Reinforcement Learning meets Federated Learning and Distributional Robustness

Yuejie Chi

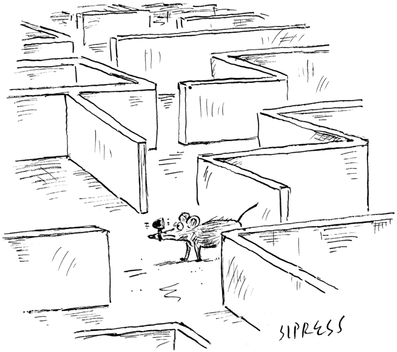
Carnegie Mellon University

Seminar Series: Women in Data Science and Mathematics
July 2023

Reinforcement learning (RL)

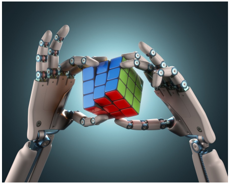
In RL, an agent learns by interacting with an environment.

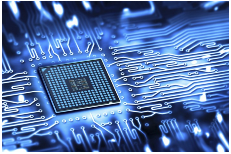
- unknown environments
- maximize total rewards
- trial-and-error
- sequential and online

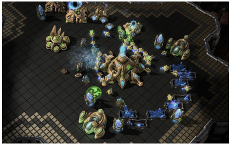


"Recalculating ... recalculating ..."

Recent successes in RL







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

Sample efficiency

Collecting data samples might be expensive or time-consuming due to the enormous state and action space

clinical trials

autonomous driving

online ads

Sample efficiency

Collecting data samples might be expensive or time-consuming due to the enormous state and action space

clinical trials

autonomous driving

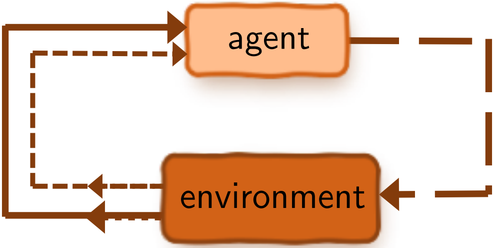
online ads

Calls for design of sample-efficient RL algorithms!

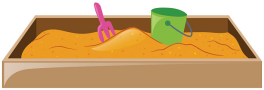
Statistical thinking in RL: non-asymptotic analysis

Non-asymptotic analyses are key to understand statistical efficiency in modern RL.

Recent advances in statistical RL

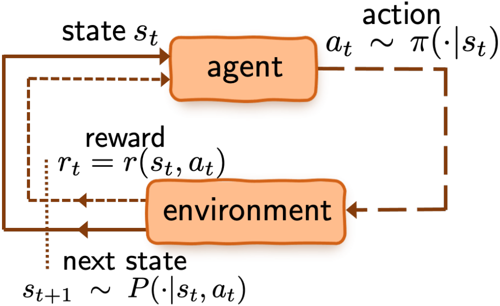


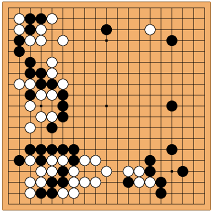
The playground: Markov decision processes



Backgrounds: Markov decision processes

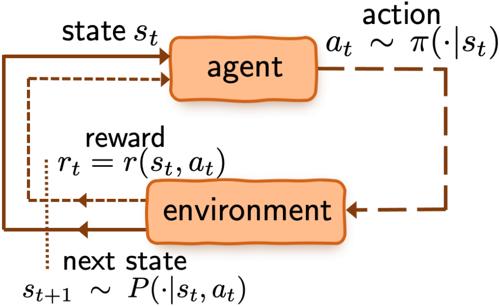
Markov decision process (MDP)

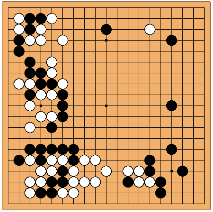




- $\mathcal{S}$: state space
- $\mathcal{A}$: action space

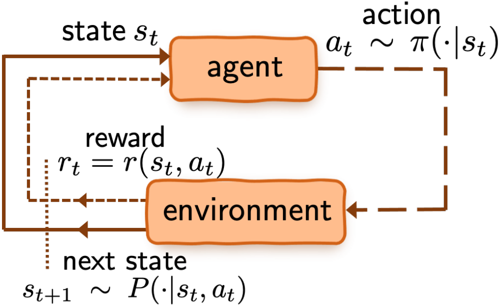
Markov decision process (MDP)

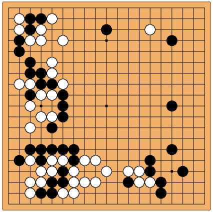




- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward

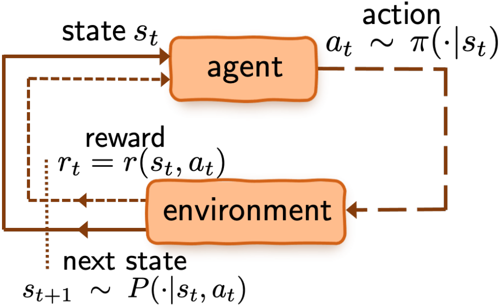
Markov decision process (MDP)

