Non-Asymptotic Analysis for Reinforcement Learning

Yuting Wei
UPenn

Yuxin Chen
UPenn

Yuejie Chi
CMU

SIGMETRICS Tutorial, June 2023
Non-asymptotic Analysis for Reinforcement Learning (Part 1)

Yuting Wei
Statistics & Data Science, Wharton University of Pennsylvania
SIGMETRICS, June 2023
Our wonderful collaborators

Gen Li  
UPenn → CUHK

Shicong Cen  
CMU

Chen Cheng  
Stanford

Laixi Shi  
CMU → Caltech

Yuling Yan  
Princeton → MIT

Changxiao Cai  
UPenn → UMich

Wenhao Zhan  
Princeton

Yuantao Gu  
Tsinghua

Jason Lee  
Princeton

Jianqing Fan  
Princeton
Recent successes in reinforcement learning (RL)

RL holds great promise in the next era of artificial intelligence.
Recap: Supervised learning

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

— pic from internet
Reinforcement learning (RL)

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

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

“Recalculating ... recalculating ...”
Sample efficiency

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

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

**Challenge:** design sample-efficient RL algorithms
Computational efficiency

Running RL algorithms might take a long time . . .

- enormous state-action space
- nonconvexity
Running RL algorithms might take a long time . . .

- enormous state-action space
- nonconvexity

**Challenge:** design computationally efficient RL algorithms
Understanding sample efficiency of RL requires a modern suite of non-asymptotic analysis tools.
This tutorial

(large-scale) optimization  (large-scale) optimization

(high-dimensional) statistics  (high-dimensional) statistics

Demystify sample- and computational efficiency of RL algorithms
This tutorial

Demystify sample- and computational efficiency of RL algorithms

Part 1. basics, and model-based RL

Part 2. value-based RL

Part 3. policy optimization

We will illustrate these approaches for learning standard, robust, and multi-agent RL with simulator/online/offline data.
Outline (Part 1)

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

- $S$: state space
- $A$: action space

\[ r(s, a) \in [0, 1] \]: immediate reward
Markov decision process (MDP)

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

$$r_t = r(s_t, a_t)$$
Infinite-horizon Markov decision process

- $S$: state space
- $A$: action space
- $r(s, a) \in [0, 1]$: immediate reward
- $\pi(\cdot|s)$: policy (or action selection rule)
Infinite-horizon Markov decision process

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

Value of policy \( \pi \): cumulative discounted reward

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

Value of policy $\pi$: cumulative discounted reward

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

- $\gamma \in [0, 1)$: discount factor
  - take $\gamma \to 1$ to approximate long-horizon MDPs
  - effective horizon: $\frac{1}{1-\gamma}$
Q-function (action-value function)

\( Q^\pi(s_0, a_0) \)

Q-function of policy \( \pi \):

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

- \((s_0, a_0, s_1, a_1, s_2, a_2, \cdots)\): induced by policy \( \pi \)
Q-function (action-value function)

Q-function of policy \( \pi \):

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

- \((a_0, s_1, a_1, s_2, a_2, \cdots)\): induced by policy \( \pi \)
finite-horizon MDPs

$h = 1, 2 \cdots, H$

- **$H$:** horizon length
- **$S$:** state space with size $S$
- **$A$:** 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

\[ h = 1, 2 \cdots, H \]

\[ \text{state } s_h \rightarrow \text{agent} \rightarrow \text{action } a_h \sim \pi_h(\cdot|s_h) \rightarrow \text{environment} \rightarrow \text{reward } r_h = r(s_h, a_h) \rightarrow \text{next state } s_{h+1} \sim P_h(\cdot|s_h, a_h) \rightarrow \text{agent} \rightarrow \text{state } s_h \]

value function: \[ V^\pi_h(s) := \mathbb{E} \left[ \sum_{t=h}^{H} r_h(s_h, a_h) \mid s_h = s \right] \]

Q-function: \[ Q^\pi_h(s, a) := \mathbb{E} \left[ \sum_{t=h}^{H} r_h(s_h, a_h) \mid s_h = s, a_h = a \right] \]
optimal policy $\pi^*$: maximizing value function $\max_\pi V^\pi$

**Proposition (Puterman’94)**

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

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

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

- optimal value / $Q$ function: $V^* := V^{\pi^*}$, $Q^* := Q^{\pi^*}$
Optimal policy and optimal value

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

- optimal value / Q function: $V^* := V^{\pi^*}$, $Q^* := Q^{\pi^*}$
- How to find this $\pi^*$?
Basic dynamic programming algorithms when MDP specification is known
Policy evaluation: Given MDP $\mathcal{M} = (S, A, r, P, \gamma)$ and policy $\pi: S \mapsto A$, how good is $\pi$? (i.e., how to compute $V^\pi(s), \forall s$?)
**Policy evaluation:** Given MDP $\mathcal{M} = (S, A, r, P, \gamma)$ and policy $\pi : S \mapsto A$, how good is $\pi$? (i.e., how to compute $V^\pi(s), \forall s$?)

