

Statistical and Algorithmic Foundations of Reinforcement Learning

Yuejie Chi
Yale University

DeepLearn 2026
Tutorial Part 1

Acknowledgement



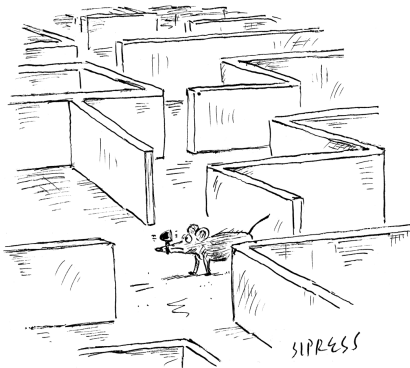
Yuxin Chen
UPenn



Yuting Wei
UPenn

Reinforcement learning (RL)

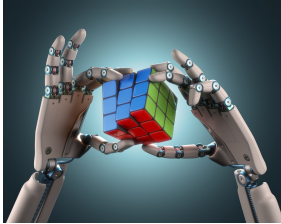
In RL, an agent learns by interacting with an *unknown* environment through trial-and-error to maximize long-term total reward.



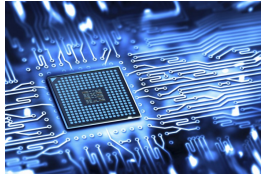
“Recalculating ... recalculating ...”



More successes of RL since AlphaGo



robotics



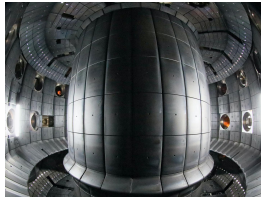
chip designs



resource management



strategic games

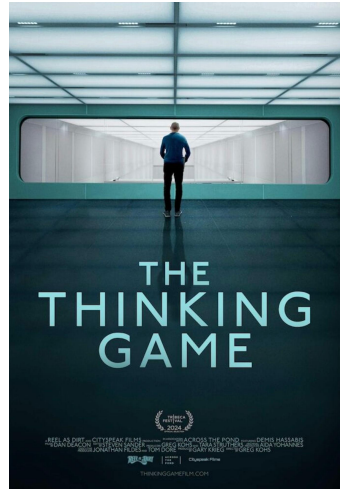


nuclear plant control



UAV and drones

One more: RL for thinking models



Turing Award Goes to 2 Pioneers of Artificial Intelligence

Andrew Barto and Richard Sutton developed reinforcement learning, a technique vital to chatbots like ChatGPT.

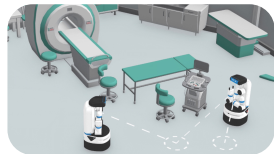


RL holds great promise in accelerating scientific, engineering and societal discoveries.

RL holds great promise in accelerating scientific, engineering and societal discoveries.

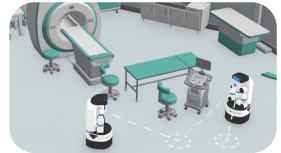
How do we build the statistical and algorithmic foundations of RL to help realizing its potential?

Reinforcement Learning



In RL, agent(s) often learn by probing the environment

Reinforcement Learning

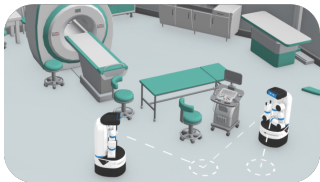


In RL, agent(s) often learn by probing the environment

- unknown environment
- delayed feedback
- explosion of dimensionality
- nonconvexity

Data efficiency

Data collection might be expensive, time-consuming, or high-stakes



clinical trials



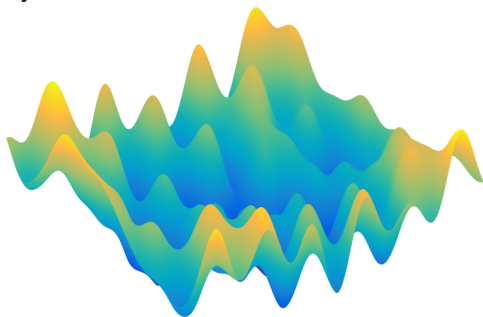
self-driving cars

Calls for design of sample-efficient RL algorithms!

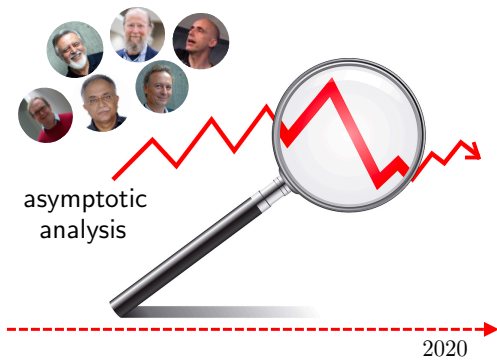
Computational efficiency

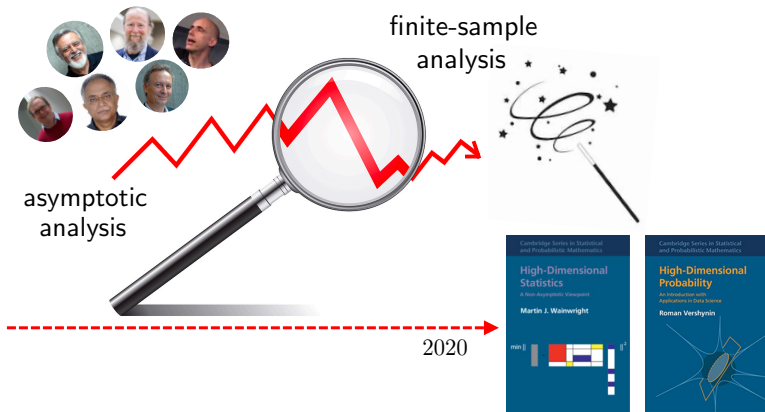
Running RL algorithms might take a long time ...

- enormous state-action space
- nonconvexity



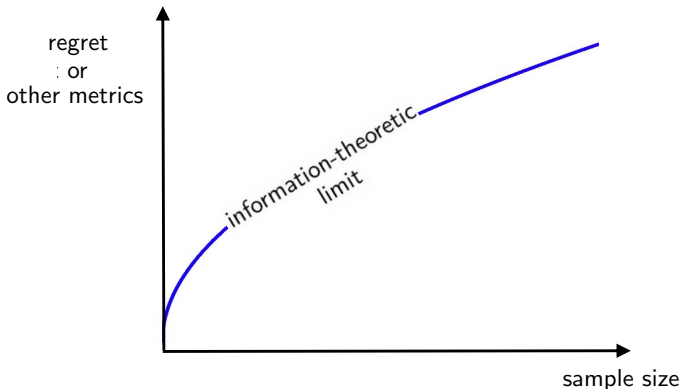
Calls for computationally efficient RL algorithms!



