From Single-agent to Federated Reinforcement Learning

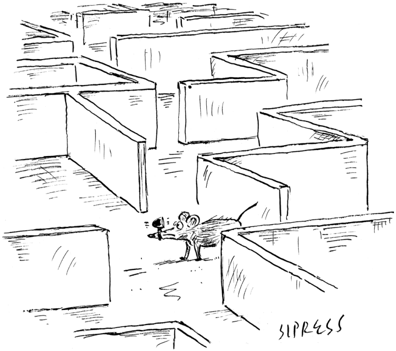
Yuejie Chi

Carnegie Mellon University

University of Chicago
February 2024

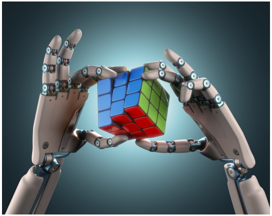
Reinforcement learning (RL)

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



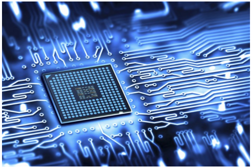
"Recalculating ... recalculating ..."

More successes of RL since AlphaGo

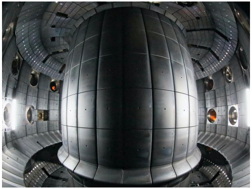


robotics

strategic games



chip designs

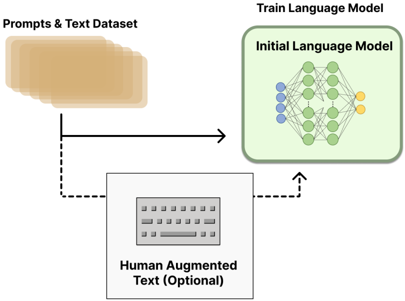


nuclear plant control

resource management

UAV and drones

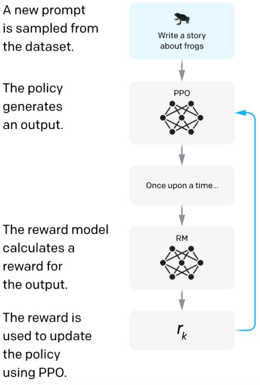
One more recent success: RLHF



ChatGPT

Bard

Optimize a policy against the reward model using reinforcement learning.



You

What is RLHF?

ChatGPT

RLHF stands for Reinforcement Learning from Human Feedback. It's a technique used in machine learning and artificial intelligence where a model learns to perform tasks or make decisions based on feedback from human trainers, rather than solely relying on pre-existing data sets or explicit programming. This approach allows the

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

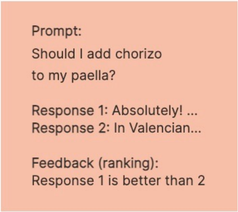
RL holds great promise in accelerating scientific, engineering and societal discoveries.

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

Sample efficiency

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

clinical trials



Prompt:
Should I add chorizo
to my paella?

Response 1: Absolutely! ...
Response 2: In Valencian...

Feedback (ranking):
Response 1 is better than 2

LLM alignment

autonomous driving

Sample efficiency

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

clinical trials

Prompt:
Should I add chorizo to my paella?

Response 1: Absolutely! ...
Response 2: In Valencian...

Feedback (ranking):
Response 1 is better than 2

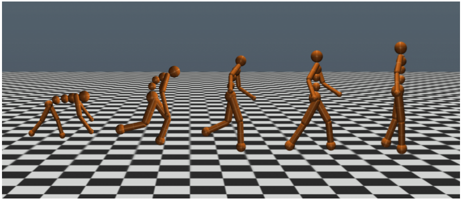
LLM alignment

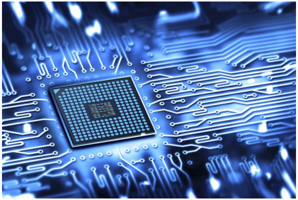
autonomous driving

Calls for design of sample-efficient RL algorithms!

Computational efficiency

Training RL algorithms might take a long time

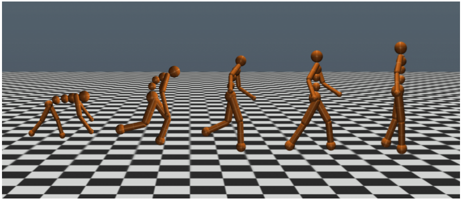


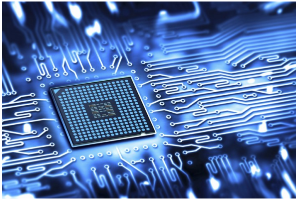


many CPUs / GPUs / TPUs + computing hours

Computational efficiency

Training RL algorithms might take a long time





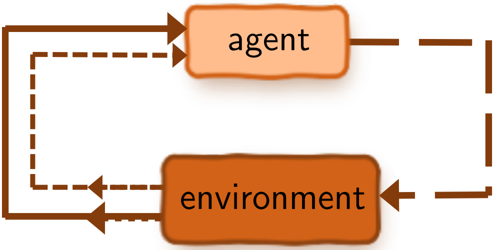
many CPUs / GPUs / TPUs + computing hours

Calls for runtime efficient RL algorithms!