**Possible scheme:**

- execute policy evaluation for each $\pi$
- find the optimal one
Policy evaluation: Bellman’s consistency equation

- $V^\pi / Q^\pi$: value / action-value function under policy $\pi$
Policy evaluation: Bellman’s consistency equation

- $V^\pi / Q^\pi$: value / action-value function under policy $\pi$

Bellman’s consistency equation

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

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

- Immediate reward
- Next state’s value

Richard Bellman
Policy evaluation: Bellman’s consistency equation

- $V^\pi / Q^\pi$: value / action-value function under policy $\pi$

Bellman’s consistency equation

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

$$Q^\pi(s, a) = r(s, a) + \gamma E_{s' \sim P(\cdot | s, a)}[V^\pi(s')]$$

- one-step look-ahead

Richard Bellman
Policy evaluation: Bellman’s consistency equation

- $V^\pi / Q^\pi$: value / action-value function under policy $\pi$

Bellman’s consistency equation

\[
V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s, a)] \\
Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)}[V^\pi(s')] \\
\text{immediate reward} \hspace{2cm} \text{next state’s value}
\]

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

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

Richard Bellman
Bellman operator

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

- one-step look-ahead
Optimal policy $\pi^*$: Bellman’s optimality principle

Bellman operator

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

- immediate reward
- next state’s value

• one-step look-ahead

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

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

$\gamma$-contraction of Bellman operator:

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

Richard Bellman
Two dynamic programming algorithms

Value iteration (VI)

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

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

Policy iteration (PI)

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

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

policy improvement: $\pi^{(t+1)}(s) = \arg\max_{a \in A} Q^{(t)}(s, a)$
When the model is unknown . . .
Need to learn optimal policy from samples w/o model specification
Three approaches

Model-based approach ("plug-in")

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

Model-based approach ("plug-in")
1. build an empirical estimate $\hat{P}$ for $P$
2. planning based on the empirical $\hat{P}$

Tutorial Part 2: Value-based approach
— learning w/o estimating the model explicitly

Tutorial Part 3: Policy-based approach
— optimization in the space of policies
Three approaches

Model-based approach ("plug-in")
1. build an empirical estimate $\hat{P}$ for $P$
2. planning based on the empirical $\hat{P}$

Tutorial Part 2: Value-based approach
— learning w/o estimating the model explicitly

Tutorial Part 3: Policy-based approach
— optimization in the space of policies
Model-based RL (a “plug-in” approach)

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

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

---

Kearns and Singh, 1999
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 \(|S||A| \times N\))
$\ell_\infty$-sample complexity: how many samples are required to learn an $\varepsilon$-optimal policy?

\[ \forall s: V^{\hat{\pi}}(s) \geq V^*(s) - \varepsilon \]
An incomplete list of works

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

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

\[ \hat{P}(s' | s, a) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{s'_{(i)} = s\} \]

\(\mathbb{1}\) denotes the empirical frequency.
Model estimation

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

Empirical estimates:

\[
\hat{P}(s'|s, a) = \frac{1}{N} \sum_{i=1}^{N} 1\{s'_i = s'\}
\]

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

empirical estimate: $\hat{P}$

• Can’t recover $P$ faithfully if sample size $\ll |S|^2|A|!$
Challenges in the sample-starved regime

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

\[
\tilde{O} \left( \frac{|S||A|}{(1 - \gamma)^3 \varepsilon^2} \right)
\]
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

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

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

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

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

• established upon leave-one-out analysis framework
Agarwal et al., 2019 still requires a burn-in sample size \( \gtrapprox \frac{|\mathcal{S}||\mathcal{A}|}{(1 - \gamma)^2} \). Question: is it possible to break this sample size barrier?
Agarwal et al., 2019 still requires a burn-in sample size $\gtrsim |S||A|^{(1-\gamma)^2}$

**Question:** is it possible to break this sample size barrier?
Agarwal et al., 2019 still requires a burn-in sample size $\gtrsim \frac{|S||A|}{(1-\gamma)^2}$

Question: is it possible to break this sample size barrier?
Agarwal et al., 2019 still requires a burn-in sample size \( \gtrsim \frac{|S||A|}{(1-\gamma)^2} \)