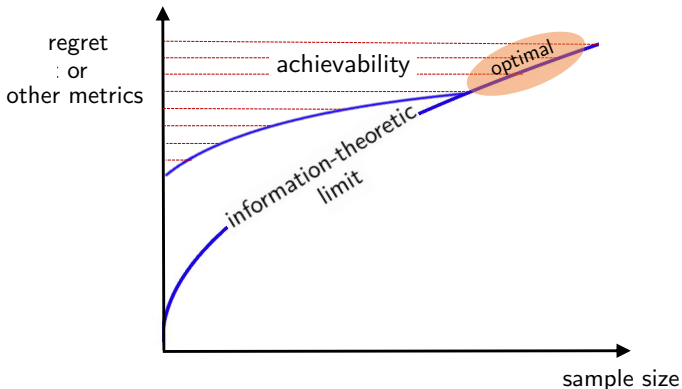


Understanding efficiency of contemporary RL requires a modern suite of non-asymptotic analysis

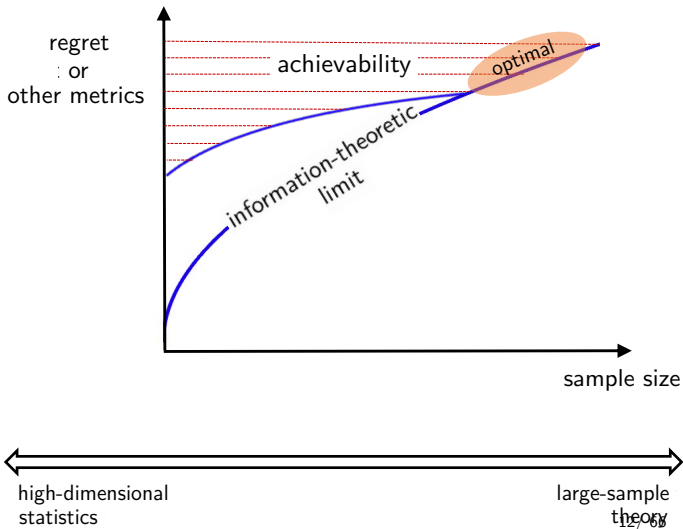
Sample complexity issues that permeate state-of-the-art RL theory



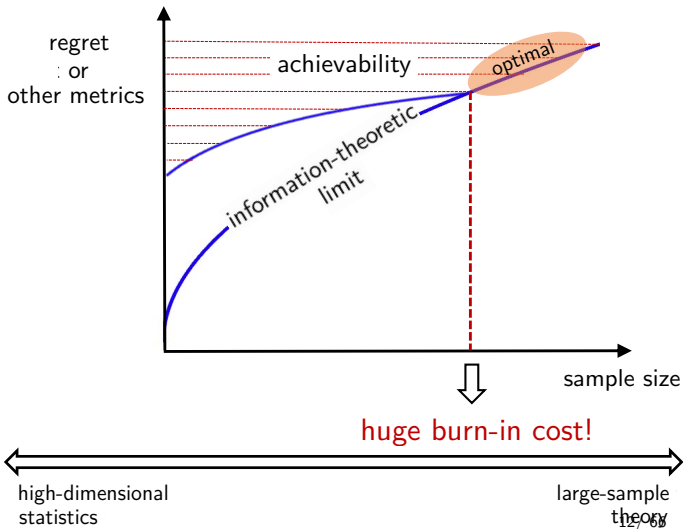
Sample complexity issues that permeate state-of-the-art RL theory



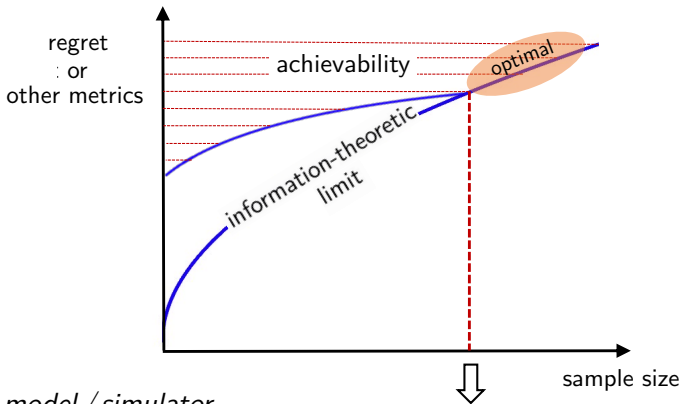
Sample complexity issues that permeate state-of-the-art RL theory



Sample complexity issues that permeate state-of-the-art RL theory

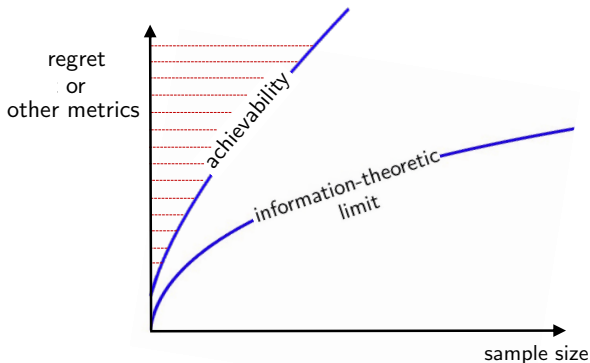


Sample complexity issues that permeate state-of-the-art RL theory



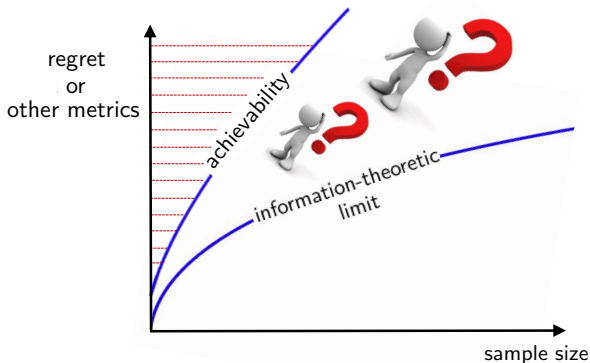
- *generative model / simulator*
- *online RL*
- *offline RL*
- ...

Sample complexity issues that permeate state-of-the-art RL theory



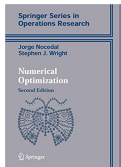
- *multi-agent RL*
- *partially observable MDPs*
- ...

Sample complexity issues that permeate state-of-the-art RL theory

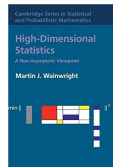


- *multi-agent RL*
- *partially observable MDPs*
- ...

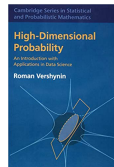
This tutorial



(large-scale) optimization

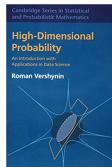
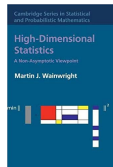
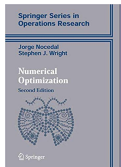


(high-dimensional) statistics



Design **sample-** and **computationally**-efficient RL algorithms

This tutorial



(large-scale) optimization

(high-dimensional) statistics

Design **sample-** and **computationally**-efficient RL algorithms

Part 1. basics, RL w/ a generative model

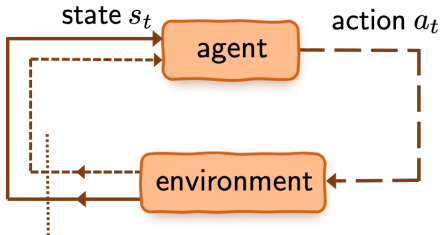
Part 2. online / offline RL, multi-agent / robust RL

Part 3. policy optimization

Part 1

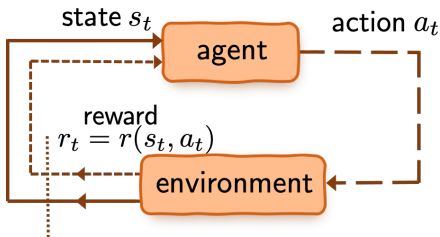
1. Basics: Markov decision processes
2. RL w/ a generative model (simulator)
 - model-based algorithms (a “plug-in” approach)
 - model-free algorithms

Markov decision process (MDP)