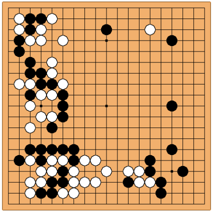




- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)

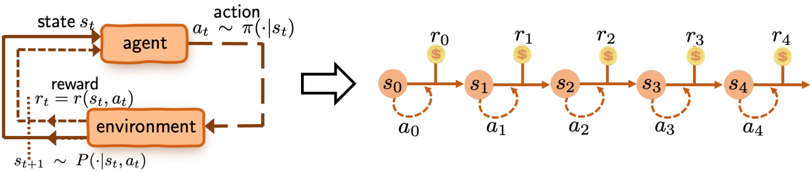
Markov decision process (MDP)





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

Value function



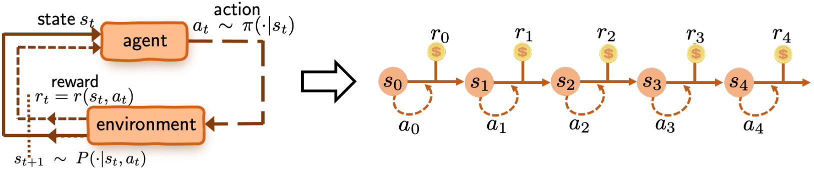
Value function of policy π:

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

Q-function of policy π:

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

Value function



Value function of policy π:

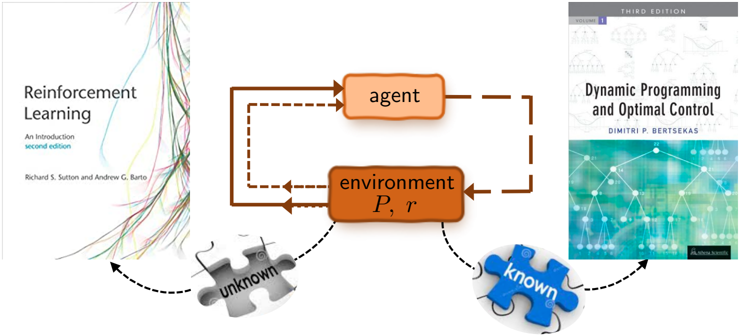
$$\forall s \in \mathcal{S}: \qquad V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right]$$

Q-function of policy π:

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

- $\gamma \in [0,1)$ is the discount factor; $\frac{1}{1-\gamma}$ is effective horizon
- Expectation is w.r.t. the sampled trajectory under π

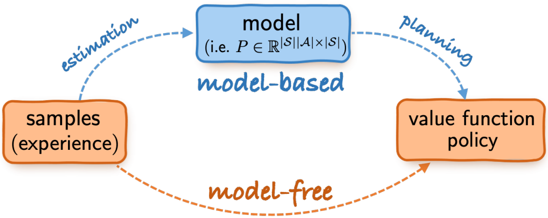
Searching for the optimal policy



Goal: find the optimal policy $\pi^\star$ that maximize $V^\pi(s)$

- optimal value / Q function: $V^\star := V^{\pi^\star}, \ Q^\star := Q^{\pi^\star}$
- optimal policy $\pi^\star(s) = \mathrm{argmax}_{a \in \mathcal{A}} Q^\star(s, a)$

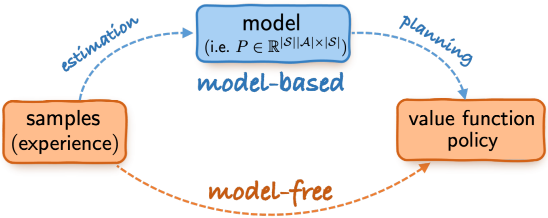
Two approaches to RL



Model-based approach ("plug-in")

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

Two approaches to RL



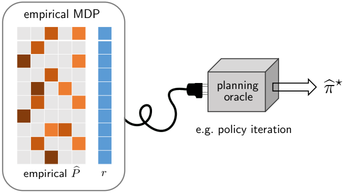
Model-based approach ("plug-in")

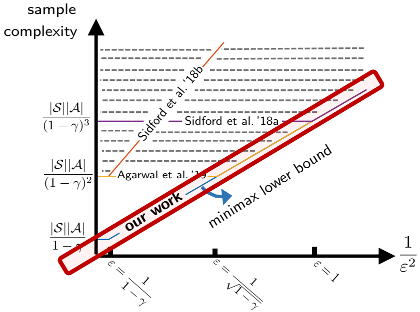
1. build an empirical estimate $\widehat{P}$ for P
2. planning based on empirical $\widehat{P}$