**Question:** is it possible to break this sample size barrier?
Perturbed model-based approach (Li et al. ’20)

Find policy based on the empirical MDP with slightly perturbed rewards

—Li et al., 2020
Theorem (Li, Wei, Chi, Chen ’20)

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

$$\tilde{O}\left(\frac{|S||A|}{(1-\gamma)^3\varepsilon^2}\right)$$
Optimal \( \ell_\infty \)-based sample complexity

**Theorem (Li, Wei, Chi, Chen ’20)**

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

\[
\tilde{O}\left(\frac{|S||A|}{(1-\gamma)^3\varepsilon^2}\right)
\]

- matches minimax lower bound: \( \tilde{\Omega}\left(\frac{|S||A|}{(1-\gamma)^3\varepsilon^2}\right) \) Azar et al., 2013
- full \( \varepsilon \)-range: \( \varepsilon \in \left(0, \frac{1}{1-\gamma}\right] \) \( \rightarrow \) no burn-in cost
- established upon more refined leave-one-out analysis and a perturbation argument
Prior art for asynchronous Q-learning

Even-Dar & Mansour '03

Qu & Wierman '20

Optimal

• with high prob., with sample complexity at most

\[ \frac{|S||A|}{(1 - \gamma)^3} \]

• \[ \frac{|S||A|}{(1 - \gamma)^2} \]

Wang, 2019

\[ |S||A| \]

Sidford et al., 2018b

\[ \frac{|S||A|}{1 - \gamma} \]

Wainwright, 2019a

Full

Pananjady and Wainwright, 2019

Khamaru et al., 2020

Agarwal et al. '19

Sidford et al. '18a

Sidford et al. '18b

Li et al. '20

Li et al. '22

1 + 3

\[ |S||A| \]

Yang and Wang, 2019

\[ (1 - \gamma)^2 \]

Wainwright, 2019b

\[ (1 - \gamma) \]

Agarwal et al., 2019

Kearns and Singh, 1999

\[ 1 \]

Sidford et al., 2018a

15 / 34

19 / 28

41 / 53

\[ \frac{1}{\varepsilon^2} \]

minimax lower bound

Li et al. '20

\[ \frac{|S||A|}{1 - \gamma} \]

Agarwal et al. '19

\[ \frac{|S||A|}{(1 - \gamma)^2} \]

Sidford et al. '18a

\[ \frac{|S||A|}{(1 - \gamma)^3} \]

\[ |S||A| \]

\[ |S| \]

\[ |A| \]

\[ \frac{1}{\varepsilon^2} \]

\[ \frac{1}{\varepsilon^2} \]

\[ \frac{1}{\varepsilon^2} \]

\[ \frac{1}{\varepsilon^2} \]

\[ \frac{1}{\varepsilon^2} \]
Model-based RL (a “plug-in” approach)

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

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

- medical records
- data of self-driving
- clicking times of ads
Offline RL / batch RL

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

medical records  data of self-driving  clicking times of ads

**Question:** Can we design algorithms based solely on historical data?
A historical dataset $\mathcal{D} = \{(s^{(i)}, a^{(i)}, s'^{(i)})\}: N$ independent copies of

$$s \sim \rho^b, \quad a \sim \pi^b(\cdot | s), \quad s' \sim P(\cdot | s, a)$$

for some state distribution $\rho^b$ and behavior policy $\pi^b$
A historical dataset $\mathcal{D} = \{(s^{(i)}, a^{(i)}, s'^{(i)})\}: N$ independent copies of 

$$s \sim \rho^b, \quad a \sim \pi^b(\cdot | s), \quad s' \sim P(\cdot | s, a)$$

for some state distribution $\rho^b$ and behavior policy $\pi^b$

**Goal:** given some test distribution $\rho$ and accuracy level $\varepsilon$, find an $\varepsilon$-optimal policy $\hat{\pi}$ based on $\mathcal{D}$ obeying

$$V^*(\rho) - V^{\hat{\pi}}(\rho) = \mathbb{E}_{s \sim \rho} [V^*(s)] - \mathbb{E}_{s \sim \rho} [V^{\hat{\pi}}(s)] \leq \varepsilon$$

— in a sample-efficient manner
Challenges of offline RL

- Distribution shift:

\[
\text{distribution}(\mathcal{D}) \neq \text{target distribution under } \pi^*
\]
Challenges of offline RL

• **Distribution shift:**

\[ \text{distribution}(\mathcal{D}) \neq \text{target distribution under } \pi^* \]

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

Samples cover all \((s, a)\) & all policies

Uniform coverage over entire space (sufficiently explored)
Challenges of offline RL

- **Distribution shift:**

  \[ \text{distribution}(\mathcal{D}) \neq \text{target distribution under } \pi^* \]

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

Practically, samples cover all \((s, a)\) & all policies

**Ideally**, uniform coverage over entire space (sufficiently explored)

**Historically**, partial coverage (inadequately explored)
How to quantify quality of historical dataset $\mathcal{D}$ (induced by $\pi^b$)?
How to quantify quality of historical dataset $\mathcal{D}$ (induced by $\pi^b$)?

