*Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning,*" G. Li, L. Shi, Y. Chen, Y. Gu, Y. Chi, *NeurIPS* 2021.

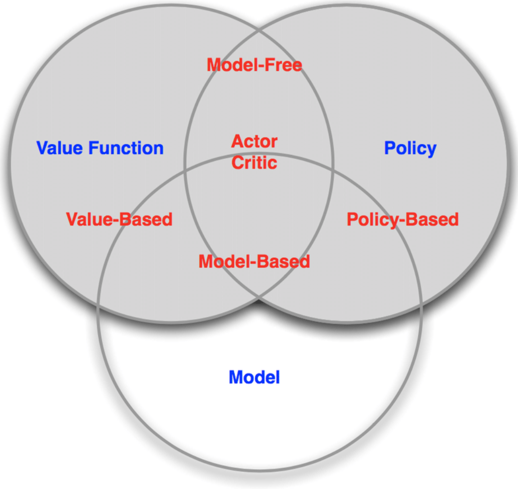
Reinforcement Learning: Fundamentals, Algorithms, and Theory (Part 3)

Yuejie Chi

Carnegie Mellon University

ICASSP, May 2022

A triad of RL approaches

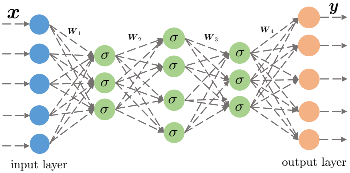


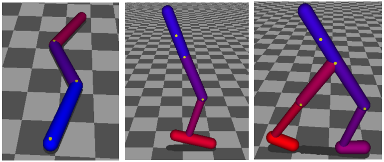
— *Figure credit: D. Silver*

Policy optimization in practice

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

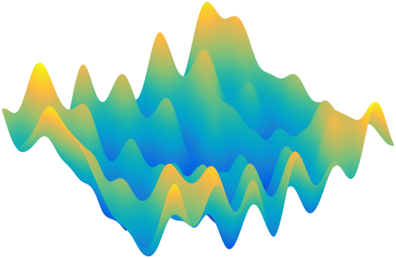
- directly optimize the policy, which is the quantity of interest;
- allow flexible differentiable parameterizations of the policy;
- work with both continuous and discrete problems.





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:

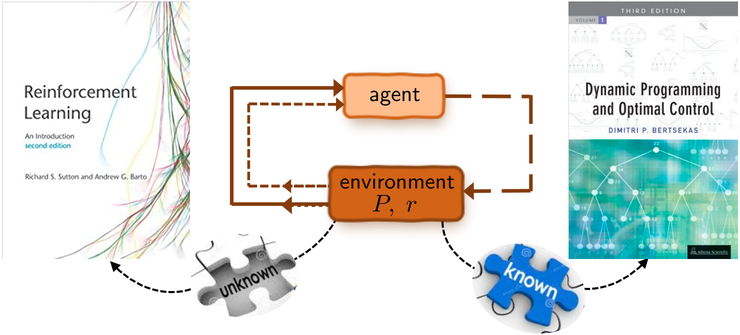
- understand finite-time convergence rates of popular heuristics;
- design fast-convergent algorithms that scale for finding policies with desirable properties.

Outline

- Backgrounds and basics
  - policy gradient method
  - policy gradient theorem
- Convergence guarantees of policy optimization
  - (natural) policy gradient methods
  - finite-time rate of global convergence
  - entropy regularization and beyond
- Concluding remarks and further pointers

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).

Examples of policy parameterization

Discrete action space: softmax parameterization with function approximation

$$\pi_\theta(a|s) \propto \exp(\phi(s,a)^\top \theta)$$

- $\phi(s,a)$ is the feature vector of each state-action pair;
- the score function $\nabla \log \pi_\theta(a|s) = \phi(s,a) - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\phi(s,\cdot)]$.

Examples of policy parameterization

Discrete action space: softmax parameterization with function approximation

$$\pi_\theta(a|s) \propto \exp(\phi(s,a)^\top \theta)$$

- $\phi(s,a)$ is the feature vector of each state-action pair;
- the score function $\nabla \log \pi_\theta(a|s) = \phi(s,a) - \mathbb{E}_{a \sim \pi_\theta(\cdot|s)}[\phi(s,\cdot)]$.