Model-free approach

1. learning w/o constructing model explicitly
2. memory-efficient

Recent advances in model-based RL

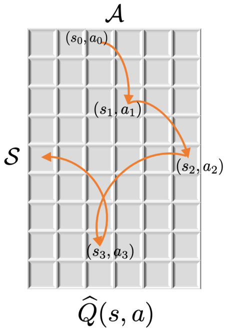


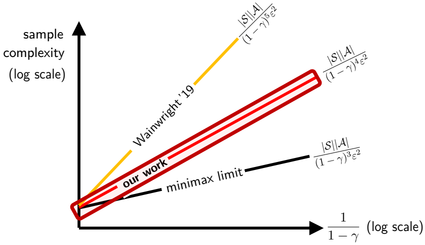


Plug-in estimators are minimax-optimal

(Sidford et al., 2018; Agarwal et al., 2019; Wang 2019; Li et al., 2020)

Recent advances in model-free RL

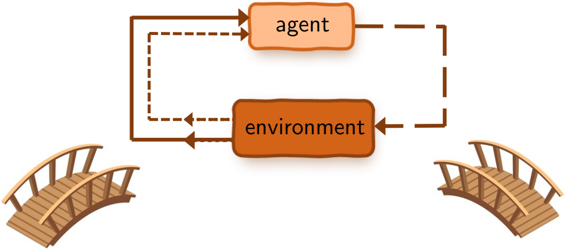




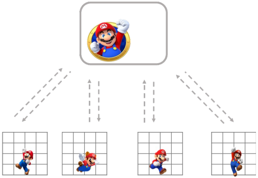
Q-learning is not minimax-optimal

(Even-Dar and Mansour, 2013; Wainwright, 2019; Chen et al., 2020; Li et al., 2021)

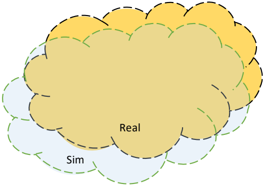
This talk: beyond standard MDP



Federated learning



Distributional robustness



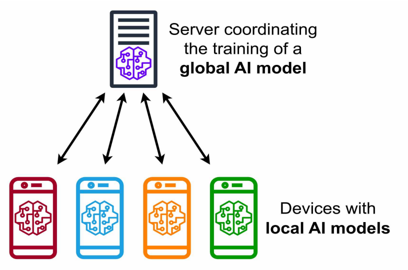
Reinforcement learning meets federated learning: linear speedup and beyond

Jiin Woo
CMU

Gauri Joshi
CMU

"The Blessing of Heterogeneity in Federated Q-Learning: Linear Speedup and Beyond," arXiv:2305.10697, short version at ICML 2023.

Federated learning




FORBES > INNOVATION > AI

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

Kevin Krewell Contributor
Tirias Research Contributor Group
Follow
Oct 15, 2021, 02:51pm EDT

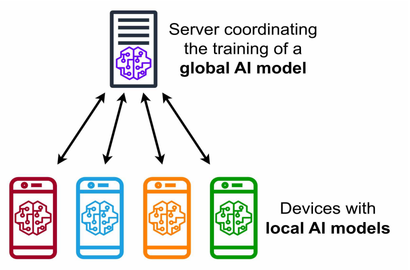

How Apple personalizes Siri without hoovering up your data

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

By Karen Hao

December 11, 2019

Federated learning




FORBES > INNOVATION > AI

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

Kevin Krewell Contributor
Tirias Research Contributor Group

Follow

Oct 15, 2021, 02:51pm EDT


How Apple personalizes Siri without hoovering up your data

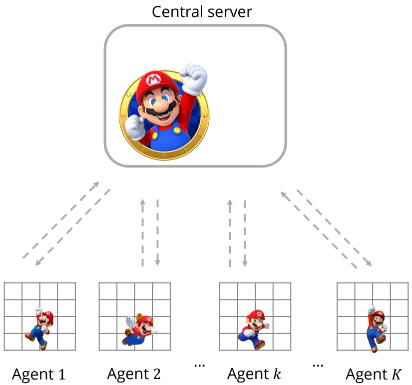
The tech giant is using privacy-preserving machine learning to improve its voice assistant while keeping your data on your phone.

By Karen Hao

December 11, 2019

Can we harness the power of federated learning for RL?

RL meets federated learning



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

Questions

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

Linear speedup:

Can we achieve linear speedup when learning with multiple agents?

Communication efficiency:

Can we perform multiple local updates to save communication?

Taming heterogeneity:

How to combine heterogeneous local updates to accelerate learning?