**Single-policy concentrability coefficient**

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

where $d_{\pi}(s,a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}((s^t, a^t) = (s, a) | \pi)$
How to quantify quality of historical dataset $\mathcal{D}$ (induced by $\pi^b$)?

**Single-policy concentrability coefficient**

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

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

- captures distributional shift
- allows for partial coverage
Key idea: pessimism in the face of uncertainty


upper confidence bounds
— promote exploration of under-explored \((s, a)\)
Key idea: pessimism in the face of uncertainty


**Upper confidence bounds**
- promote exploration of under-explored \((s, a)\)

**Lower confidence bounds**
- stay cautious about under-explored \((s, a)\)
Key idea: pessimism in the face of uncertainty


A model-based offline algorithm: VI-LCB

1. build empirical model $\hat{P}$

2. (value iteration) for $t \leq \tau_{\text{max}}$:

$$\hat{Q}_t(s, a) \leftarrow \left[ r(s, a) + \gamma \langle \hat{P}(\cdot | s, a), \hat{V}_{t-1} \rangle \right]_+$$

for all $(s, a)$, where $\hat{V}_t(s) = \max_a \hat{Q}_t(s, a)$
A key idea: pessimism in the face of uncertainty

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

A model-based offline algorithm: VI-LCB

1. build empirical model $\hat{P}$

2. (pessimistic value iteration) for $t \leq \tau_{\text{max}}$:

$$
\hat{Q}_t(s, a) \leftarrow \left[ r(s, a) + \gamma \langle \hat{P}(\cdot | s, a), \hat{V}_{t-1} \rangle - \underbrace{b(s, a; \hat{V}_{t-1})}_{\text{penalize poorly visited } (s,a)} \right] + 
$$

for all $(s, a)$, where $\hat{V}_t(s) = \max_a \hat{Q}_t(s, a)$
Key idea: pessimism in the face of uncertainty


A model-based offline algorithm: VI-LCB

1. build empirical model $\hat{P}$

2. (pessimistic value iteration) for $t \leq \tau_{\text{max}}$:

$$\hat{Q}_t(s, a) \leftarrow \left[ r(s, a) + \gamma \langle \hat{P}(\cdot | s, a), \hat{V}_{t-1} \rangle - b(s, a; \hat{V}_{t-1}) \right]_+$$

penalize poorly visited $(s, a)$

compared w/ prior works

• no need of variance reduction
• variance-aware penalty
Theorem (Li, Shi, Chen, Chi, Wei ’22)

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

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \leq \varepsilon$$

with high prob., with sample complexity at most

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

• matches minimax lower bound: $\tilde{\Omega}\left(\frac{SC^*}{(1-\gamma)^3\varepsilon^2}\right)$

• depends on distribution shift (as reflected by $C^*$)

• full $\varepsilon$-range (no burn-in cost)
Minimax optimality of model-based offline RL

Theorem (Li, Shi, Chen, Chi, Wei ’22)

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

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \leq \varepsilon$$

with high prob., with sample complexity at most

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

- matches minimax lower bound: $\tilde{\Omega}\left(\frac{SC^*}{(1-\gamma)^3\varepsilon^2}\right)$, Rashidinejad et al, 2021
- depends on distribution shift (as reflected by $C^*$)
- full $\varepsilon$-range (no burn-in cost)
Prior art for asynchronous Q-learning paper

Even-Dar & Mansour '03
Beck & Srikant '12
Qu & Wierman '20

Sample complexity

$SC^*$
$\frac{SC^*}{(1 - \gamma)^3}$
$\frac{SC^*}{(1 - \gamma)^5}$

minimax lower bound

Li et al. '22

Li et al. '20
Li et al. '22

Rashidinejad et al. Yan et al.

Xie et al. Shi et al.