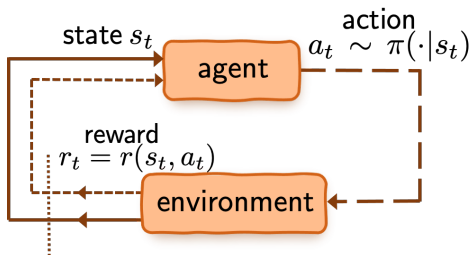
- $\mathcal{S} = \{1, \dots, S\}$: state space (containing S states)
- $\mathcal{A} = \{1, \dots, A\}$: action space (containing A actions)

Markov decision process (MDP)



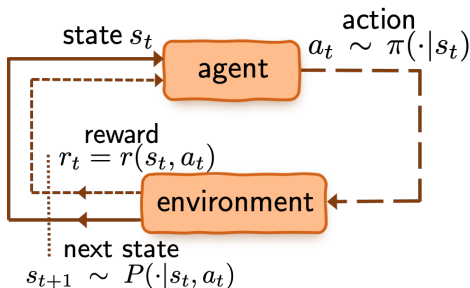
- $\mathcal{S} = \{1, \dots, S\}$: state space (containing S states)
- $\mathcal{A} = \{1, \dots, A\}$: action space (containing A actions)
- $r(s, a) \in [0, 1]$: immediate reward

Discounted infinite-horizon MDPs



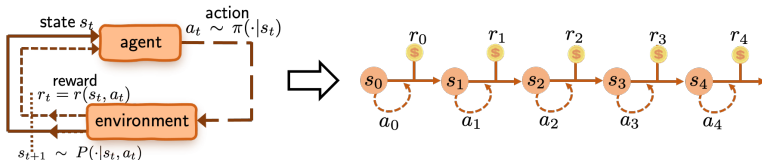
- $\mathcal{S} = \{1, \dots, S\}$: state space (containing S states)
- $\mathcal{A} = \{1, \dots, A\}$: action space (containing A actions)
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot | s)$: policy (or action selection rule)

Discounted infinite-horizon MDPs



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

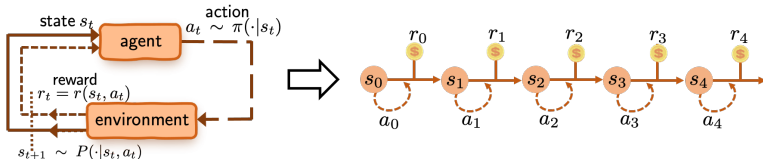
Value function



Value of policy π : cumulative **discounted** reward

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

Value function

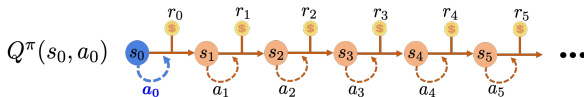


Value of policy π : cumulative **discounted** reward

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

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

Q-function (action-value function)

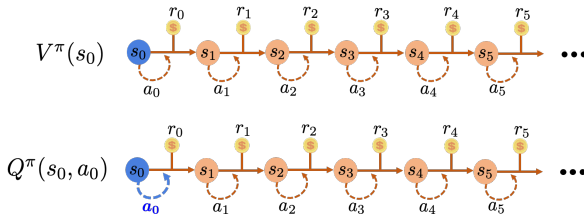


Q-function of policy π :

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

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

Q-function (action-value function)

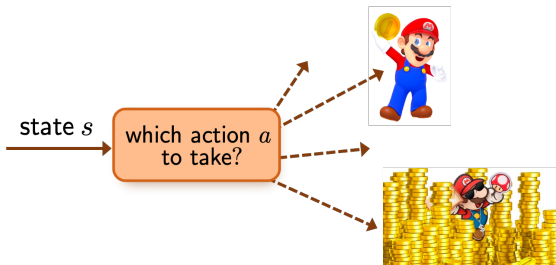


Q-function of policy π :

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

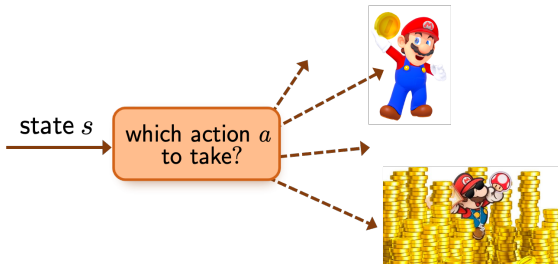
- $(\cancel{a_0}, s_1, a_1, s_2, a_2, \dots)$: induced by policy π

Optimal policy and optimal value



- **optimal policy** π^* : maximizing value function $\max_{\pi} V^{\pi}$

Optimal policy and optimal value



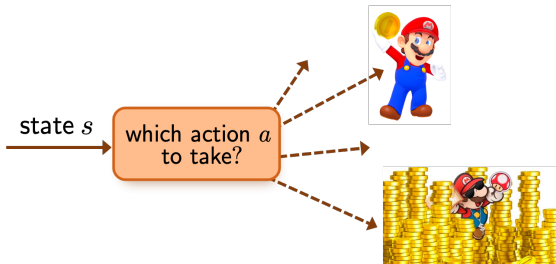
- **optimal policy** π^* : maximizing value function $\max_{\pi} V^{\pi}$

Theorem (Puterman'94)

For infinite horizon discounted MDP, there always exists a deterministic policy π^ , such that*

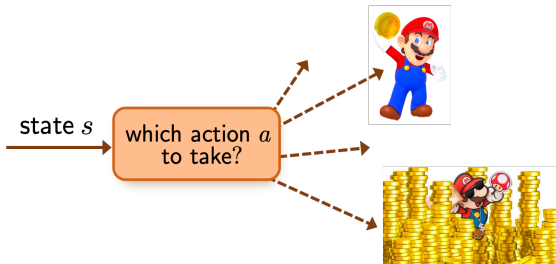
$$V^{\pi^*}(s) \geq V^{\pi}(s), \quad \forall s, \text{ and } \pi.$$

Optimal policy and optimal value



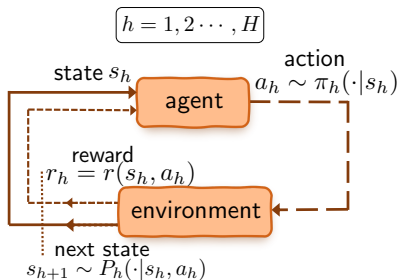
- **optimal policy** π^* : maximizing value function $\max_{\pi} V^{\pi}$
- **optimal value / Q function**: $V^* := V^{\pi^*}$, $Q^* := Q^{\pi^*}$

Optimal policy and optimal value



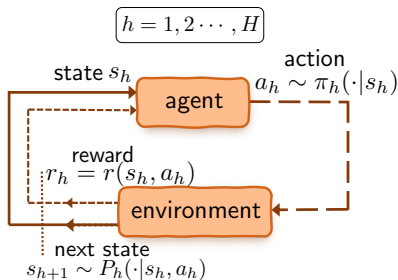
- **optimal policy** π^* : maximizing value function $\max_{\pi} V^{\pi}$
- **optimal value / Q function**: $V^* := V^{\pi^*}$, $Q^* := Q^{\pi^*}$
- A question to keep in mind: *how to find optimal π^* ?*

Finite-horizon MDPs (nonstationary)



- 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)
- $P_h(\cdot | s, a)$: transition probabilities in step h

Finite-horizon MDPs (nonstationary)