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

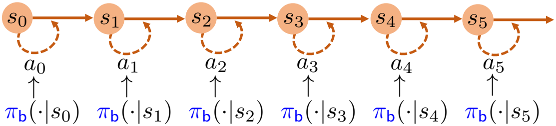
$\underbrace{\text{Stochastic approximation}}_{\text{Robbins \& Monro, 1951}}$ for solving the **Bellman equation**

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

where

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

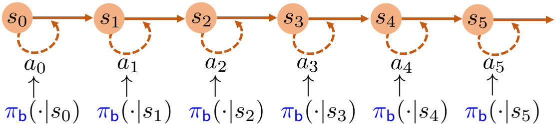
Asynchronous Q-learning



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

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

Asynchronous Q-learning



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

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

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

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

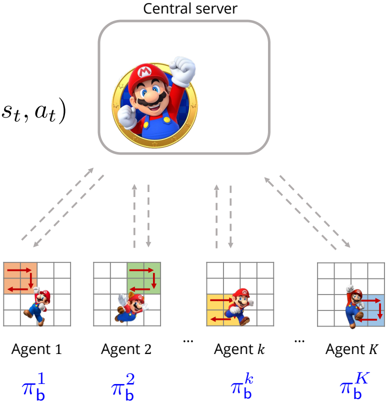
How to federate asynchronous Q-learning?

Federated asynchronous Q-learning with local updates

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

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

and sends it to the server.



Federated asynchronous Q-learning with local updates

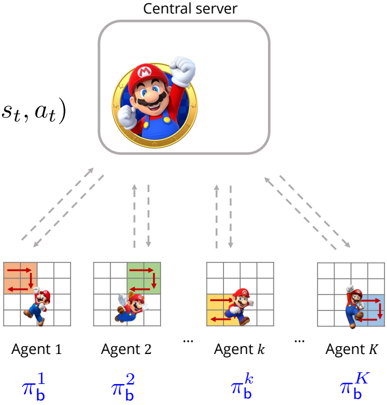
- **The agent** k performs τ rounds of local Q-learning updates:

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

  and sends it to the server.

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

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



Federated asynchronous Q-learning with local updates

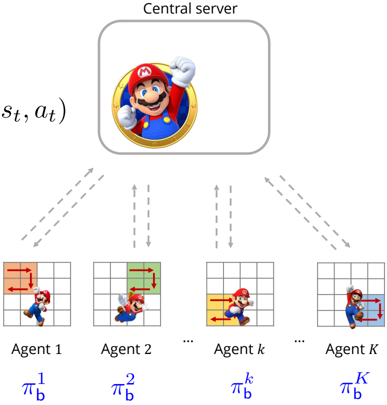
- **The agent** k performs τ rounds of local Q-learning updates:

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

and sends it to the server.

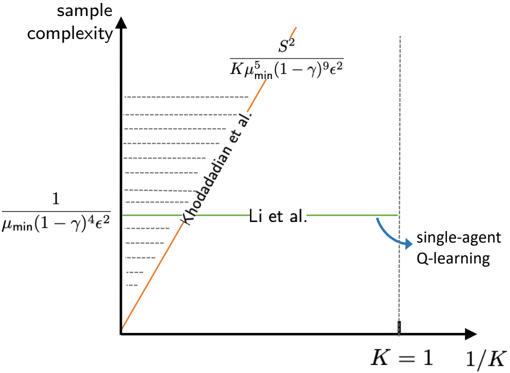
- **The server** averages the local updates and communicates it back to agents:

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



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

Prior art

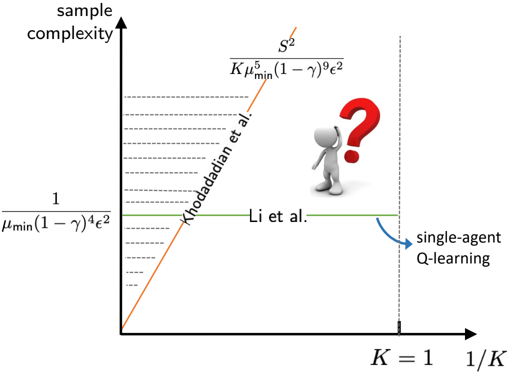


Key quantity: minimum state-action occupancy probability

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

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

Prior art



Key quantity: minimum state-action occupancy probability

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

Can we improve the dependency on the salient parameters?

Our theorem

Theorem (Jiin, Joshi, Chi, ICML 2023)

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

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

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

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

Our theorem

Theorem (Jiin, Joshi, Chi, ICML 2023)

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

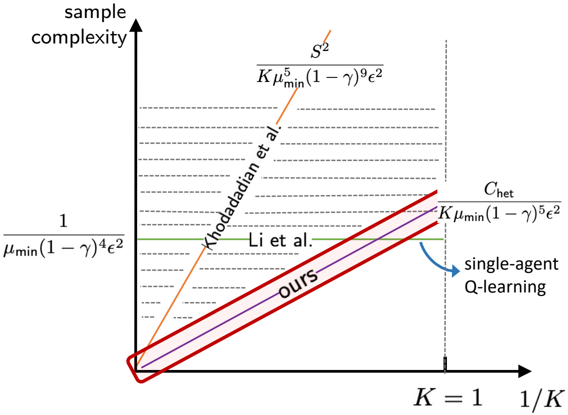
$$\widetilde{O}\left(\frac{C_{\mathsf{het}}}{K \mu_{\mathsf{min}} (1-\gamma)^5 \epsilon^2}\right)$$

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

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

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

Comparison with prior art



Linear speedup with near-optimal parameter dependencies!