Kearns and others

Agarwal et al., 2019
Khamaru et al., 2020
Yang and Wang, 2019

Duality

Lagrangian

variables

learning rate

no burn-in cost

standard form problem

rescaled linear

$\frac{SC^*}{(1 - \gamma)^3 \varepsilon^2}$

\[ \frac{SC^*}{1 - \gamma} \]

\[ \frac{SC^*}{(1 - \gamma)^3} \]

\[ \frac{SC^*}{(1 - \gamma)^5} \]

$\frac{1}{\varepsilon^2}$
Model-based RL (a “plug-in” approach)

1. Sampling from a generative model (simulator)
2. Offline RL / batch RL
3. Robust RL
Safety and robustness in RL

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

Training environment ≠ Test environment
Safety and robustness in RL

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

![Image of training and test environments]

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

Uncertainty set of the nominal transition kernel $P^o$:

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

Robust value/Q function of policy $\pi$:

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

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

The optimal robust policy $\pi^*$ maximizes $V^{\pi,\sigma}(\rho)$
Robust Bellman’s optimality equation: the optimal robust policy $\pi^*$ and optimal robust value $V^{*,\sigma} := V^{\pi^*,\sigma}$ satisfy

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

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

(Iyengar. ’05, Nilim and El Ghaoui. ’05)
Robust Bellman’s optimality equation: the optimal robust policy $\pi^*$ and optimal robust value $V^{\pi^*}\sigma := V^{\pi^*}\sigma$ satisfy

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

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

Robust value iteration:

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

where $V(s) = \max_a Q(s, a)$.
Learning distributionally robust MDPs

Sample trajectory and behavior policy $(s, a)$

**Goal**: estimate optimal value function $V^*$ based on sample trajectory

**Key quantities of sample trajectory**
- minimum state-action occupancy probability $\mu_{\text{min}} := \min \mu_{\text{fi b}}(s, a)$
- mixing time: $t_{\text{mix}}$

**Nominal Transition kernel**

$P^o(\cdot|s, a)$

Stationary distribution

**Goal of robust RL**: given $D := \{(s_i, a_i, s'_i)\}_{i=1}^N$ from the nominal environment $P_0$, find an $\varepsilon$-optimal robust policy $\hat{\pi}$ obeying $V^{\star},\sigma(\rho) - V^{\hat{\pi}},\sigma(\rho) \leq \varepsilon$ — in a sample-efficient manner
Goal of robust RL: given $D := \{(s_i, a_i, s'_i)\}_{i=1}^N$ from the nominal environment $P^0$, find an $\varepsilon$-optimal robust policy $\hat{\pi}$ obeying

$$V^{*,\sigma}(\rho) - V^{\hat{\pi},\sigma}(\rho) \leq \varepsilon$$

— in a sample-efficient manner
A curious question

Two approaches

Model-based approach ("plug-in")
1. build empirical estimate \( P \) for \( P \)
2. planning based on empirical \( P \)

Model-free approach (e.g. Q-learning, SARSA)
— learning w/o constructing model explicitly

Learn the optimal policy of the nominal MDP?
Learn the robust policy around the nominal MDP?

empirical MDP
A curious question

Two approaches
Model-based approach ("plug-in")
1. build empirical estimate \( P \) for \( P \)
2. planning based on empirical \( P \)

Model-free approach (e.g. Q-learning, SARSA)
— learning w/o constructing model explicitly

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?
When the uncertainty set is TV

Sample complexity

\[
\frac{SA}{(1 - \gamma)^4 \varepsilon^2}
\]

Upper bound [Clavier et al.]

\[
\frac{SA}{(1 - \gamma)^3 \varepsilon^2}
\]

Standard MDPs upper & minimax lower bound

\[
\frac{SA}{(1 - \gamma)^2 \varepsilon^2 \sigma}
\]

Upper & minimax lower bound (this work)

\[
\frac{SA(1 - \gamma)}{\varepsilon^2}
\]

Lower bound [Yang et al.]

\[
\frac{SA(1 - \gamma)}{\sigma^4 \varepsilon^2}
\]

Upper & minimax lower bound (this work)
When the uncertainty set is TV

RMDPs are easier to learn than standard MDPs.
When the uncertainty set is $\chi^2$ divergence

Sample complexity

- $S^2 A \frac{1}{(1 - \gamma)^4 \varepsilon^2}$
- $SA \frac{1}{(1 - \gamma)^4 \varepsilon^2}$
- $SA \frac{1}{(1 - \gamma)^3 \varepsilon^2}$
- $SA \frac{1}{(1 - \gamma) \varepsilon^2}$

Upper bound

- Standard MDPs upper & minimax lower bound
- Upper bound (this work)
- Lower bound [Yang et al.]