Continuous action space: Gaussian policy

$$a \sim \mathcal{N}(\mu(s), \sigma^2), \quad \mu(s) = \phi(s)^\top \theta$$

- $\phi(s)$ is the feature of each state;
- σ^2 is the variance (kept constant for simplicity);
- the score function $\nabla \log \pi_\theta(a|s) = \frac{(a-\mu(s))\phi(s)}{\sigma^2}$.

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)

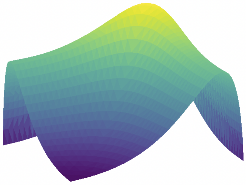
For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate.*

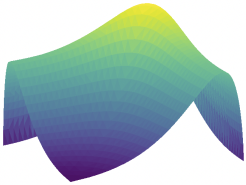
Finite-time global convergence guarantees

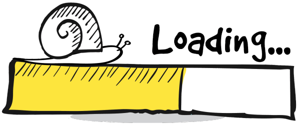
Global convergence of the PG method?



- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.

Global convergence of the PG method?

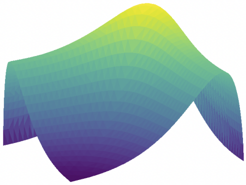


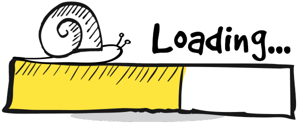


- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.
- (Mei et al., 2020) Softmax PG converges to global opt in

$O\left(\frac{1}{\epsilon}\right)$ iterations

Global convergence of the PG method?

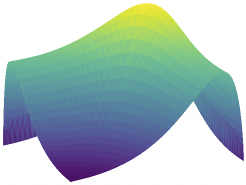


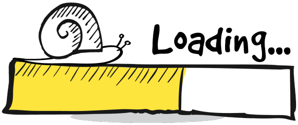


- (Agarwal et al., 2019) showed that softmax PG converges asymptotically to the global optimal policy.
- (Mei et al., 2020) Softmax PG converges to global opt in

$$c\left(|\mathcal{S}|, |\mathcal{A}|, \frac{1}{1-\gamma}, \cdots\right) O\left(\frac{1}{\epsilon}\right) \text{ iterations}$$

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

$$c\left(|\mathcal{S}|, |\mathcal{A}|, \frac{1}{1-\gamma}, \cdots\right) O\left(\frac{1}{\epsilon}\right) \text{ iterations}$$

Is the rate of PG good, bad or ugly?

A negative message

Theorem (Li, Wei, Chi, Gu, 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 $\|V^{(t)} - V^{\star}\|_{\infty} \leq 0.15$.

A negative message

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

There exists an MDP s.t. it takes softmax PG at least

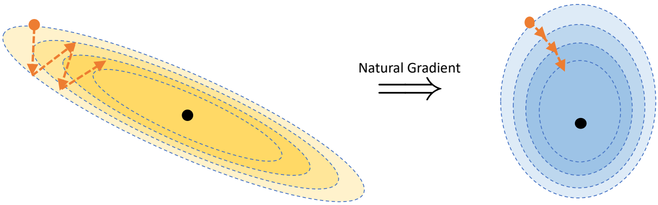
$$\frac{1}{\eta}\,|\mathcal{S}|^{2^{\Theta(\frac{1}{1-\gamma})}} \text{ iterations}$$

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:*

$$\mathcal{F}_\rho^\theta := \mathbb{E}\left[\big(\nabla_\theta \log \pi_\theta(a|s)\big)\big(\nabla_\theta \log \pi_\theta(a|s)\big)^\top \right].$$

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

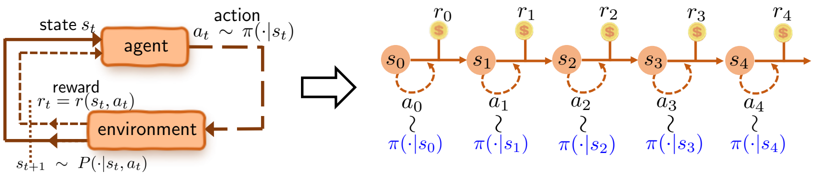
$$V^{(t)}(\rho) \geq V^{\star}(\rho) - \left(\frac{\log |\mathcal{A}|}{\eta} + \frac{1}{(1-\gamma)^2} \right) \frac{1}{t}.$$