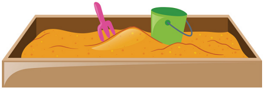
Statistical thinking in RL: non-asymptotic analysis

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

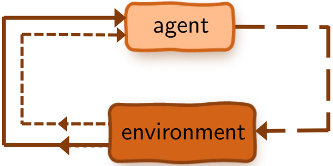
Recent advances in statistical RL



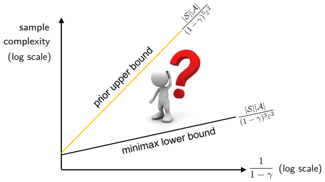
The playground: Markov decision processes



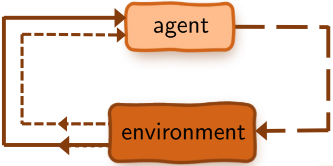
This talk: from single-agent to federated Q-learning



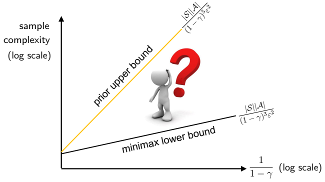
Single-agent Q-learning



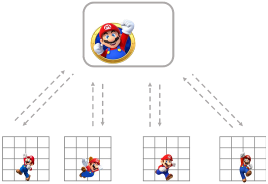
This talk: from single-agent to federated Q-learning



Single-agent Q-learning

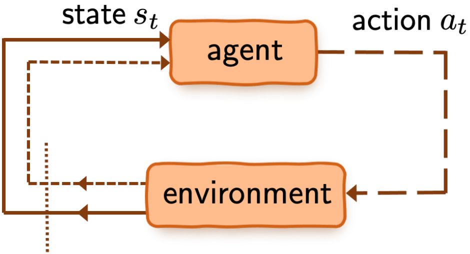


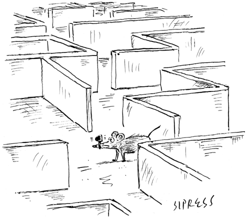
Federated Q-learning



Backgrounds:
Markov decision processes

Markov decision process (MDP)

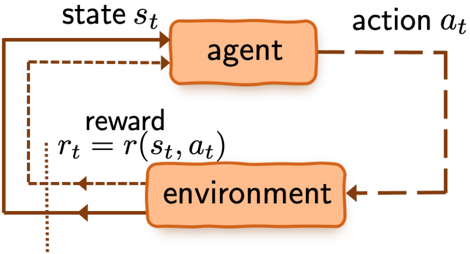


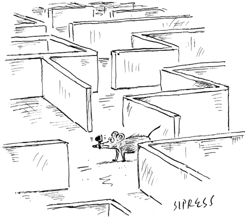


"Recalculating ... recalculating ..."

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

Markov decision process (MDP)

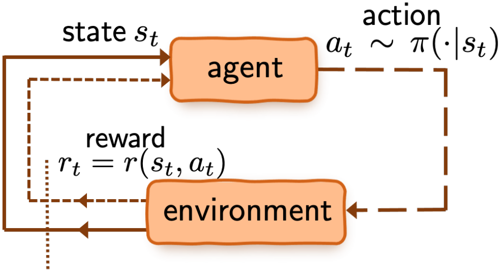


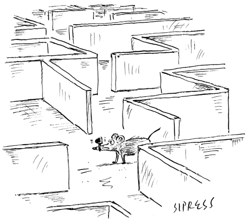


"Recalculating ... recalculating ..."

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

Markov decision process (MDP)

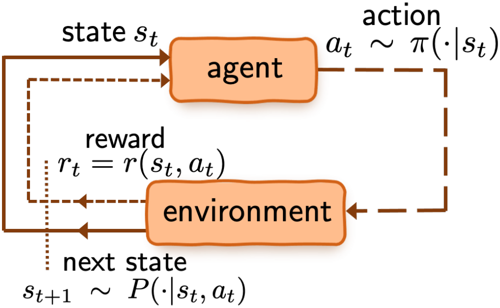


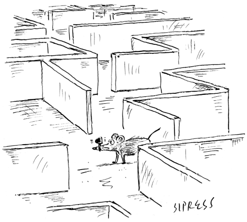


"Recalculating ... recalculating ..."

- $\mathcal{S}$: 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)

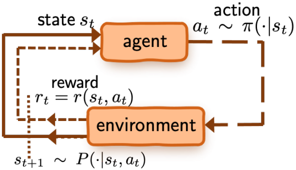


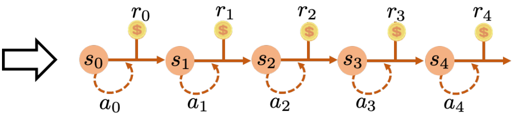


"Recalculating ... recalculating ..."