Benefit of heterogeneity?

- **Curse of heterogeneity?** performance degenerates when local behavior policies are heterogeneous (i.e. $C_{\text{het}} \gg 1$).

Benefit of heterogeneity?

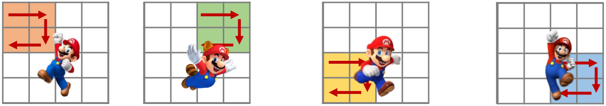
- **Curse of heterogeneity?** performance degenerates when local behavior policies are heterogeneous (i.e. $C_{\text{het}} \gg 1$).
- **Full coverage:** require full coverage of every agent over the entire state-action space (i.e. $\mu_{\text{min}} > 0$).

Benefit of heterogeneity?

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

Benefit of heterogeneity?

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

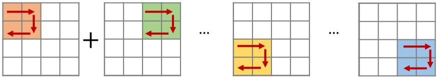


Agent 1 Agent 2 … Agent k … Agent K

Is it possible to alleviate these requirements?

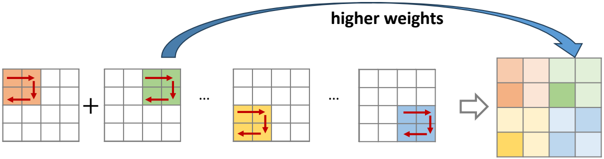
Importance averaging

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



Importance averaging

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



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

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

where

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

Our theorem

Theorem (Jiin, Joshi, Chi, ICML 2023)

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

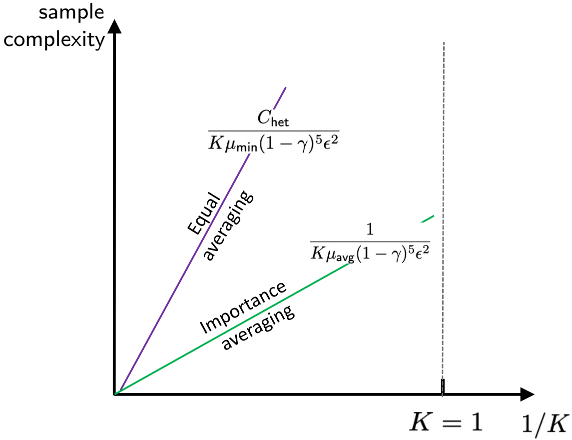
$$\widetilde{O}\left(\frac{1}{K\,\mu_{\mathsf{avg}}(1-\gamma)^5\epsilon^2}\right)$$

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

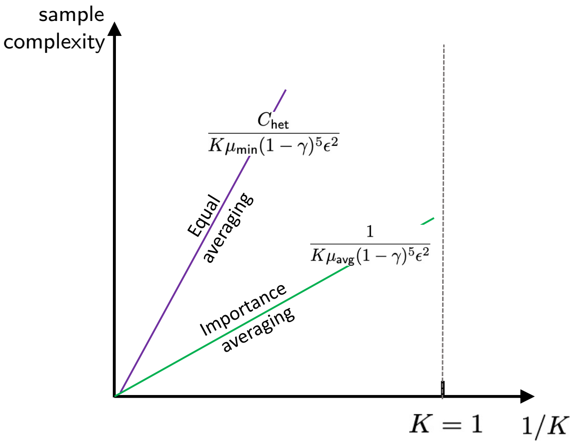
$$\mu_{\mathsf{avg}} = \min_{s,a} \frac{1}{K}\sum_{k=1}^{K} \mu_{\mathsf{b}}^k(s,a) \ge \mu_{\mathsf{min}}.$$

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

Equal averaging versus importance averaging

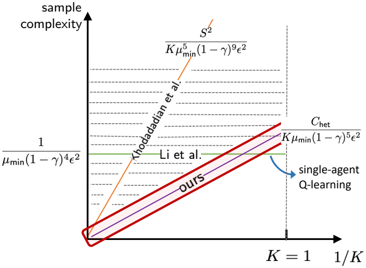


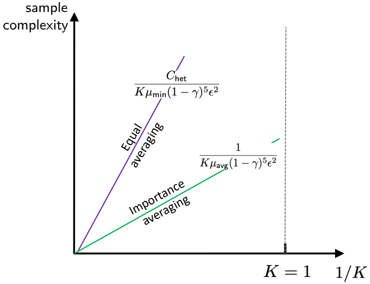
Equal averaging versus importance averaging



Importance averaging does not require full coverage of individual agents!