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

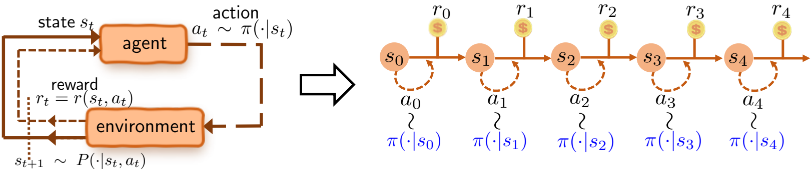


To encourage exploration, promote the stochasticity of the policy using the **"soft"** value function (Williams and Peng, 1991):

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \tau \mathcal{H}\big(\pi(\cdot|s_t)\big)\Big) \,\Big|\, s_0 = s\right]$$

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):

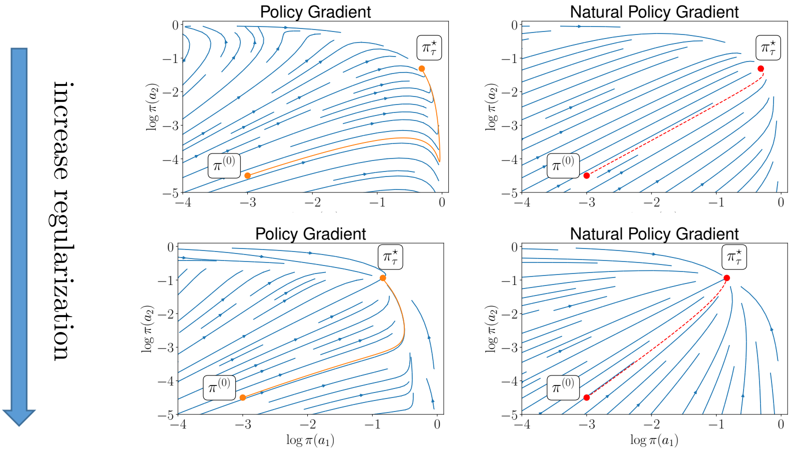
$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \tau \mathcal{H}(\pi(\cdot|s_t))\Big) \,\Big|\, s_0 = s\right]$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

$$\text{maximize}_\theta \quad V_\tau^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V_\tau^{\pi_\theta}(s)\right]$$

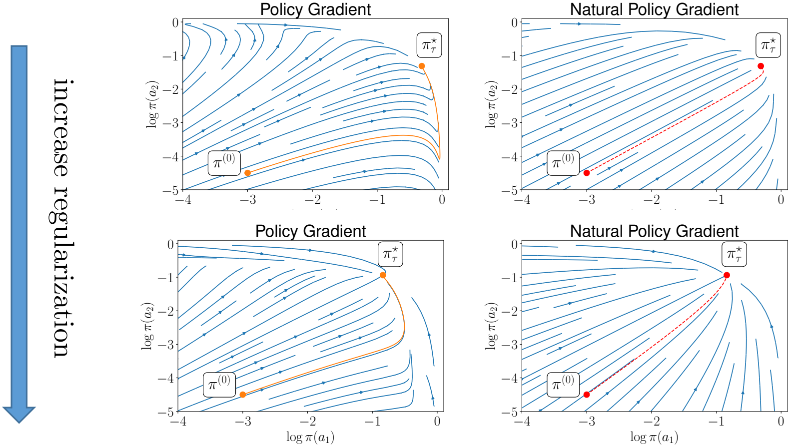
Entropy-regularized natural gradient helps!

Toy example: a bandit with 3 arms of rewards 1, 0.9 and 0.1.



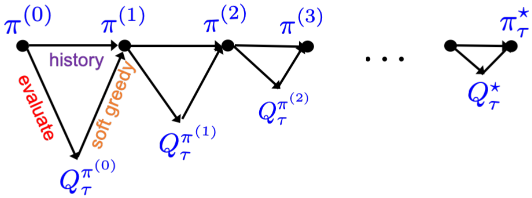
Entropy-regularized natural gradient helps!