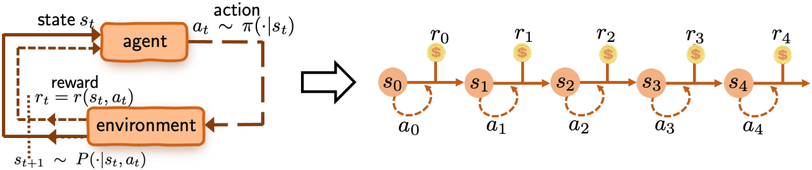
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $r(s,a) \in [0,1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)
- $P(\cdot|s,a)$: transition probabilities

Value function





Value function



Value function of policy π:

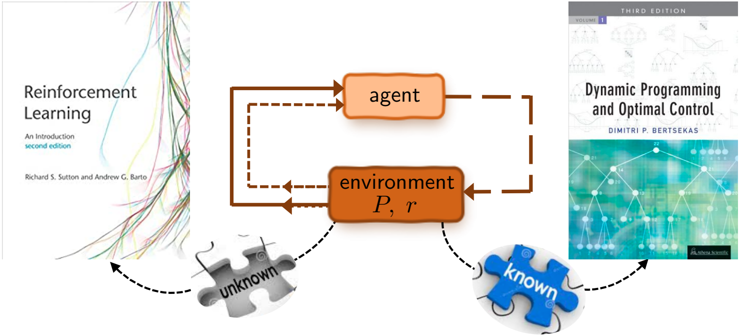
$$\forall s \in \mathcal{S}: \qquad V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right]$$

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 $\pi^\star(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^\star(s,a)$

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'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

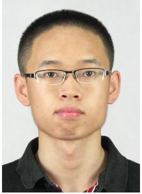
$$\mathcal{T}(Q^\star) = Q^\star$$

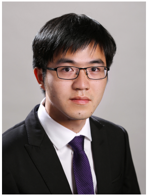
γ-contraction of Bellman operator:

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

Richard Bellman

Is Q-learning minimax-optimal?





Gen Li
CUHK

Changxiao Cai
UMich

Yuxin Chen
UPenn

Yuting Wei
UPenn

Q-learning: a classical model-free algorithm

Chris Watkins

Peter Dayan

Stochastic approximation for solving the **Bellman equation**

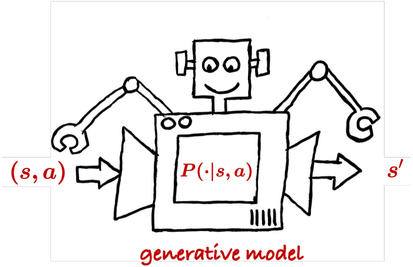
Robbins & Monro, 1951

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

where

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

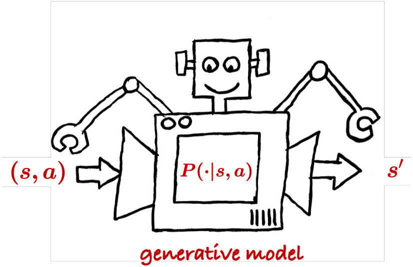
Synchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$ using samples collected from the generative model:

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

Synchronous Q-learning



Stochastic approximation for solving Bellman equation $Q^\star = \mathcal{T}(Q^\star)$ using samples collected from the generative model:

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

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

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

Prior art: achievability

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

Prior art: achievability

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

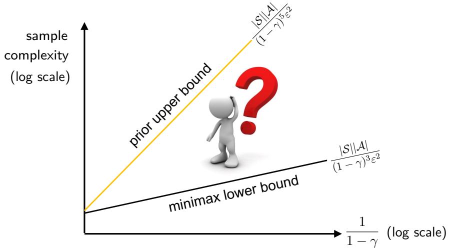
Minimax lower bound (Azar et al., 2013): $\widetilde{\Omega}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\Big)$.

Prior art: achievability

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

Minimax lower bound (Azar et al., 2013): $\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$.

paper	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	$\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^5\varepsilon^2}$



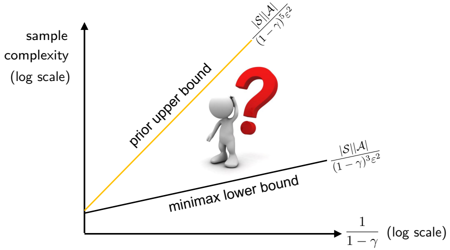
All prior results require sample size of at least $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^5\varepsilon^2}$!

Prior art: achievability

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

Minimax lower bound (Azar et al., 2013): $\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$.

paper	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 prior results require sample size of at least $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^5\varepsilon^2}$!

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

A sharpened sample complexity of Q-learning

Theorem (Li, Cai, Chen, Wei, Chi, OR 2024)

For any $0 < \varepsilon \le 1$, Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$ with sample complexity at most

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

- Improves dependency on the effective horizon $\frac{1}{1-\gamma}$.