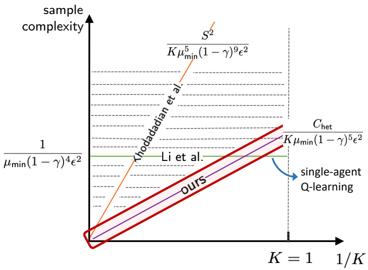
Summary

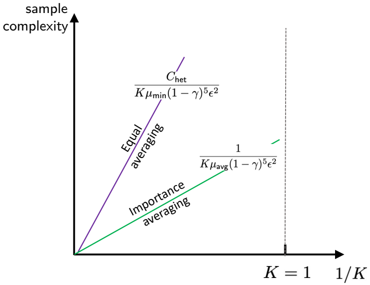




Provable benefits of federated Q-learning: near-optimal linear speedup!

Summary





Provable benefits of federated Q-learning: near-optimal linear speedup!

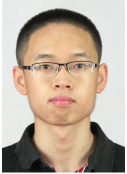
Ongoing and future work:

- Other problems in RL such as policy evaluation and offline RL.
- Multi-task RL: heterogeneous environments across agents.

RL meets distributional robustness: towards minimax-optimal sample complexity

Laixi Shi

Caltech



Gen Li

UPenn

Yuxin Chen

UPenn

Yuting Wei

UPenn

Matthieu Geist

Google

"The Curious Price of Distributional Robustness in Reinforcement Learning with a Generative Model," arXiv:2305.16589.

Safety and robustness in RL

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

Training environment

$\neq$

Test environment

Safety and robustness in RL

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

Training environment

$\neq$

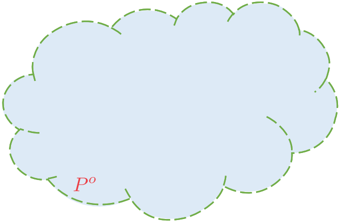
Test environment

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

Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

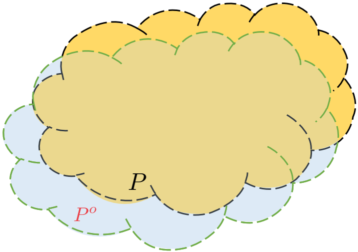
$$\mathcal{U}^\sigma(P^o) = \left\{P : \rho\big(P, P^o\big) \le \sigma\right\}$$

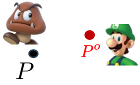


Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

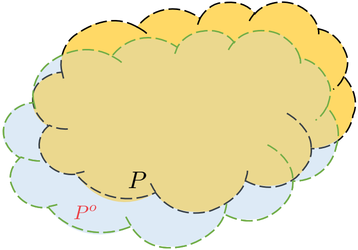
$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \le \sigma\}$$

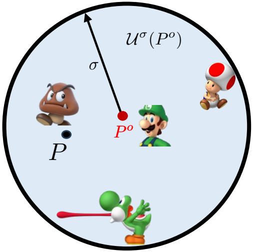




Uncertainty set of the nominal transition kernel P^o:

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

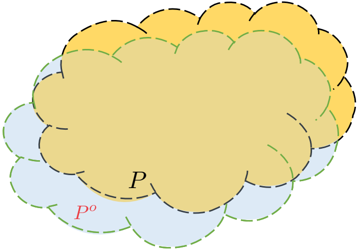


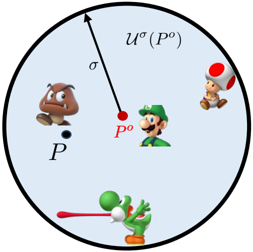


Modeling environment uncertainty

Uncertainty set of the nominal transition kernel P^o:

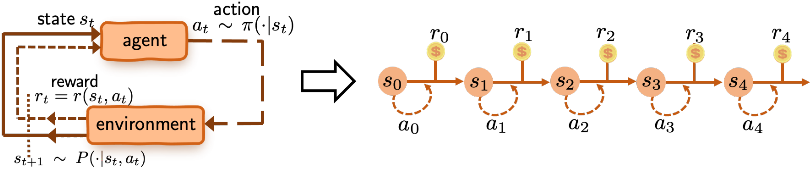
$$\mathcal{U}^\sigma(P^o) = \{P : \rho(P, P^o) \leq \sigma\}$$





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

Robust value/Q function



Robust value/Q function of policy π:

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

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

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

Distributionally robust MDP

Robust MDP

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

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

Distributionally robust MDP

Robust MDP

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

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

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

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

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

Distributionally robust MDP

Robust MDP

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

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

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

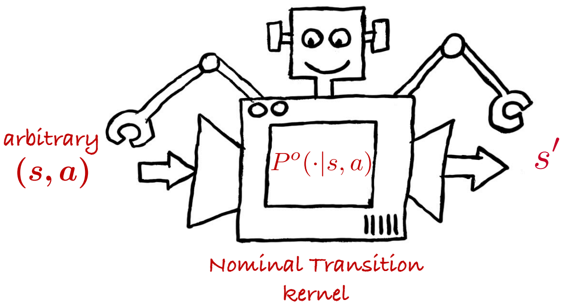
$$Q^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{P_{s,a} \in \mathcal{U}^\sigma(P^o_{s,a})} \left\langle P_{s,a}, V^{\star,\sigma} \right\rangle ,$$
$$V^{\star,\sigma}(s) = \max_a Q^{\star,\sigma}(s,a)$$