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

$$\pi^{(t+1)}(\cdot|s) \propto \underbrace{\pi^{(t)}(\cdot|s)}_{\textit{current policy}}{}^{1-\frac{\eta\tau}{1-\gamma}} \underbrace{\exp(Q_{\tau}^{(t)}(s,\cdot)/\tau)}_{\textit{soft greedy}}{}^{\frac{\eta\tau}{1-\gamma}}$$

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 $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \leq \epsilon$, 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 ($0 < \eta < \frac{1-\gamma}{\tau}$):**

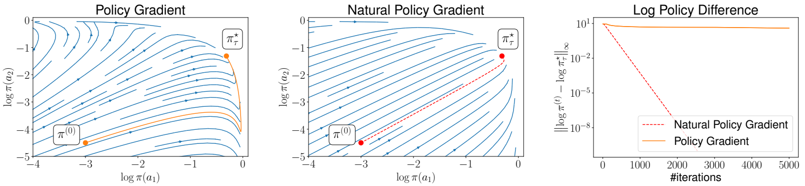
$$\frac{1}{\eta\tau} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

- **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

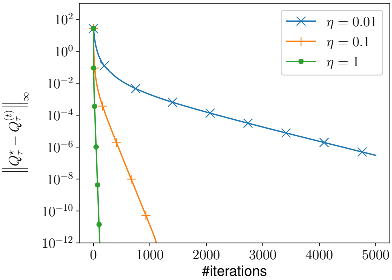
$$V_\tau^\star(\rho) - V_\tau^{(t)}(\rho) \le \left(V_\tau^\star(\rho) - V_\tau^{(0)}(\rho)\right)$$
$$\cdot \exp\left(- \frac{(1-\gamma)^4 t}{(8/\tau + 4 + 8\log|\mathcal{A}|)|\mathcal{S}|} \left\| \frac{d_\rho^{\pi_\tau^\star}}{\rho} \right\|_\infty^{-1} \min_s \rho(s) \underbrace{\left(\inf_{0\le k\le t-1} \min_{s,a} \pi^{(k)}(a|s) \right)^2}_{\text{can be exponential in } |\mathcal{S}| \text{ and } \frac{1}{1-\gamma}} \right)$$

Much faster convergence of entropy-regularized NPG
at a **dimension-free** rate!

Comparison with unregularized NPG

Regularized NPG

$\tau = 0.001$

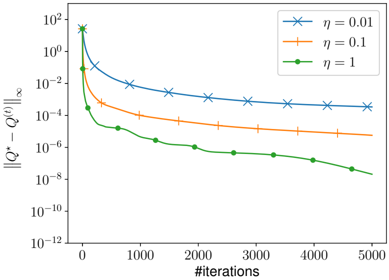


Linear rate: $\frac{1}{\eta\tau}\log\left(\frac{1}{\epsilon}\right)$

Ours

Vanilla NPG

$\tau = 0$

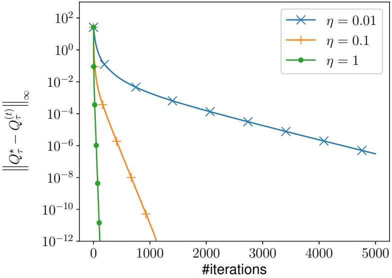


Sublinear rate: $\frac{1}{\min\{\eta,(1-\gamma)^2\}\epsilon}$

(Agarwal et al. 2019)

Regularized NPG

$\tau = 0.001$

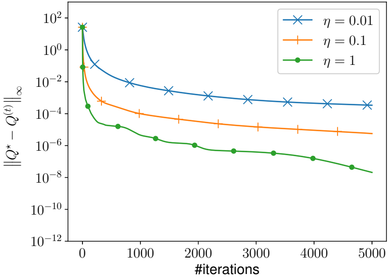


Linear rate: $\frac{1}{\eta\tau}\log\left(\frac{1}{\epsilon}\right)$

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

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

Soft Bellman equation: $Q_\tau^\star$ is *unique* solution to

$$\mathcal{T}_\tau(Q_\tau^\star) = Q_\tau^\star$$

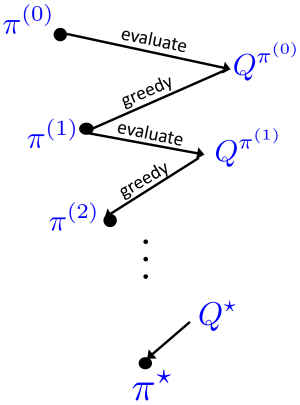
γ**-contraction of soft Bellman operator:**