A sharpened sample complexity of Q-learning

Theorem (Li, Cai, Chen, Wei, Chi, OR 2024)

For any $0 < \varepsilon \le 1$, Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$ with sample complexity at most

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

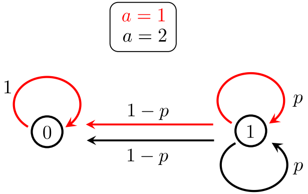
- Improves dependency on the effective horizon $\frac{1}{1-\gamma}$.
- Allows both constant and rescaled linear learning rate:

$$\frac{1}{1+\frac{c_1(1-\gamma)T}{\log^2 T}} \le \eta_t \le \frac{1}{1+\frac{c_2(1-\gamma)t}{\log^2 T}}$$

A curious numerical example

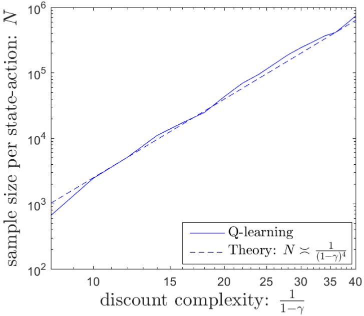
Numerical evidence: $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$ samples seem necessary ...

— *observed in Wainwright '19*



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

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



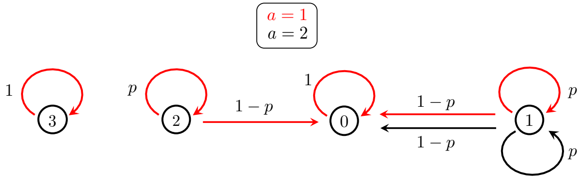
Q-learning is not minimax optimal

Theorem (Li, Cai, Chen, Wei, Chi, OR 2024)

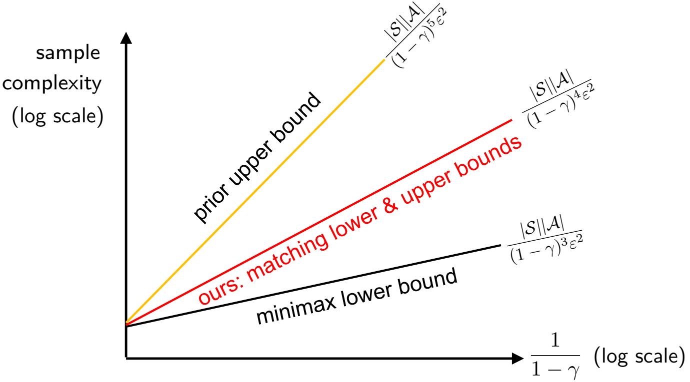
Assume $3/4 < \gamma \le 1$. For any $0 < \varepsilon \le 1$, there exists some MDP such that to achieve $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$, Q-learning needs at least a sample complexity of

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

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

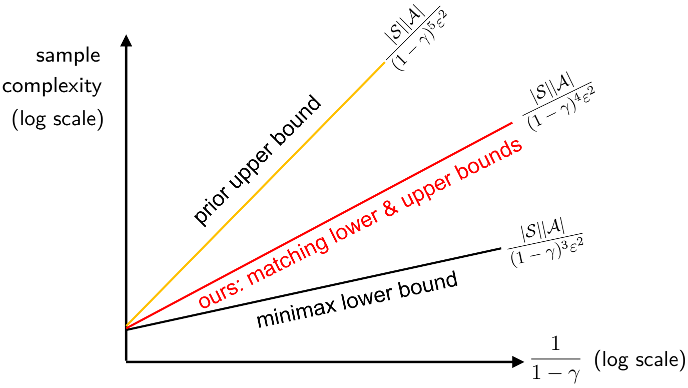


Where we stand now



Q-learning requires a sample size of $\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^4\varepsilon^2}$.

Where we stand now



Q-learning is not minimax optimal!

Why is Q-learning sub-optimal?

Over-estimation of Q-functions (Thrun and Schwartz, 1993; Hasselt, 2010):

- $\max_{a \in \mathcal{A}} \mathbb{E}X(a)$ 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, 2015).
- Motivated the design of double Q-learning (Hasselt, 2010).

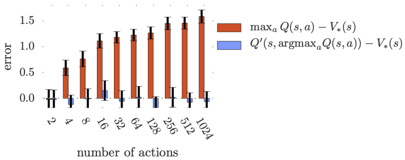


Figure 1: The orange bars show the bias in a single Q-learning update when the action values are $Q(s,a) = V_*(s) + \epsilon_a$ and the errors $\{\epsilon_a\}_{a=1}^m$ are independent standard normal random variables. The second set of action values Q', used for the blue bars, was generated identically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

Why is Q-learning sub-optimal?

Over-estimation of Q-functions (Thrun and Schwartz, 1993; Hasselt, 2010):

- $\max_{a\in\mathcal{A}} \mathbb{E}X(a)$ 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, 2015).
- Motivated the design of double Q-learning (Hasselt, 2010).