Lower bound (this work)

Standard MDPs

$O(1 - \gamma)$ $O(1)$ $O\left(\frac{1}{(1 - \gamma)}\right)$

$\sigma$
When the uncertainty set is $\chi^2$ divergence

RMDPs can be **harder** to learn than standard MDPs.
Summary of this part

Model-based RL (a “plug-in” approach)

• Sampling from a generative model (simulator)
• Offline RL / batch RL
• Robust RL

Papers:
“Breaking the sample size barrier in model-based reinforcement learning with a generative model,” G Li, Y Wei, Y Chi, Y Chen, NeurIPS’20, Operators Research’23
“Settling the sample complexity of model-based offline reinforcement learning,” G Li, L Shi, Y Chen, Y Chi, Y Wei, 2022
“The curious price of distributional robustness in reinforcement learning with a generative model,” L Shi, G Li, Y Wei, Y Chen, M Geist, Y Chi, 2023
Non-Asymptotic Analysis for Reinforcement Learning (Part 2)

Yuxin Chen
Wharton Statistics & Data Science, SIGMETRICS 2023
Multi-agent RL with a generative model
Multi-agent reinforcement learning (MARL)
Two-player zero-sum Markov games (finite-horizon)

- $S = [S]$: state space
- $H$: horizon
- $A = [A]$: action space of max-player
- $B = [B]$: action space of min-player
Two-player zero-sum Markov games (finite-horizon)

- $S = [S]$: state space
- $A = [A]$: action space of max-player
- $H$: horizon
- $B = [B]$: action space of min-player
- immediate reward: max-player $r(s, a, b) \in [0, 1]$
  min-player $-r(s, a, b)$
Two-player zero-sum Markov games (finite-horizon)

- $S = [S]$: state space
- $A = [A]$: action space of max-player
- $H$: horizon
- $B = [B]$: action space of min-player
- immediate reward: max-player $r(s, a, b) \in [0, 1]$  
  min-player $-r(s, a, b)$
- $\mu : S \times [H] \rightarrow \Delta(A)$: policy of max-player
- $\nu : S \times [H] \rightarrow \Delta(B)$: policy of min-player

\[
\begin{align*}
\text{state } s_h & \rightarrow \text{max-player} \\
& \quad a_h \sim \mu_h(\cdot | s_h) \\
\text{reward } r_h & \rightarrow \text{min-player} \\
& \quad b_h \sim \nu_h(\cdot | s_h) \\
\text{state } s_h & \rightarrow \text{environment} \\
\text{reward } -r_h & \rightarrow \text{environment}
\end{align*}
\]
Two-player zero-sum Markov games (finite-horizon)

- $S = [S]$: state space
- $H$: horizon
- $A = [A]$: action space of max-player
- $B = [B]$: action space of min-player
- Immediate reward: max-player $r(s, a, b) \in [0, 1]$
  min-player $-r(s, a, b)$
- $\mu : S \times [H] \to \Delta(A)$: policy of max-player
- $\nu : S \times [H] \to \Delta(B)$: policy of min-player
- $P_h(\cdot \mid s, a, b)$: unknown transition probabilities
**Value function** under *independent* policies \((\mu, \nu)\) (no coordination)

\[
V^{\mu,\nu}(s) := \mathbb{E} \left[ \sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid s_1 = s \right]
\]
Value function under independent policies \((\mu, \nu)\) (no coordination)

\[
V^{\mu,\nu}(s) := \mathbb{E} \left[ \sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid s_1 = s \right]
\]

- Each agent seeks optimal policy maximizing her own value
**Value function** under *independent* policies \((\mu, \nu)\) *(no coordination)*

\[
V^{\mu,\nu}(s) := \mathbb{E} \left[ \sum_{h=1}^{H} r_h(s_h, a_h, b_h) \mid s_1 = s \right]
\]

- Each agent seeks **optimal policy** maximizing her own value
- But two agents have conflicting goals . . .
Compromise: Nash equilibrium (NE)

An NE policy pair \((\mu^*, \nu^*)\) obeys

\[
\max_{\mu} V^{\mu, \nu^*} = V^{\mu^*, \nu^*} = \min_{\nu} V^{\mu^*, \nu}
\]

John von Neumann  John Nash
Compromise: Nash equilibrium (NE)

An NE policy pair \((\mu^*, \nu^*)\) obeys

\[
\max_{\mu} V^{\mu,\nu^*} = V^{\mu^*,\nu^*} = \min_{\nu} V^{\mu^*,\nu}
\]

- no unilateral deviation is beneficial
Compromise: Nash equilibrium (NE)

An NE policy pair \((\mu^*, \nu^*)\) obeys

\[
\max_{\mu} V_{\mu, \nu^*} = V_{\mu^*, \nu^*} = \min_{\nu} V_{\mu^*, \nu}
\]

- no unilateral deviation is beneficial
- no coordination between two agents (they act \textit{independently})
Compromise: Nash equilibrium (NE)

An $\varepsilon$-NE policy pair $(\hat{\mu}, \hat{\nu})$ obeys

$$\max_{\mu} V^{\mu, \hat{\nu}} - \varepsilon \leq V^{\hat{\mu}, \hat{\nu}} \leq \min_{\nu} V^{\hat{\mu}, \nu} + \varepsilon$$