Distributionally robust value iteration (DRVI):

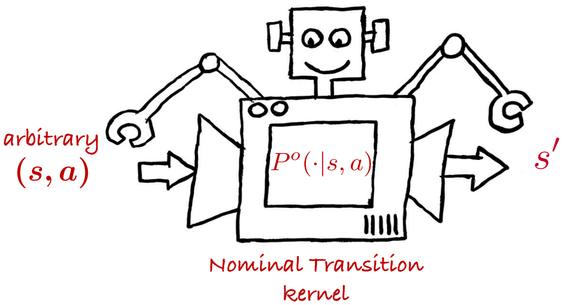
$$Q(s,a) \leftarrow r(s,a) + \gamma \inf_{P_{s,a} \in \mathcal{U}^\sigma(P^o_{s,a})} \langle P_{s,a}, V \rangle,$$

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

Learning distributionally robust MDPs



Learning distributionally robust MDPs

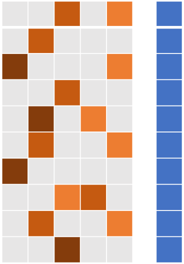


Goal of robust RL: given $\mathcal{D} := \{(s_i, a_i, s_i')\}_{i=1}^N$ from the *nominal* environment P^0, find an ϵ-optimal robust policy $\widehat{\pi}$ obeying

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

— *in a sample-efficient manner*

A curious question

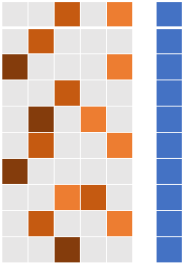


empirical MDP

Learn the optimal policy of the nominal MDP?

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

A curious question



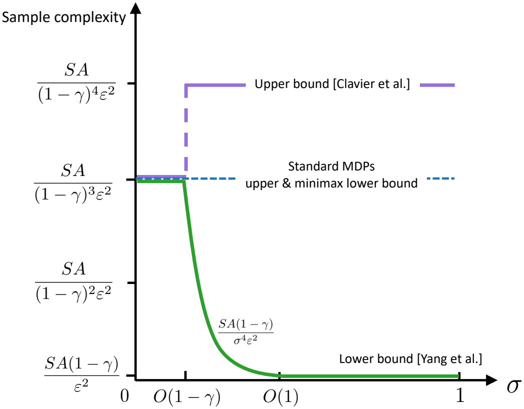
empirical MDP

Learn the optimal policy of the nominal MDP?

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

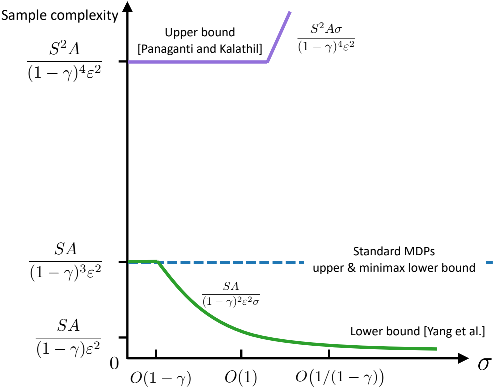
Robustness-statistical trade-off? Is there a statistical premium that one needs to pay in quest of additional robustness?

Prior art: TV uncertainty



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

Prior art: χ^2 uncertainty



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

Our theorem under TV uncertainty

Theorem (Shi et al., 2023)

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

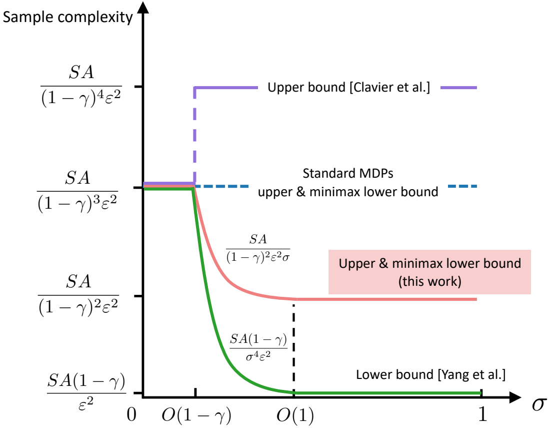
$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^2 \max\{1-\gamma, \sigma\}\epsilon^2}\right)$$

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

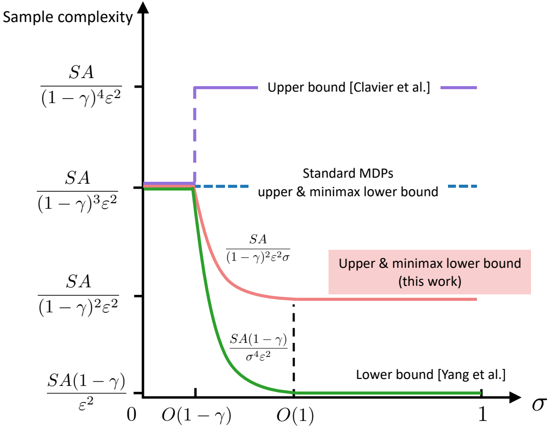
$$\widetilde{\Omega}\left(\frac{SA}{(1-\gamma)^2 \max\{1-\gamma, \sigma\}\epsilon^2}\right).$$

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

When the uncertainty set is TV



When the uncertainty set is TV



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

Our theorem under χ^2 uncertainty

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

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

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

ignoring logarithmic factors.