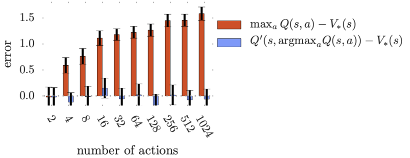
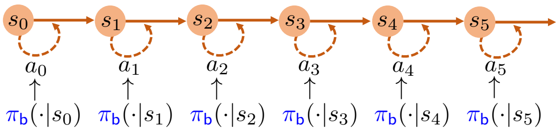


Figure 1: The orange bars show the bias in a single Q-learning update when the action values are $Q(s,a) = V_*(s) + \epsilon_a$ and the errors $\{\epsilon_a\}_{a=1}^m$ are independent standard normal random variables. The second set of action values Q', used for the blue bars, was generated identically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

Our work provides theoretical footings regarding the over-estimation issue of vanilla Q-learning.

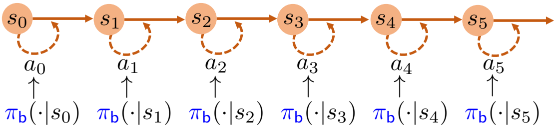
Asynchronous Q-learning



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

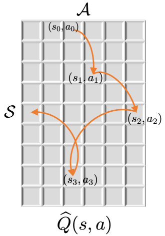
$$\underbrace{Q_{t+1}(s_t, a_t) = (1-\eta)Q_t(s_t, a_t) + \eta \mathcal{T}_t(Q_t)(s_t, a_t)}_{\textit{only} \text{ update } (s_t, a_t)\text{-th entry}}, \quad t \geq 0$$

Asynchronous Q-learning



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

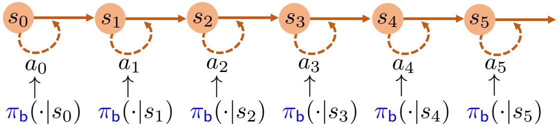
$$\underbrace{Q_{t+1}(s_t, a_t) = (1-\eta)Q_t(s_t, a_t) + \eta \mathcal{T}_t(Q_t)(s_t, a_t),}_{\textit{only}\text{ update }(s_t, a_t)\text{-th entry}} \quad t \geq 0$$



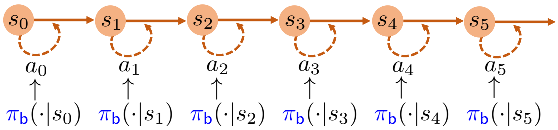
$$\mathcal{T}_t(Q)(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')$$

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

Sample complexity of asynchronous Q-learning



Sample complexity of asynchronous Q-learning



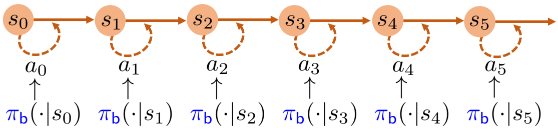
Key quantities:

- minimum state-action occupancy probability

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

- mixing time: t_{mix}

Sample complexity of asynchronous Q-learning



Key quantities:

- minimum state-action occupancy probability

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

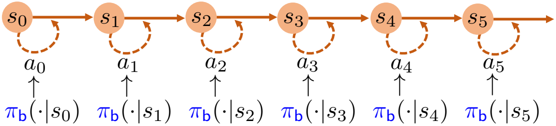
- mixing time: t_{mix}

Theorem (Li, Cai, Chen, Wei, Chi, OR 2024)

For any $0 < \varepsilon < 1$*, sample complexity of async Q-learning to yield* $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ *with high prob is at most*

$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)} \qquad \textit{(up to log factor)}$$

Sample complexity of asynchronous Q-learning



Key quantities:

- minimum state-action occupancy probability

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

- mixing time: t_{mix}

Theorem (Li, Cai, Chen, Wei, Chi, OR 2024)

For any $0 < \varepsilon < 1$, sample complexity of async Q-learning to yield $\|\widehat{Q} - Q^\star\|_\infty \leq \varepsilon$ with high prob is at most

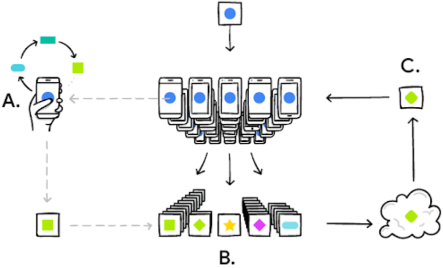
$$\frac{1}{\mu_{\mathsf{min}}(1-\gamma)^4\varepsilon^2} + \frac{t_{\mathsf{mix}}}{\mu_{\mathsf{min}}(1-\gamma)} \qquad \textit{(up to log factor)}$$

Federated Q-learning: linear speedup and beyond

Jiin Woo
CMU

Gauri Joshi
CMU

Can we harness the power of federated learning?




FORBES > INNOVATION > AI

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

Kevin Krewell Contributor
Tirias Research Contributor Group

Follow

Oct 15, 2021, 02:51pm EDT


How Apple personalizes Siri without hoovering up your data

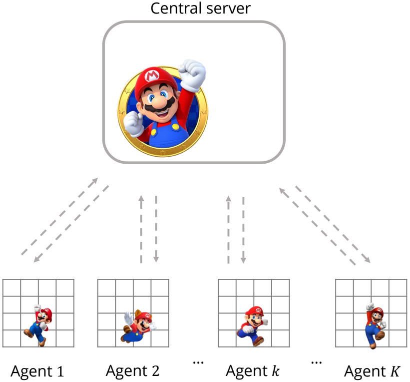
The tech giant is using privacy-preserving machine learning to improve its voice assistant while keeping your data on your phone.