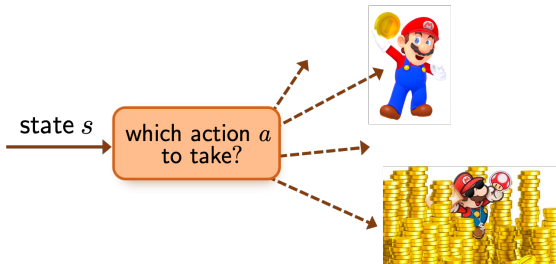


$$\text{value function: } V_{\mathbf{h}}^{\pi}(s) := \mathbb{E} \left[\sum_{t=\mathbf{h}}^H r_t(s_t, a_t) \mid s_h = s \right]$$

$$\text{Q-function: } Q_{\mathbf{h}}^{\pi}(s, a) := \mathbb{E} \left[\sum_{t=\mathbf{h}}^H r_t(s_t, a_t) \mid s_h = s, a_{\mathbf{h}} = a \right]$$



Optimal policy and optimal value



- **optimal policy** π^* : maximizing value function at all steps
- **optimal value / Q function**: $V_h^* := V_h^{\pi^*}$, $Q_h^* := Q_h^{\pi^*}$, $\forall h$
- **Question**: *how to find optimal π^* ?*

*Basic dynamic programming algorithms
when MDP specification is **known***

A simpler problem: **policy evaluation**

— given MDP $\mathcal{M}$ and policy π , how to compute V^π , Q^π ?

A simpler problem: **policy evaluation**

— given MDP $\mathcal{M}$ and policy π , how to compute V^π , Q^π ?

solution: Bellman's consistency equation

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

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



Richard Bellman

A simpler problem: **policy evaluation**

— given MDP $\mathcal{M}$ and policy π , how to compute V^π , Q^π ?

solution: Bellman's consistency equation

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

- one-step look-ahead



Richard Bellman

A simpler problem: **policy evaluation**

— given MDP $\mathcal{M}$ and policy π , how to compute V^π , Q^π ?

solution: **Bellman's consistency equation**

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

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

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

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



Richard Bellman

Back to main question: how to find optimal policy π^* ?

solution: Bellman's optimality principle

- Bellman operator:

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

- one-step look-ahead
- γ -contraction: $\|\mathcal{T}(Q_1) - \mathcal{T}(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$

Back to main question: how to find optimal policy π^* ?

solution: Bellman's optimality principle

- Bellman operator:

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

- one-step look-ahead
- γ -contraction: $\|\mathcal{T}(Q_1) - \mathcal{T}(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$

- Bellman equation: Q^* is *unique* solution to

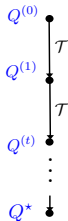
$$\mathcal{T}(Q^*) = Q^*$$

Two dynamic programming algorithms

Value iteration (VI)

For $t = 0, 1, \dots$

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

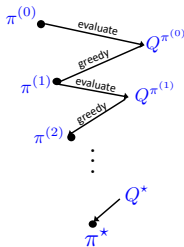


Policy iteration (PI)

For $t = 0, 1, \dots$

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

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



Iteration complexity

Theorem (Linear convergence of policy/value iteration)

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

Iteration complexity

Theorem (Linear convergence of policy/value iteration)

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

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

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

Iteration complexity

Theorem (Linear convergence of policy/value iteration)

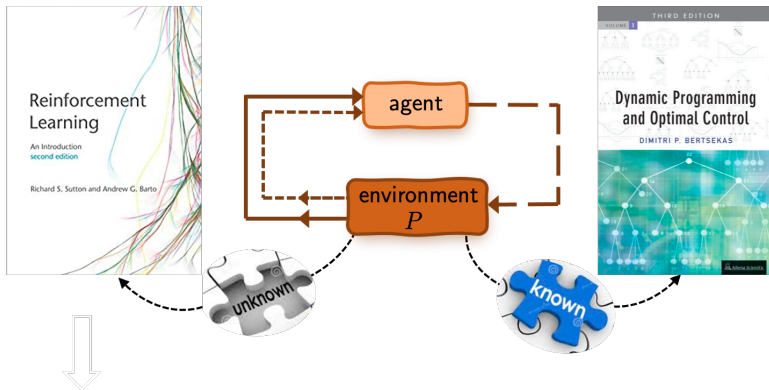
$$\|Q^{(t)} - Q^*\|_{\infty} \leq \gamma^t \|Q^{(0)} - Q^*\|_{\infty}$$

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

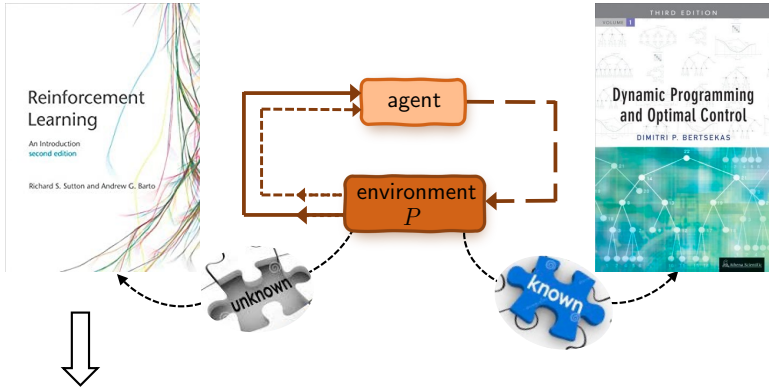
$$\frac{1}{1 - \gamma} \log \left(\frac{\|Q^{(0)} - Q^*\|_{\infty}}{\varepsilon} \right) \text{ iterations}$$

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

When the model is unknown ...

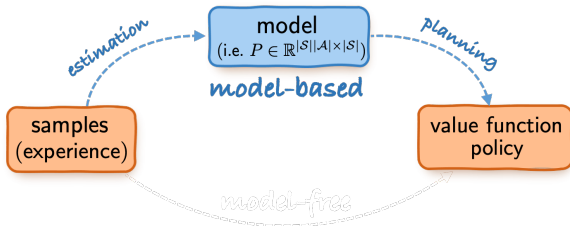


When the model is unknown ...



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

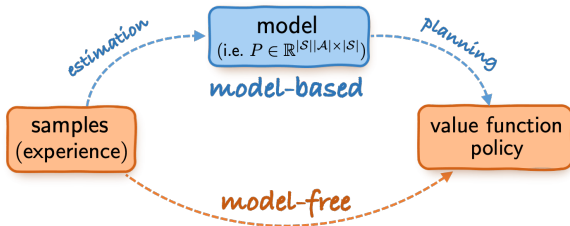
Two approaches



Model-based approach (“plug-in”)

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

Two approaches



Model-based approach (“plug-in”)

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

Model-free approach

— learning w/o estimating the model explicitly

Sampling mechanisms

1. RL w/ a generative model (a.k.a. simulator)
 - can query arbitrary state-action pairs to draw samples

Sampling mechanisms

1. RL w/ a generative model (a.k.a. simulator)
 - can query arbitrary state-action pairs to draw samples
2. online RL
 - execute MDP in real time to obtain sample trajectories

Sampling mechanisms

1. RL w/ a generative model (a.k.a. simulator)
 - can query arbitrary state-action pairs to draw samples
2. online RL
 - execute MDP in real time to obtain sample trajectories
3. offline RL
 - use pre-collected historical data

Sampling mechanisms

1. RL w/ a generative model (a.k.a. simulator)
 - can query arbitrary state-action pairs to draw samples
2. online RL
 - execute MDP in real time to obtain sample trajectories
3. offline RL
 - use pre-collected historical data