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

Richard Bellman

Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

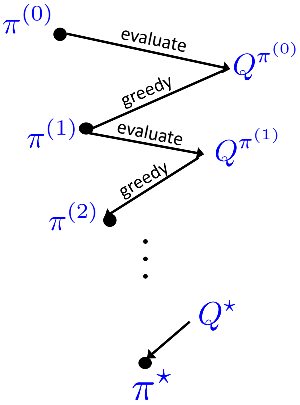
Policy iteration



Bellman operator

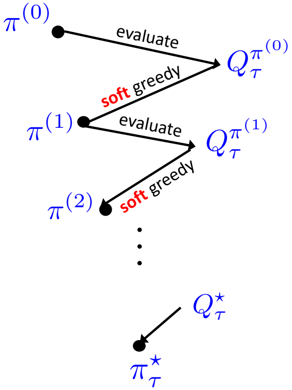
Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

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

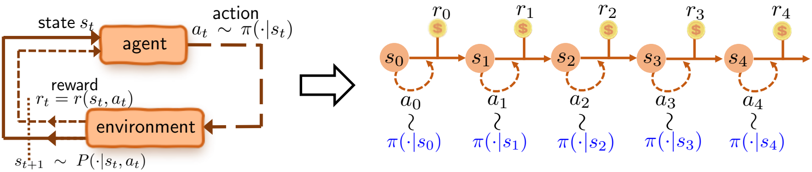
sparse exploration

Tsallis entropy

constrained and safe RL

log-barrier

Regularized RL in general form

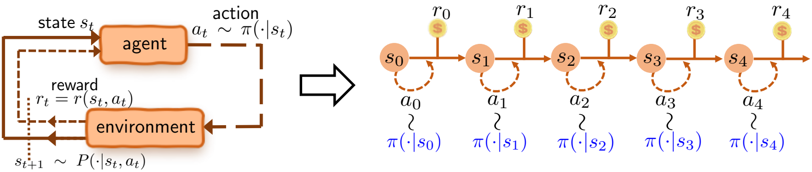


The regularized value function is defined as

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t - \tau h_{s_t}(\pi(\cdot|s_t))\Big) \,\Big|\, s_0 = s\right],$$

where h_s is convex (and possibly nonsmooth) w.r.t. $\pi(\cdot|s)$.

Regularized RL in general form



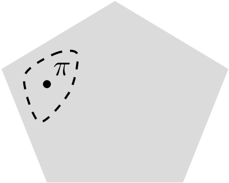
The regularized value function is defined as

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t - \tau h_{s_t}(\pi(\cdot|s_t))\Big) \,\Big|\, s_0 = s\right],$$

where h_s is convex (and possibly nonsmooth) w.r.t. $\pi(\cdot|s)$.

$$\text{maximize}_\pi \quad V_\tau^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V_\tau^\pi(s)\right]$$

Detour: a mirror descent view of entropy-regularized NPG



Entropy-regularized NPG = mirror descent with KL divergence (Lan, 2021; Shani et al., 2020):

$$\pi^{(t+1)}(\cdot|s) = \underset{p\in\Delta(\mathcal{A})}{\operatorname{argmin}} \big\langle -Q_\tau^{(t)}(s,\cdot),\, p\big\rangle - \tau\mathcal{H}(p) + \frac{1}{\eta}\mathsf{KL}\big(p\|\pi^{(t)}(\cdot|s)\big)$$

$$\propto \underbrace{\pi^{(t)}(\cdot|s)}_{\text{current policy}}{}^{\frac{1}{1+\eta\tau}} \underbrace{\exp(Q_\tau^{(t)}(s,\cdot)/\tau)}_{\text{soft greedy}}{}^{\frac{\eta\tau}{1+\eta\tau}}$$

for all $s \in \mathcal{S}$.

Generalized policy mirror descent (GPMD)

Definition (Generalized Bregman divergence, Kiwiel 1997)

The generalized Bregman divergence w.r.t. to a convex $h : \Delta(\mathcal{A}) \mapsto \mathbb{R}$ is defined as:

$$
\begin{aligned}
D_h(p, q; g) &= h(p) - h(q) - \langle g, p - q \rangle \\
&= h(p) - h(q) - \langle g - c \cdot \mathbf{1}, p - q \rangle,
\end{aligned}
$$

for $p, q \in \Delta(\mathcal{A})$, where $g \in \partial h(q)$ and $c \in \mathbb{R}$.