- no unilateral deviation is beneficial
- no coordination between two agents (they act independently)
Learning NEs with a simulator

input: any \((s, a, b, h)\)
output: an independent sample \(s' \sim P_h(\cdot | s, a, b)\)
Learning NEs with a simulator

**input:** any \((s, a, b, h)\)

**output:** an independent sample \(s' \sim P_h(\cdot | s, a, b)\)

**Question:** how many samples are sufficient to learn an \(\varepsilon\)-Nash policy pair?
Model-based approach (non-adaptive sampling)

— Zhang, Kakade, Başar, Yang ’20

1. for each \((s, a, b, h)\), call simulator \(N\) times
Model-based approach (non-adaptive sampling)

— Zhang, Kakade, Baṣar, Yang ’20

1. for each \((s, a, b, h)\), call simulator \(N\) times
Model-based approach (non-adaptive sampling)

— Zhang, Kakade, Başar, Yang ’20

1. for each \((s, a, b, h)\), call simulator \(N\) times
2. build empirical model \(\hat{P}\)
Model-based approach (non-adaptive sampling)  

— Zhang, Kakade, Başar, Yang ’20

1. for each \((s, a, b, h)\), call simulator \(N\) times
2. build empirical model \(\hat{P}\), and run “plug-in” methods
Model-based approach (non-adaptive sampling)

— Zhang, Kakade, Başar, Yang ’20

Call generative model

One can query the generative model with any state-action-step combination \((s, a, b, h)\), and obtain \(s' \sim P_h(s' | s, a, b)\).

for any \((s, h)\)

for each \((a, b)\)

1. for each \((s, a, b, h)\), call simulator \(N\) times
2. build empirical model \(\hat{P}\), and run “plug-in” methods

sample complexity: \(\frac{H^4 S A B}{\varepsilon^2}\)
Curse of multiple agents

1 player: $A$

Let’s look at the size of joint action space . . .
Curse of multiple agents

Let’s look at the size of joint action space . . .
Curse of multiple agents

Let’s look at the size of joint action space …
Curse of multiple agents

1 player: $A$

2 players: $AB$

$m$ players: $A_1A_2\cdots A_m$

# joint actions blows up geometrically in # players!
Theorem 1 (Li, Chi, Wei, Chen '22)

For any $0 < \varepsilon \leq H$, one can design an algorithm that finds an $\varepsilon$-Nash policy pair $(\hat{\mu}, \hat{\nu})$ with high prob., with sample complexity at most 

$$\tilde{\mathcal{O}} \left( H^4 S (A + B) \varepsilon^2 \right).$$

1

$A + B$
Theorem 1 (Li, Chi, Wei, Chen '22)

For any $0 < \epsilon \leq H$, one can design an algorithm that finds an $\epsilon$-Nash policy pair $(\hat{\mu}, \hat{\nu})$ with high prob., with sample complexity at most $\tilde{O}(H^4 S (A + B) \epsilon^2)$. 

V-learning

model-based

$H^4$

$H^6$

$A + B$

$AB$

#actions
Theorem 1 (Li, Chi, Wei, Chen ’22)

For any $0 < \varepsilon \leq H$, one can design an algorithm that finds an $\varepsilon$-Nash policy pair $(\hat{\mu}, \hat{\nu})$ with high prob., with sample complexity at most

$$\tilde{O}\left(\frac{H^4S(A + B)}{\varepsilon^2}\right)$$

(minimax-optimal $\forall \varepsilon$)
Model-free / value-based RL

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

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

- one-step look-ahead
A starting point: Bellman optimality principle

Bellman operator

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

- immediate reward
- next state’s value

- one-step look-ahead

Bellman equation: \( Q^* \) is unique solution to

\[ T(Q^*) = Q^* \]
A starting point: Bellman optimality principle

Bellman operator

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

- one-step look-ahead

Bellman equation: \( Q^* \) is unique solution to

\[ T(Q^*) = Q^* \]

- takeaway message: it suffices to solve the Bellman equation
- challenge: how to solve it using stochastic samples?
Stochastic approximation for solving the Bellman equation

Robbins & Monro, 1951

\[ T(Q) - Q = 0 \]

where

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

immediate reward

next state's value
Q-learning: a stochastic approximation algorithm

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

$$Q_{t+1}(s, a) = Q_t(s, a) + \eta_t(\mathcal{T}_t(Q_t)(s, a) - Q_t(s, a)), \quad t \geq 0$$

sample transition $(s, a, s')$
Q-learning: a stochastic approximation algorithm

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

$$Q_{t+1}(s, a) = Q_t(s, a) + \eta_t(\mathcal{T}_t(Q_t)(s, a) - Q_t(s, a)), \quad t \geq 0$$

sample transition $(s, a, s')$

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

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

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

--- Kearns, Singh ’99

Each iteration, draw an independent sample \((s, a, s')\) for given \((s, a)\)
Synchronous Q-learning