Question: *how many samples are sufficient to learn an ε -optimal policy?*

$$\underbrace{\hat{V}^{\pi} \geq V^* - \varepsilon}$$

Exploration vs exploitation

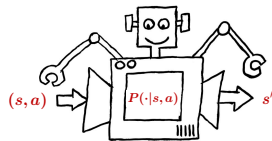
Exploration



offline RL



online RL



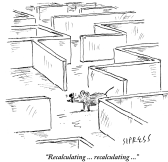
generative model

Exploration vs exploitation

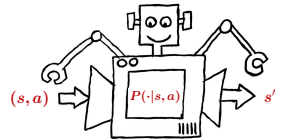
Exploration



offline RL



online RL



generative model

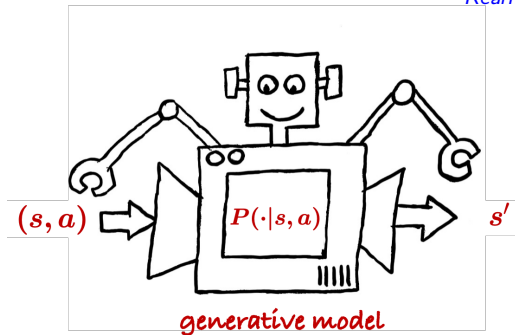
Varying levels of trade-offs between exploration and exploitation.

Part 1

1. Basics: Markov decision processes
2. RL w/ a generative model (simulator)
 - model-based algorithms (a “plug-in” approach)
 - model-free algorithms

A generative model / simulator

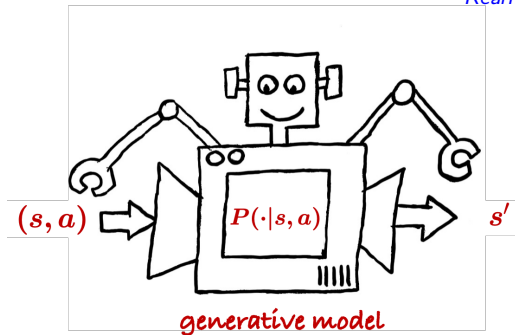
— *Kearns and Singh, 1999*



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

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 $\hat{\pi}$ based on samples (in total $SA \times N$)

ℓ_∞ -**sample complexity**: how many samples are required to
learn an ε -optimal policy?

$$\forall s: \underbrace{\widehat{V}^\pi(s) \geq V^*(s) - \varepsilon}$$

Minimax lower bound

Theorem (minimax lower bound; Azar et al., 2013)

For all $\varepsilon \in [0, \frac{1}{1-\gamma})$, there exists some MDP such that the total number of samples need to be **at least**

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

to achieve $V^* - V^{\hat{\pi}} \leq \varepsilon$, where $\hat{\pi}$ is the output of any RL algorithm.

Minimax lower bound

Theorem (minimax lower bound; Azar et al., 2013)

For all $\varepsilon \in [0, \frac{1}{1-\gamma})$, there exists some MDP such that the total number of samples need to be **at least**

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

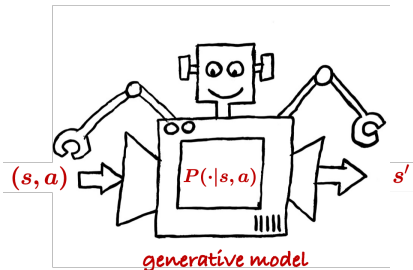
to achieve $V^* - V^{\hat{\pi}} \leq \varepsilon$, where $\hat{\pi}$ is the output of any RL algorithm.

- holds for both finding the optimal Q-function and the optimal policy over the entire range of ε
- much smaller than the model dimension $|\mathcal{S}|^2|\mathcal{A}|$

An incomplete list of works

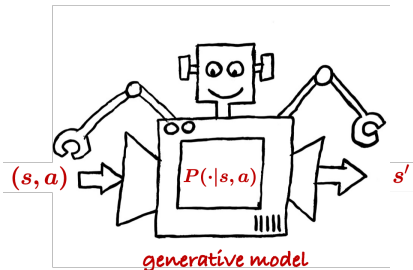
- Kearns and Singh, 1999
- Kakade, 2003
- Kearns et al., 2002
- Azar et al., 2013
- Sidford et al., 2018a, 2018b
- Wang, 2019
- Agarwal et al., 2019
- Wainwright, 2019a, 2019b
- Pananjady and Wainwright, 2019
- Yang and Wang, 2019
- Khamaru et al., 2020
- Mou et al., 2020
- Cui and Yang, 2021
- ...

Model estimation



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

Model estimation



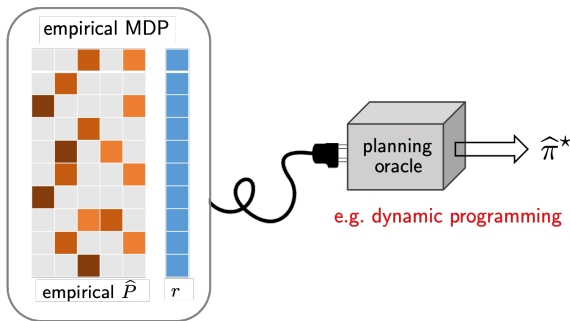
Sampling: for each (s, a) ,
collect N ind. samples
 $\{(s, a, s'_{(i)})\}_{1 \leq i \leq N}$

Empirical estimates:

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

Empirical MDP + planning

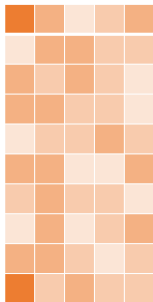
— Azar et al., 2013, Agarwal et al., 2019



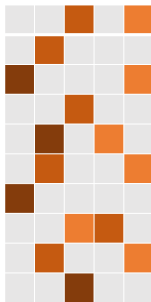
Find policy based on the empirical MDP (*empirical maximizer*)
using, e.g., policy iteration

$(\hat{P}, r)$

Challenges in the sample-starved regime



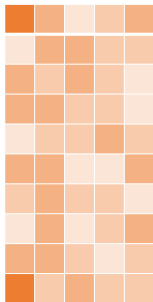
truth: $P \in \mathbb{R}^{SA \times S}$



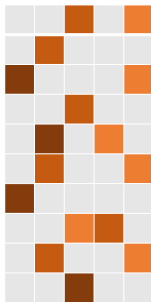
empirical estimate:
 $\hat{P}$

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

Challenges in the sample-starved regime



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



empirical estimate:
 $\hat{P}$

- Can't recover P faithfully if sample size $\ll S^2 A$!
- 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 $\hat{\pi}^*$ of empirical MDP achieves

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

with high prob., with sample complexity at most

$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

ℓ_∞ -based sample complexity

Theorem (Agarwal, Kakade, Yang '19)

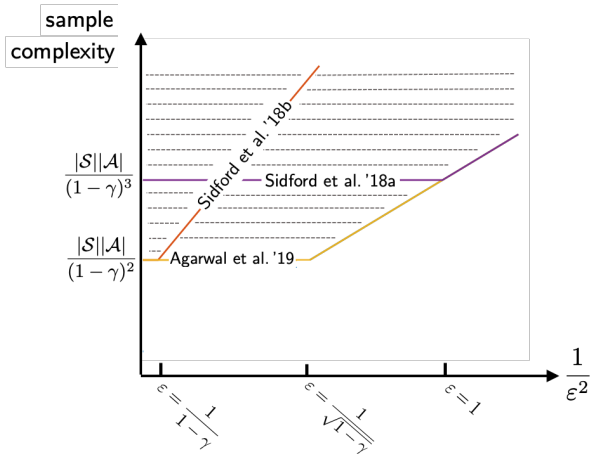
For any $0 < \varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$, the optimal policy $\hat{\pi}^*$ of empirical MDP achieves