By Karen Hao

December 11, 2019

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

RL meets federated learning



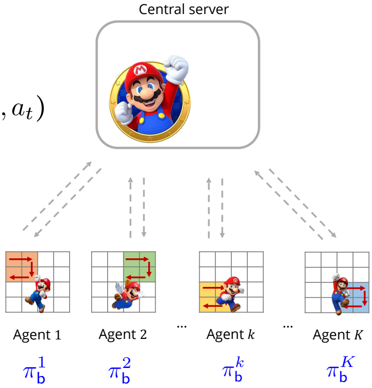
Federated reinforcement learning: enables multiple agents to collaboratively learn a global policy without sharing datasets.

Federated asynchronous Q-learning with local updates

- **Local Q-update:** 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

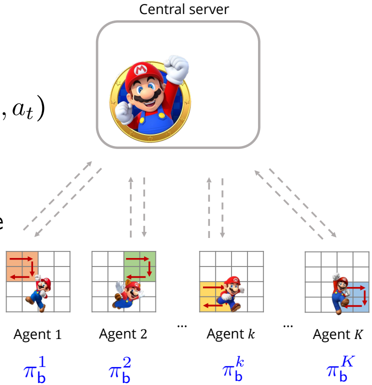
- **Local Q-update:** agent k performs τ rounds of local Q-learning updates:

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

and sends it to the server.

- **Periodic averaging:** 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

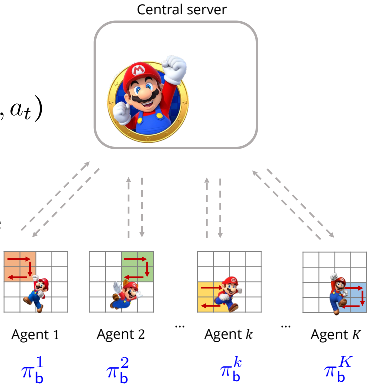
- **Local Q-update:** agent k performs τ rounds of local Q-learning updates:

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

  and sends it to the server.

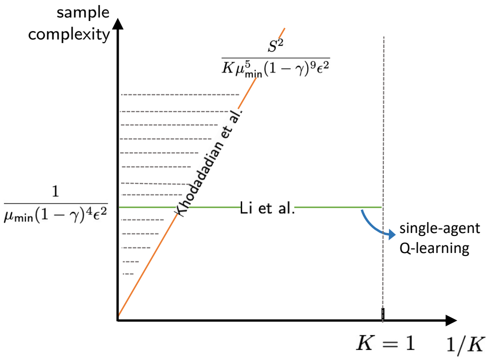
- **Periodic averaging:** the server averages the local updates and communicates it back to agents:

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



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

Prior art

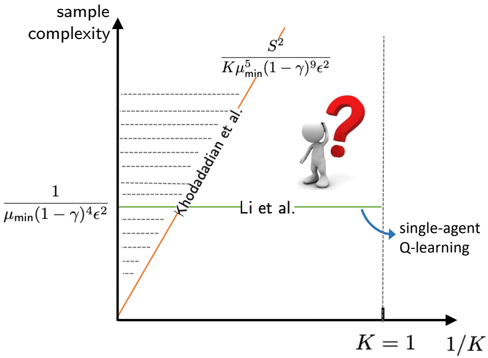


Key quantity: minimum state-action occupancy probability

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

Linear speedup only when $K \gg \frac{S^2}{\mu_{\min}^4(1-\gamma)^5}$

Prior art



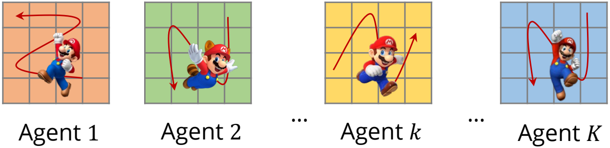
Key quantity: minimum state-action occupancy probability

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

But more curiously...

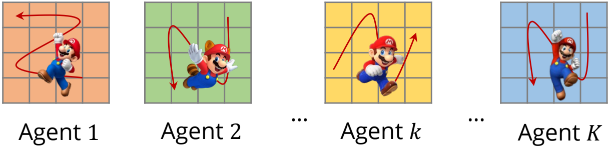
The benefit of collaboration?

Prior art requires **full coverage** of every agent over the entire state-action space (i.e., $\mu_{\text{min}} > 0$)...

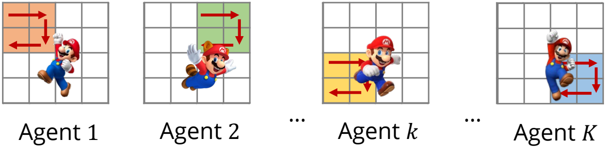


The benefit of collaboration?

Prior art requires **full coverage** of every agent over the entire state-action space (i.e., $\mu_{\text{min}} > 0$)...