Generalized policy mirror descent (GPMD)

Definition (Generalized Bregman divergence, Kiwiel 1997)

The generalized Bregman divergence w.r.t. to a convex $h : \Delta(\mathcal{A}) \mapsto \mathbb{R}$ is defined as:

$$\begin{aligned} D_h(p, q; g) &= h(p) - h(q) - \langle g, p - q \rangle \\ &= h(p) - h(q) - \langle g - c \cdot \mathbf{1}, p - q \rangle, \end{aligned}$$

for $p, q \in \Delta(\mathcal{A})$, where $g \in \partial h(q)$ and $c \in \mathbb{R}$.

A natural idea

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

$$\pi^{(t+1)}(\cdot|s) = \operatorname*{argmin}_{p \in \Delta(\mathcal{A})} \langle -Q_\tau(s, \cdot), p \rangle + \tau h_s(p) + \frac{1}{\eta} D_{h_s}(p, \pi^{(t)}(\cdot|s); \partial h_s(\pi^{(t)}(\cdot|s)))$$

PMD with Generalized Bregman Divergence (**GPMD**)

Plugging in a recursive surrogate $\{\xi^{(t)}\}$ of $\partial h_s(\pi^{(t)}(\cdot|s))$, we obtain the formal algorithm.

Generalized policy mirror descent (GPMD) method

For $t = 0, 1, \cdots$, *update*

$$\pi^{(t+1)}(\cdot|s) = \operatorname*{argmin}_{p \in \Delta(\mathcal{A})} \langle -Q_\tau(s,\cdot), p \rangle + \tau h_s(p) + \frac{1}{\eta} D_{h_s}(p, \pi^{(t)}(\cdot|s); \xi^{(t)}(s,\cdot)),$$

and

$$\xi^{(t+1)}(s,\cdot) = \frac{1}{1+\eta\tau}\xi^{(t)}(s,\cdot) + \frac{\eta}{1+\eta\tau} Q_\tau^{(t)}(s,\cdot).$$

The subproblem does not admit closed-form solution in general.

Linear convergence with exact gradient

Exact oracle: perfect evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$; exact solution to subproblems.

— *Read our paper for the inexact case!*

Linear convergence with exact gradient

Exact oracle: perfect evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$; exact solution to subproblems.

— *Read our paper for the inexact case!*

Theorem (Zhan*, Cen*, Huang, Chen, Lee, Chi '21)

For any learning rate $\eta > 0$, the GPMD updates satisfy

- **Linear convergence of soft Q-functions:**

$$\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \leq C_1 \gamma \left(1 - \frac{\eta\tau(1-\gamma)}{1+\eta\tau}\right)^t$$

where $C_1 = \|Q_\tau^\star - Q_\tau^{(0)}\|_\infty + \frac{2}{1+\eta\tau}\|Q_\tau^\star - \tau\xi^{(0)}\|_\infty$.

Implications

To reach $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \le \epsilon$, the iteration complexity is at most

- **General learning rates ($\eta > 0$):**

$$\frac{1+\eta\tau}{\eta\tau(1-\gamma)} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

- **Regularized policy iteration ($\eta = \infty$):**

$$\frac{1}{1-\gamma} \log\left(\frac{\|Q_\tau^\star - Q_\tau^{(0)}\|_\infty \gamma}{\epsilon}\right)$$

Implications

To reach $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \le \epsilon$, the iteration complexity is at most

- **General learning rates ($\eta > 0$):**

$$\frac{1+\eta\tau}{\eta\tau(1-\gamma)} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

- **Regularized policy iteration ($\eta = \infty$):**

$$\frac{1}{1-\gamma} \log\left(\frac{\|Q_\tau^\star - Q_\tau^{(0)}\|_\infty \gamma}{\epsilon}\right)$$

Global linear convergence of GPMD at a **dimension-free** rate!