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

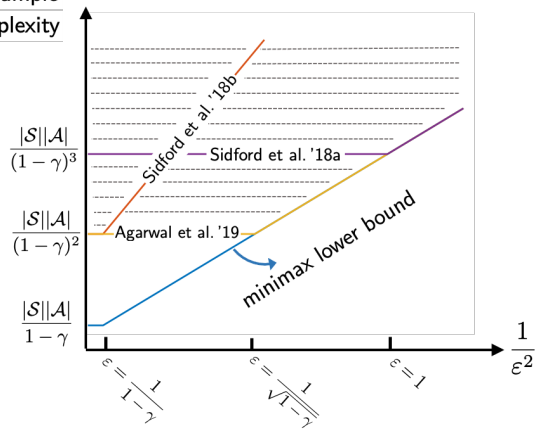
with high prob., with sample complexity at most

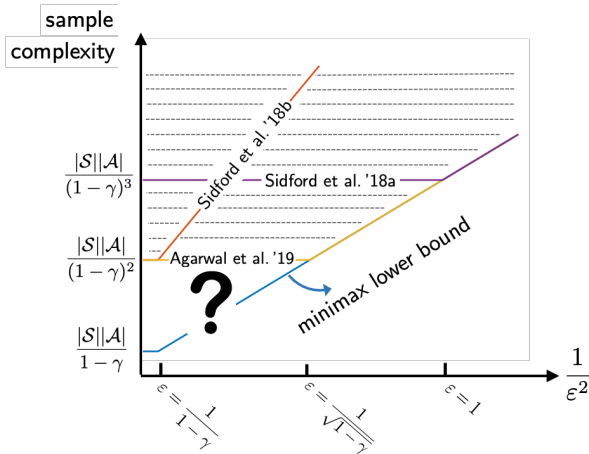
$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

- matches minimax lower bound: $\tilde{\Omega}(\frac{SA}{(1-\gamma)^3\varepsilon^2})$ when $\varepsilon \leq \frac{1}{\sqrt{1-\gamma}}$
(equivalently, when sample size exceeds $\frac{SA}{(1-\gamma)^2}$) [Azar et al., 2013](#)

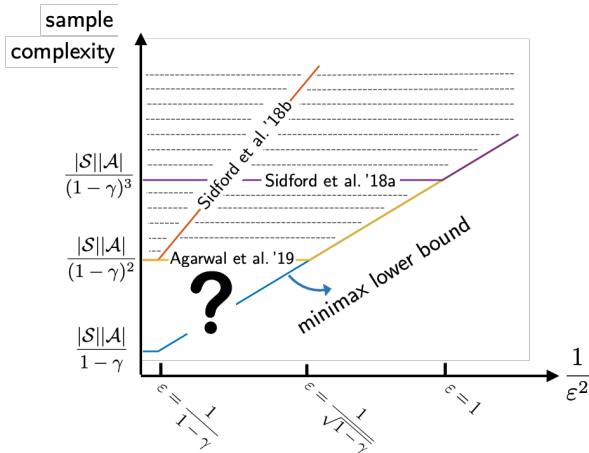


sample
complexity





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

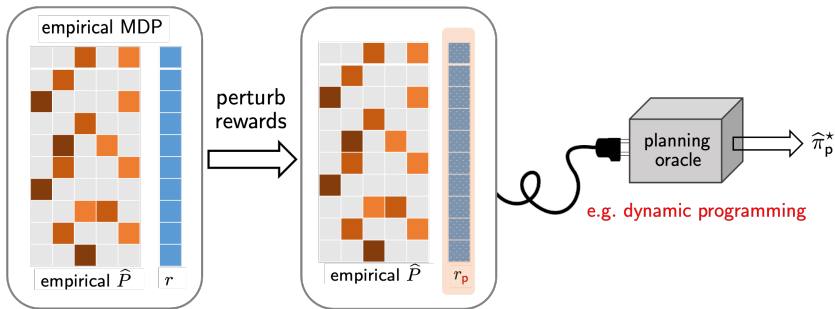


Agarwal et al., 2019 still requires a burn-in sample size $\gtrsim \frac{SA}{(1-\gamma)^2}$

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

Perturbed model-based approach (Li et al. '24)

— Li, Wei, Chi, Chen, 2024



Find policy based on **empirical** MDP w/ **slightly perturbed** rewards

Optimal ℓ_∞ -based sample complexity

Theorem (Li, Wei, Chi, Chen '20; OR '24)

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

$$\|V^{\hat{\pi}_p^*} - V^*\|_\infty \leq \varepsilon$$

with high prob., with sample complexity at most

$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

Optimal ℓ_∞ -based sample complexity

Theorem (Li, Wei, Chi, Chen '20; OR '24)

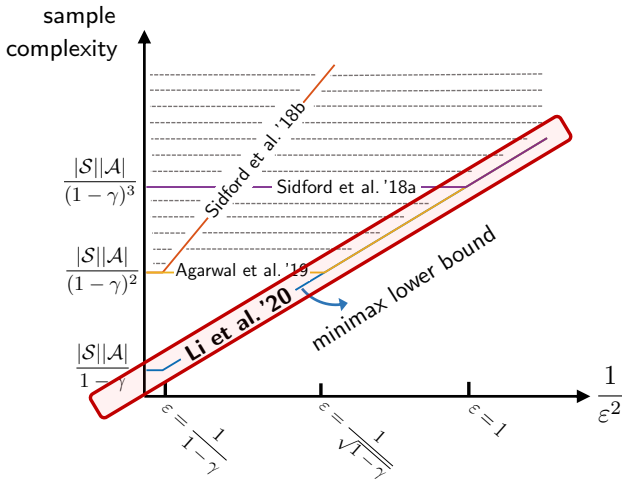
For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, the optimal policy $\hat{\pi}_p^*$ of perturbed empirical MDP achieves

$$\|V^{\hat{\pi}_p^*} - V^*\|_\infty \leq \varepsilon$$

with high prob., with sample complexity at most

$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

- matches minimax lower bound: $\tilde{\Omega}(\frac{SA}{(1-\gamma)^3\varepsilon^2})$ [Azar et al., 2013](#)
- full ε -range: $\varepsilon \in (0, \frac{1}{1-\gamma}] \longrightarrow$ no burn-in cost

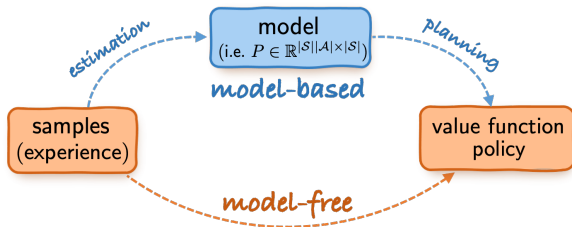


Model based RL is minimax optimal under generative models and does NOT suffer from a sample size barrier

Part 1

1. Basics: Markov decision processes
2. RL w/ a generative model (simulator)
 - model-based algorithms (a “plug-in” approach)
 - model-free algorithms