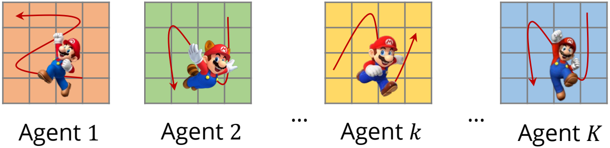


However, the power of collaboration really shines if we only need...

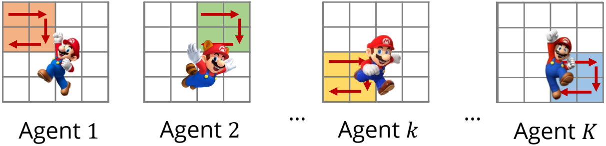


The benefit of collaboration?

Prior art requires **full coverage** of every agent over the entire state-action space (i.e., $\mu_{\text{min}} > 0$)...



However, the power of collaboration really shines if we only need...



Can we enable collaborative coverage while improve the dependency on salient parameters?

Key metrics

Collaborative coverage: minimum entry of the average stationary distribution

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

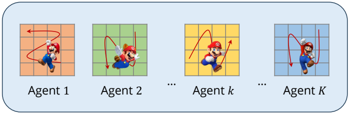
Key metrics

Collaborative coverage: minimum entry of the average stationary distribution

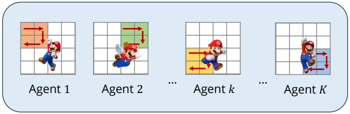
$$\mu_{\mathsf{avg}} = \min_{s,a} \frac{1}{K} \sum_{k=1}^{K} \mu_{\mathsf{b}}^{k}(s,a) \geq \mu_{\mathsf{min}}.$$

Heterogeneity of local behavior policies: density ratio of individual / average behavior policies

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



$C_{\mathsf{het}} = 1$



$C_{\mathsf{het}} = K$

Our theorem

Theorem (Jiin, Joshi, Chi, 2023+)

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

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

ignoring the burn-in cost that depends on the mixing times.

Our theorem

Theorem (Jiin, Joshi, Chi, 2023+)

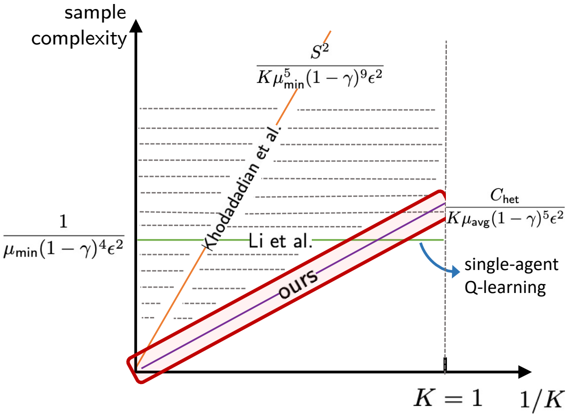
For sufficiently small $\varepsilon > 0$, federated asynchronous Q-learning yields $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$ with sample complexity at most

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

ignoring the burn-in cost that depends on the mixing times.

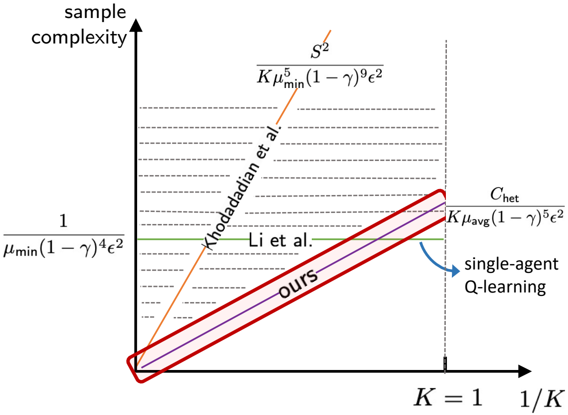
- The sync period obeys $\tau \le \frac{1}{4\eta} \min\left\{\frac{1-\gamma}{4}, \frac{1}{K}\right\}$; communication complexity is almost independent of ϵ.
- Near-optimal linear speedup when the local behavior policies are similar, $C_{\mathsf{het}} \approx 1$.
- Key idea: leave-one-out type arguments to decouple complicated statistical dependencies due to Markovian sampling and local updates.

Comparison with prior art



Linear speedup with near-optimal parameter dependencies!

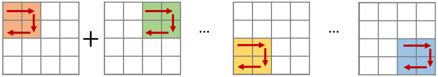
Curse of heterogeneity?



Still not good enough! Performance degenerates when local behavior policies are heterogeneous (i.e. $1 \ll C_{\text{het}}$). ☹

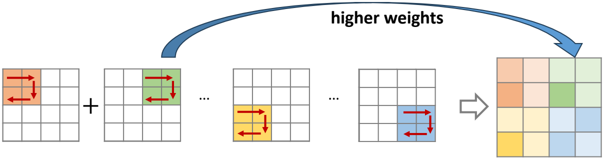
Importance averaging

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, 2023+)

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

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

ignoring the burn-in cost that depends on the mixing times.