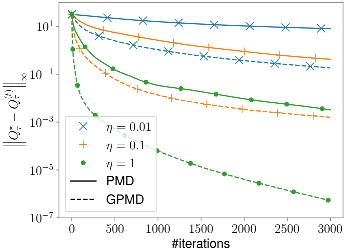
Comparison with PMD (Lan, 2021)

Policy mirror descent (PMD) method (Lan, 2021)

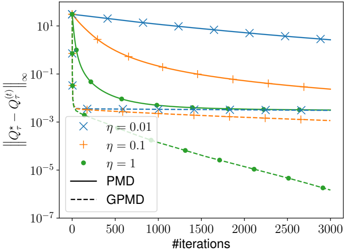
For $t = 0, 1, \cdots,$

$$\pi^{(t+1)}(\cdot|s) = \operatorname*{argmin}_{p \in \Delta(\mathcal{A})} \langle -Q_\tau(s,\cdot), p\rangle + \tau h_s(p) + \frac{1}{\eta}\mathsf{KL}(p||\pi^{(t)}(\cdot|s))$$

h_s = **Tsallis Entropy**



h_s = **Log Barrier**



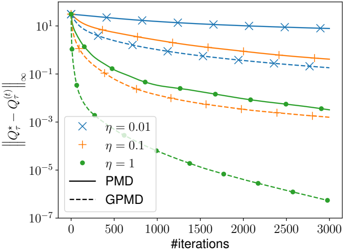
Comparison with PMD (Lan, 2021)

Policy mirror descent (PMD) method (Lan, 2021)

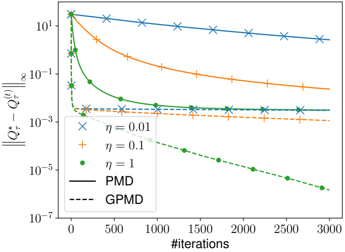
For $t = 0, 1, \cdots,$

$$\pi^{(t+1)}(\cdot|s) = \underset{p \in \Delta(\mathcal{A})}{\operatorname{argmin}} \langle -Q_\tau(s,\cdot), p\rangle + \tau h_s(p) + \frac{1}{\eta}\mathsf{KL}(p||\pi^{(t)}(\cdot|s))$$

h_s **= Tsallis Entropy**



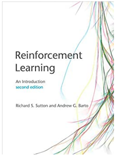
h_s **= Log Barrier**

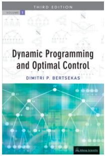


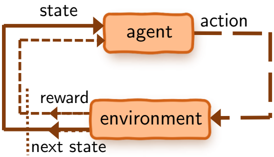
GPMD achieves faster convergence than PMD!

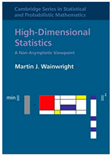
Concluding Remarks

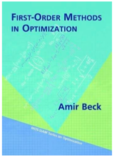
Concluding remarks











Understanding non-asymptotic performances of RL algorithms is a fruitful playground!

Future directions:

- function approximation
- multi-agent RL
- offline RL
- many more...

Beyond the tabular setting

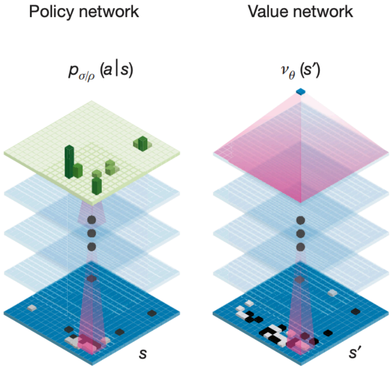
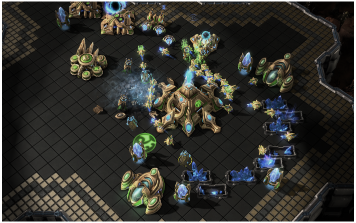


Figure credit: (Silver et al., 2016)

- function approximation for dimensionality reduction
- Provably efficient RL algorithms under minimal assumptions

(Osband and Van Roy, 2014; Dai et al., 2018; Du et al., 2019; Jin et al., 2020)

Multi-agent RL



- **Competitive setting:** finding Nash equilibria for Markov games
- **Collaborative setting:** multiple agents jointly optimize the policy to maximize the total reward

(Zhang, Yang, and Basar, 2021; Cen, Wei, and Chi, 2021)

Offline RL

Can we design RL algorithms based on history data?

(Rashidinejad et al., 2021; Xie et al., 2021; Li et al., 2022)

Bibliography I

Disclaimer: this straw-man list is by no means exhaustive (in fact, it is quite the opposite given the fast pace of the field), and biased towards materials most related to this tutorial; readers are invited to further delve into the references therein to gain a more complete picture.