Model-based vs. model-free RL

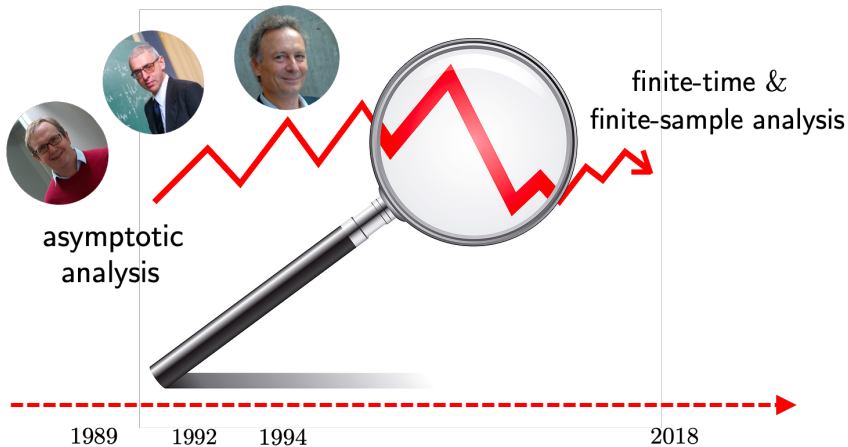


Model-based approach (“plug-in”)

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

Model-free / value-based approach

- learning w/o modeling & estimating environment explicitly
- memory-efficient, online, ...



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

A starting point: Bellman optimality principle

Bellman operator

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

- one-step look-ahead

A starting point: Bellman optimality principle

Bellman operator

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

- one-step look-ahead

Bellman equation: Q^* is *unique* solution to

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

A starting point: Bellman optimality principle

Bellman operator

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

- one-step look-ahead

Bellman equation: Q^* is *unique* solution to

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

- **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

Stochastic approximation for solving the **Bellman equation**

Robbins & Monro, 1951

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

where

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

Q-learning: a stochastic approximation algorithm



Chris Watkins



Peter Dayan

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

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

Q-learning: a stochastic approximation algorithm



Chris Watkins



Peter Dayan

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

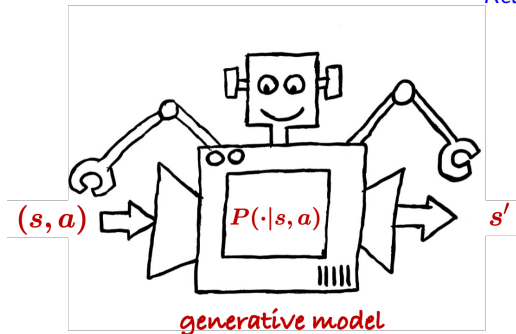
$$\underbrace{Q_{t+1}(s, a) = Q_t(s, a) + \eta_t(\mathcal{T}_t(Q_t)(s, a) - Q_t(s, a))}_{\text{sample transition } (s, a, s')}, \quad t \geq 0$$

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

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

A generative model / simulator

— *Kearns, Singh, 1999*



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

Synchronous Q-learning



Chris Watkins



Peter Dayan

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

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

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

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

synchronous: all state-action pairs are updated simultaneously

- total sample size: TSA

Sample complexity of synchronous Q-learning

Theorem (Li, Cai, Chen, Wei, Chi'21, OR'24)

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

$$\begin{cases} \tilde{O}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right) & \text{if } A \geq 2 \\ \tilde{O}\left(\frac{S}{(1-\gamma)^3\varepsilon^2}\right) & \text{if } A = 1 \end{cases} \quad (\text{TD learning})$$

Sample complexity of synchronous Q-learning

Theorem (Li, Cai, Chen, Wei, Chi'21, OR'24)

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

$$\begin{cases} \tilde{O}\left(\frac{SA}{(1-\gamma)^4 \varepsilon^2}\right) & \text{if } A \geq 2 \\ \tilde{O}\left(\frac{S}{(1-\gamma)^3 \varepsilon^2}\right) & \text{if } A = 1 \end{cases} \quad (\text{TD learning})$$

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

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

Sample complexity of synchronous Q-learning

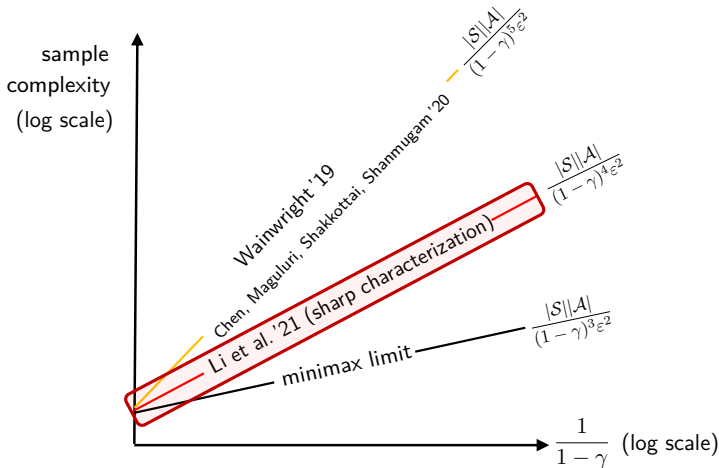
Theorem (Li, Cai, Chen, Wei, Chi'21, OR'24)

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

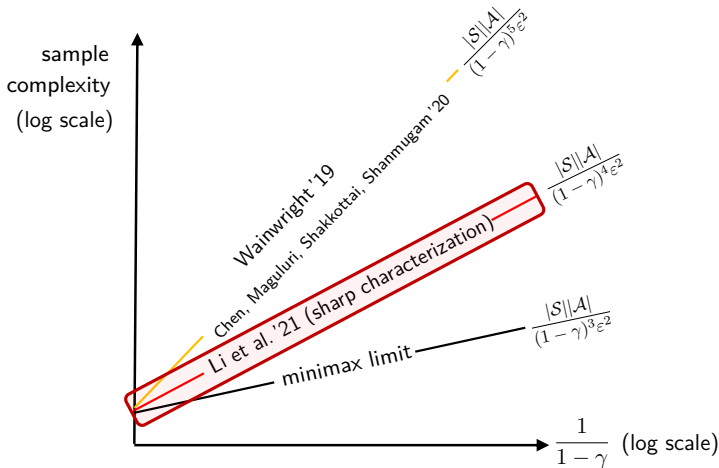
$$\begin{cases} \tilde{O}\left(\frac{SA}{(1-\gamma)^4 \varepsilon^2}\right) & \text{if } A \geq 2 & (?) \\ \tilde{O}\left(\frac{S}{(1-\gamma)^3 \varepsilon^2}\right) & \text{if } A = 1 & (\text{minimax optimal}) \end{cases}$$

other papers	sample complexity
Even-Dar & Mansour, 2003	$2^{\frac{1}{1-\gamma}} \frac{SA}{(1-\gamma)^4 \varepsilon^2}$
Beck, Srikant, 2012	$\frac{S^2 A^2}{(1-\gamma)^5 \varepsilon^2}$
Wainwright, 2019	$\frac{SA}{(1-\gamma)^5 \varepsilon^2}$
Chen, Maguluri, Shakkottai, Shanmugam, 2020	$\frac{SA}{(1-\gamma)^5 \varepsilon^2}$

All this requires sample size at least $\frac{SA}{(1-\gamma)^4 \epsilon^2}$ ($A \geq 2$) ...



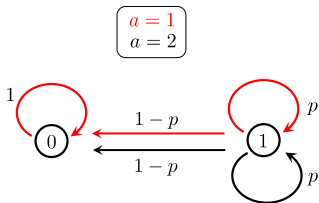
All this requires sample size at least $\frac{SA}{(1-\gamma)^4 \epsilon^2}$ ($A \geq 2$) ...