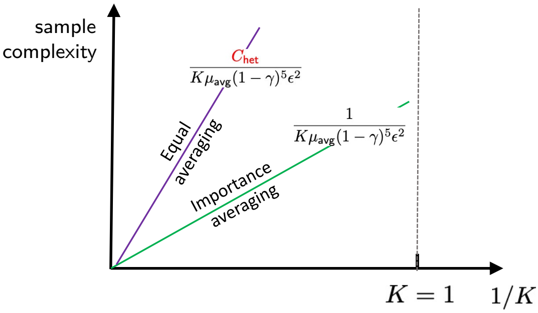
Our theorem

Theorem (Jiin, Joshi, Chi, 2023+)

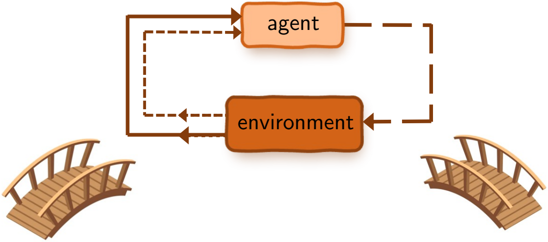
For sufficiently small $\varepsilon > 0$, federated asynchronous Q-learning with importance averaging yields $\|\widehat{Q} - Q^\star\|_\infty \le \varepsilon$ with sample complexity at most

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

ignoring the burn-in cost that depends on the mixing times.



Summary



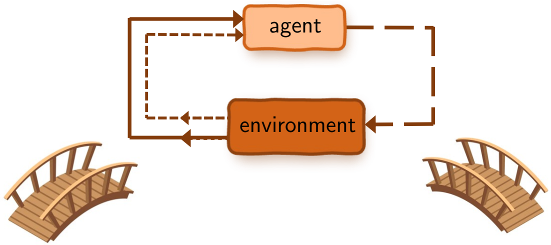
Single-agent Q-learning

Vanilla Q-learning is not minimax-optimal!

Federated Q-learning

Linear speedup even with heterogenous behavior policies!

Summary



Single-agent Q-learning

Vanilla Q-learning is not minimax-optimal!

Federated Q-learning

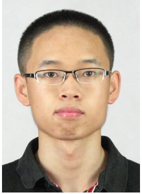
Linear speedup even with heterogenous behavior policies!

Ongoing work:

- Federated offline RL: how should we inject pessimism?
- Multi-task RL: heterogeneous environments across agents.

Bonus track:
robustness-statistical trade-offs in RL

Laixi Shi
CMU→Caltech



Gen Li
CUHK

Matthieu Geist
Google

Yuxin Chen
UPenn

Yuting Wei
UPenn

Safety and robustness in RL

Training environment

$\neq$

Test environment

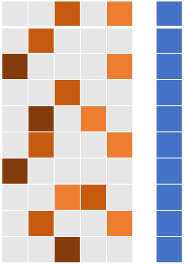
Training environment

$\neq$

Test environment

Can we learn optimal policies that are robust to model perturbations and sim-to-real gaps?

A curious question



empirical MDP

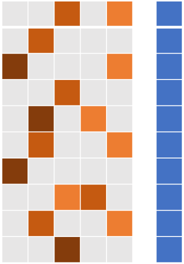
Learn the optimal policy of the nominal MDP?

Planning using *standard MDPs*

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

Planning using *robust MDPs*

A curious question



empirical MDP

Learn the optimal policy of the nominal MDP?

Planning using *standard MDPs*

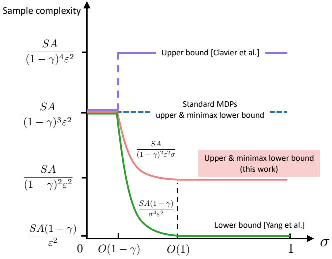
Learn the **robust** policy around the nominal MDP?

Planning using *robust MDPs*

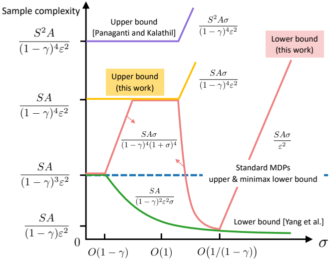
Robustness-statistical trade-off? Is there a statistical premium that one needs to pay in quest of additional robustness?

Surprising message

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



RMDP is simpler
under TV uncertainty



RMDP can be much harder
under χ^2 uncertainty

The statistical price of robustness depends on the choice of the uncertainty set! (NeurIPS 2023)

Thanks!

Statistical RL is a fruitful playground and still going strong!

- Is Q-Learning Minimax Optimal? A Tight Sample Complexity Analysis, *Operations Research*, 2024.
- The Blessing of Heterogeneity in Federated Q-Learning: Linear Speedup and Beyond, short version at *ICML 2023*.
- The Curious Price of Distributional Robustness in Reinforcement Learning with a Generative Model, short version at *NeurIPS 2023*.

Thanks!

The χ Group

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