Books and monographs:

- Sutton and Barto. *Reinforcement learning: An introduction, 2nd edition*. MIT press, 2018.
- Agarwal, Jiang, Kakade, and Sun. *Reinforcement learning: Theory and algorithms*, monograph, 2021+.
- Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.
- Szepesvári. *Algorithms for reinforcement learning*. Synthesis lectures on artificial intelligence and machine learning, 2010.
- Bertsekas and Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Policy optimization:

- Williams. "*Simple statistical gradient-following algorithms for connectionist reinforcement learning.*" Machine Learning, 1992.
- Sutton, McAllester, Singh, and Mansour. "*Policy gradient methods for reinforcement learning with function approximation.*" NeurIPS 1999.
- Kakade. "*A natural policy gradient.*" NeurIPS 2001.
- Fazel, Ge, Kakade, and Mesbahi. "*Global convergence of policy gradient methods for the linear quadratic regulator.*" ICML 2018.
- Agarwal, Kakade, Lee, and Mahajan. "*On the theory of policy gradient methods: Optimality, approximation, and distribution shift.*" Journal of Machine Learning Research, 2021.
- Mei, Xiao, Szepesvári, and Schuurmans. "*On the global convergence rates of softmax policy gradient methods.*" ICML 2020.
- Bhandari and Russo. "*Global optimality guarantees for policy gradient methods.*" arXiv preprint arXiv:1906.01786, 2019.

Bibliography III

- Cai, Yang, Jin, and Wang. "*Provably efficient exploration in policy optimization.*" ICML 2020.
- Shani, Efroni, and Mannor."*Adaptive trust region policy optimization: Global convergence and faster rates for regularized MDPs.*" AAAI 2020.
- Li, Gen, Wei, Chi, Gu, and Chen. "*Softmax policy gradient methods can take exponential time to converge.*" arXiv preprint arXiv:2102.11270, 2021.
- Cen, Cheng, Chen, Wei, and Chi. "*Fast global convergence of natural policy gradient methods with entropy regularization.*" Operations Research, 2021+.
- Zhan, Cen, Huang, Chen, Lee, and Chi."*Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence.*" arXiv preprint arXiv:2105.11066, 2021.
- Lan. "*Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes.*" arXiv preprint arXiv:2102.00135, 2021.

Bibliography IV

- Liu, Zhang, Basar, and Yin. "*An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods.*" NeurIPS 2020.
- Zhang, Koppel, Bedi, Szepesvári, and Wang. "*Variational policy gradient method for reinforcement learning with general utilities.*" NeurIPS 2020.
- Cen, Wei, and Chi. "*Fast policy extragradient methods for competitive games with entropy regularization.*" arXiv preprint arXiv:2105.15186, 2021.

Additional ad-hoc pointers:

- Neu, Jonsson, and Gómez. "*A unified view of entropy-regularized Markov Decision Processes.*" arXiv preprint arXiv:1705.07798, 2017.
- Dai, Shaw, Li, Xiao, He, Liu, Chen, and Song."*SBEED: Convergent reinforcement learning with nonlinear function approximation.*" ICML 2018.
- Geist, Scherrer, and Pietquin. "*A theory of regularized Markov Decision Processes.*" ICML 2019.

Bibliography V

- Du, Kakade, Wang, and Yang. "*Is a good representation sufficient for sample efficient reinforcement learning?*" ICLR 2019.
- Jin, Yang, Wang, and Jordan. "*Provably efficient reinforcement learning with linear function approximation.*" COLT 2020.
- Zhang, Yang, and Basar. "*Multi-agent reinforcement learning: A selective overview of theories and algorithms.*" Handbook of Reinforcement Learning and Control, 2021.
- Rashidinejad, Zhu, Ma, Jiao, and Russell. "*Bridging offline reinforcement learning and imitation learning: A tale of pessimism.*" arXiv preprint arXiv:2103.12021, 2021.
- Li, Shi, Chen, Chi, Wei, "*Settling the sample complexity of model-based offline reinforcement learning.*" arXiv preprint arXiv:2204.05275, 2022.

Thanks!

https://users.ece.cmu.edu/~yuejiec/