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

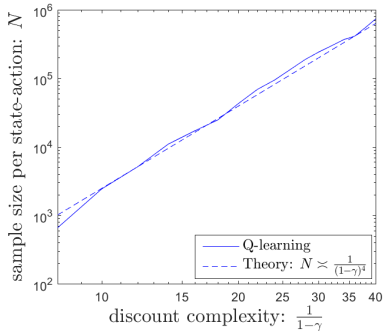
A numerical example: $\frac{SA}{(1-\gamma)^4 \varepsilon^2}$ samples seem necessary ...

— *observed in Wainwright '19*



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

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



Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi '21, OR'24)

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

$$\tilde{\Omega}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right) \text{ samples}$$

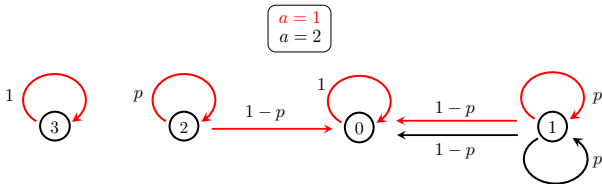
Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi '21, OR'24)

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

$$\tilde{\Omega}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right) \text{ samples}$$

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

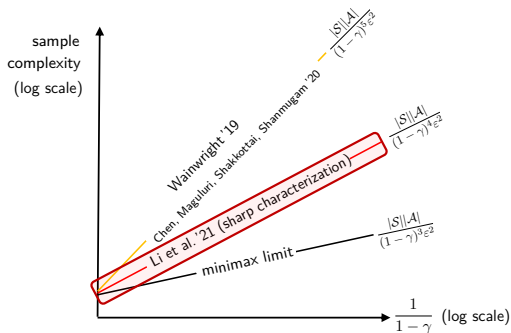


Q-learning is NOT minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi '21, OR'24)

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

$$\tilde{\Omega}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right) \text{ samples}$$



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)

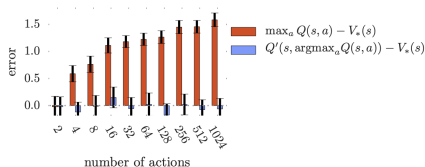


Figure 1: The orange bars show the bias in a single Q-learning update when the action values are $Q(s, a) = V_*(s) + \epsilon_a$ and the errors $\{\epsilon_a\}_{a=1}^m$ are independent standard normal random variables. The second set of action values Q' , used for the blue bars, was generated identically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

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)

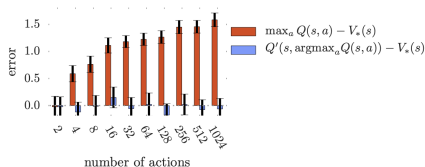


Figure 1: The orange bars show the bias in a single Q-learning update when the action values are $Q(s, a) = V_*(s) + \epsilon_a$ and the errors $\{\epsilon_a\}_{a=1}^m$ are independent standard normal random variables. The second set of action values Q' , used for the blue bars, was generated identically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

A provable improvement: Q-learning with variance reduction

(Wainwright 2019)

*Improving sample complexity via **variance reduction***

— a powerful idea from finite-sum stochastic optimization

Variance-reduced Q-learning updates (Wainwright, 2019)

— inspired by SVRG (Johnson & Zhang, 2013)

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

Variance-reduced Q-learning updates (Wainwright, 2019)

— inspired by SVRG (Johnson & Zhang, 2013)

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

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

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

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

An epoch-based stochastic algorithm

— inspired by [Johnson & Zhang, 2013](#)

update $\bar{Q}$ variance-reduced
 Q-learning



for each epoch

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

Sample complexity of variance-reduced Q-learning

Theorem (Wainwright '19)

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

$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

- allows for more aggressive learning rates

Sample complexity of variance-reduced Q-learning

Theorem (Wainwright '19)

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

$$\tilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$$

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

Reference: general RL textbooks I

- "*Reinforcement learning: An introduction*," R. S. Sutton, A. G. Barto, MIT Press, 2018
- "*Reinforcement learning: Theory and algorithms*," A. Agarwal, N. Jiang, S. Kakade, W. Sun, 2019
- "*Reinforcement learning and optimal control*," D. Bertsekas, Athena Scientific, 2019
- "*Algorithms for reinforcement learning*," C. Szepesvari, Springer, 2022
- "*Bandit algorithms*," T. Lattimore, C. Szepesvari, Cambridge University Press, 2020

Reference: model-based algorithms I

- “*Finite-sample convergence rates for Q-learning and indirect algorithms*,” M. Kearns, S. Satinder, *NeurIPS*, 1998
- “*On the sample complexity of reinforcement learning*,” S. Kakade, 2003
- “*A sparse sampling algorithm for near-optimal planning in large Markov decision processes*,” M. Kearns, Y. Mansour, A. Y. Ng, *Machine learning*, 2002
- “*Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model*,” M. G. Azar, R. Munos, H. J. Kappen, *Machine learning*, 2013
- “*Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time*,” *Mathematics of Operations Research*, 2020
- “*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

Reference: model-based algorithms II

- “Variance reduced value iteration and faster algorithms for solving Markov decision processes,” A. Sidford, M. Wang, X. Wu, Y. Ye, *SODA*, 2018
- “Model-based reinforcement learning with a generative model is minimax optimal,” A. Agarwal, S. Kakade, L. Yang, *COLT*, 2020
- “Instance-dependent ℓ_∞ -bounds for policy evaluation in tabular reinforcement learning,” A. Pananjady, M. J. Wainwright, *IEEE Trans. on Information Theory*, 2020
- “Spectral methods for data science: A statistical perspective,” Y. Chen, Y. Chi, J. Fan, C. Ma, *Foundations and Trends® in Machine Learning*, 2021
- “Breaking the sample size barrier in model-based reinforcement learning with a generative model,” G. Li, Y. Wei, Y. Chi, Y. Chen, *Operations Research*, 2024

Reference: model-free algorithms 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
- "Q-learning," C. Watkins, P. Dayan, *Machine Learning*, 1992
- "Learning rates for Q-learning," E. Even-Dar, Y. Mansour, *Journal of Machine Learning Research*, 2003
- "The asymptotic convergence-rate of Q-learning," C. Szepesvari, *NeurIPS*, 1998
- "Error bounds for constant step-size Q-learning," C. Beck, R. Srikant, *Systems & Control Letters*, 2012
- "Stochastic approximation with cone-contractive operators: Sharp ℓ_∞ bounds for Q-learning," M. Wainwright, 2019

Reference: model-free algorithms II

- “*Is Q-learning minimax optimal? a tight sample complexity analysis*,” G. Li, C. Cai, Y. Chen, Y. Wei, Y. Chi, *Operations Research*, 2024
- “*Variance-reduced Q-learning is minimax optimal*,” M. Wainwright, 2019
- “*Sample-optimal parametric Q-learning using linearly additive features*,” L. Yang, M. Wang, *ICML*, 2019
- “*Asynchronous stochastic approximation and Q-learning*,” J. Tsitsiklis, *Machine learning*, 1994
- “*Finite-time analysis of asynchronous stochastic approximation and Q-learning*,” G. Qu, A. Wierman, *COLT*, 2020
- “*Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes*,” Z. Chen, S. T. Maguluri, S. Shakkottai, K. Shanmugam, *NeurIPS*, 2020
- “*Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction*,” G. Li, Y. Wei, Y. Chi, Y. Gu, Y. Chen, *IEEE Trans. on Information Theory*, 2022