Our theorem under χ^2 uncertainty

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

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

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

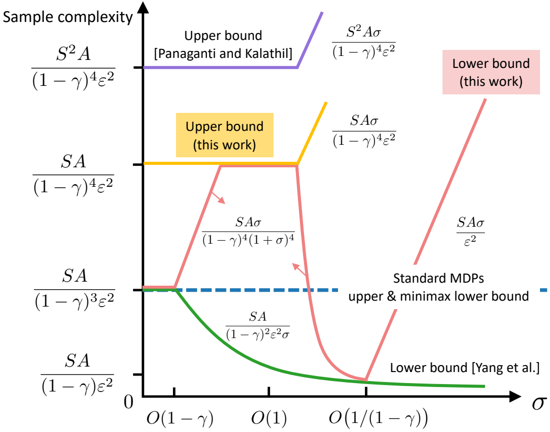
ignoring logarithmic factors.

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

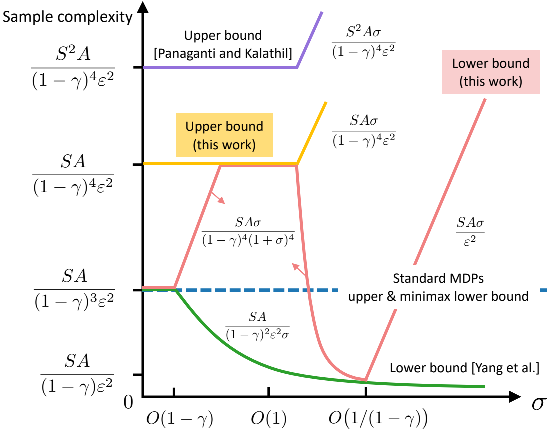
In addition, no algorithm succeeds when the sample size is below

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

When the uncertainty set is χ^2 divergence

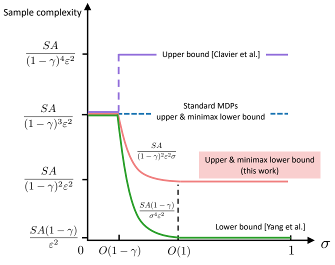


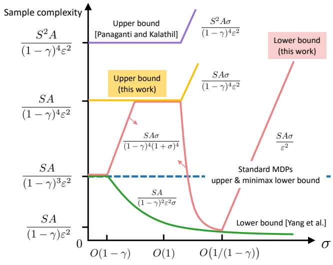
When the uncertainty set is χ^2 divergence



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

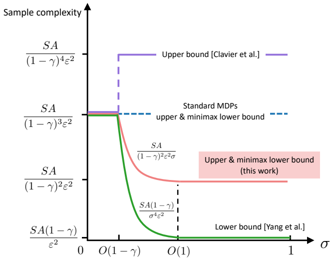
Summary

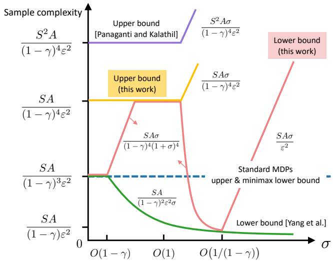




The price of robustness varies: choice of the uncertainty set matters.

Summary





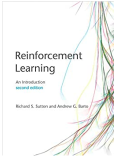
The price of robustness varies: choice of the uncertainty set matters.

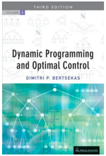
Ongoing and future work:

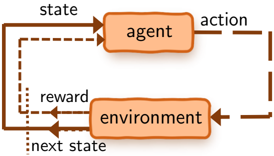
- Other choices of uncertainty sets: KL divergence.
- Function approximation.

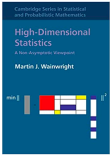
Concluding remarks

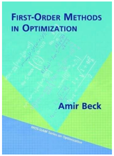
Concluding remarks











Understanding non-asymptotic performances of RL algorithms sheds light to their empirical successes (and failures)!

Thanks!

- The Blessing of Heterogeneity in Federated Q-Learning: Linear Speedup and Beyond, arXiv: 2305.10697. Short version at ICML 2023.
- The Curious Price of Distributional Robustness in Reinforcement Learning with a Generative Model, arXiv:2305.16589.

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