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

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

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

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

**synchronous:** all state-action pairs are updated simultaneously

- total sample size: $T|S||A|$
Sample complexity of synchronous Q-learning

**Theorem 2 (Li, Cai, Chen, Wei, Chi ’21)**

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

$$\tilde{O}\left(\frac{|S||A|}{(1-\gamma)^4\varepsilon^2}\right) \quad \text{if } |A| \geq 2$$

$$\tilde{O}\left(\frac{|S|}{(1-\gamma)^3\varepsilon^2}\right) \quad \text{if } |A| = 1 \quad (TD \ learning)$$
Sample complexity of synchronous Q-learning

**Theorem 2 (Li, Cai, Chen, Wei, Chi ’21)**

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

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

- Covers both constant and rescaled linear learning rates:

  \[\eta_t \equiv \frac{1}{1 + \frac{c_1(1-\gamma)T}{\log^2 T}} \quad \text{or} \quad \eta_t = \frac{1}{1 + \frac{c_2(1-\gamma)t}{\log^2 T}}\]
Sample complexity of synchronous Q-learning

Theorem 2 (Li, Cai, Chen, Wei, Chi ’21)

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

$$\tilde{O}\left(\frac{|S||A|}{(1-\gamma)^4\varepsilon^2}\right) \quad \text{if } |A| \geq 2 \quad (\text{?})$$

$$\tilde{O}\left(\frac{|S|}{(1-\gamma)^3\varepsilon^2}\right) \quad \text{if } |A| = 1 \quad (\text{minimax optimal})$$

<table>
<thead>
<tr>
<th>other papers</th>
<th>sample complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Even-Dar &amp; Mansour ’03</td>
<td>$2\frac{1}{1-\gamma} \frac{</td>
</tr>
<tr>
<td>Beck &amp; Srikant ’12</td>
<td>$\frac{</td>
</tr>
<tr>
<td>Wainwright ’19</td>
<td>$\frac{</td>
</tr>
<tr>
<td>Chen, Maguluri, Shakkottai, Shanmugam ’20</td>
<td>$\frac{</td>
</tr>
</tbody>
</table>
All this requires sample size at least \( \frac{|S||A|}{(1-\gamma)^4 \varepsilon^2} \) \((|A| \geq 2)\) ...
All this requires sample size at least \( \frac{|S||A|}{(1-\gamma)^4 \epsilon^2} \) \( (|A| \geq 2) \) …

**Question:** Is Q-learning sub-optimal, or is it an analysis artifact?
A numerical example: \[
\frac{|S||A|}{(1-\gamma)^4\varepsilon^2}
\] 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 3 (Li, Cai, Chen, Wei, Chi, 2021)**

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

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

samples.
Q-learning is NOT minimax optimal

**Theorem 3** (Li, Cai, Chen, Wei, Chi, 2021)

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**

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

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

![Diagram](https://via.placeholder.com/150)
Q-learning is NOT minimax optimal

**Theorem 3 (Li, Cai, Chen, Wei, Chi, 2021)**

*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*

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

---

**Diagram:**

The diagram illustrates the sample complexity of various algorithms in the context of learning rates and sample sizes. The axes are labeled:

- Sample complexity (log scale)
- $1 / (1 - \gamma)$ (log scale)

Key points include:

- Wainwright '19
- Chen, Maguluri, Shakkottai, Shamugam 20
- Li et al. '21 (sharp characterization)

The minimax limit is marked along with the sample complexity curves for different algorithms.
Improving sample complexity via variance reduction

— a powerful idea from finite-sum stochastic optimization
**Variance-reduced Q-learning updates** (Wainwright ’19)

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

\[
Q_t(s, a) = (1 - \eta)Q_{t-1}(s, a) + \eta\left(\mathcal{T}_t(Q_{t-1}) - \mathcal{T}_t(\overline{Q}) + \widetilde{\mathcal{T}}(\overline{Q})\right)(s, a)
\]

*use $\overline{Q}$ to help reduce variability*
Variance-reduced Q-learning updates (Wainwright ’19)

— inspired by SVRG (Johnson & Zhang ’13)

\[
Q_t(s, a) = (1 - \eta)Q_{t-1}(s, a) + \eta \left( T_t(Q_{t-1}) - T_t(\overline{Q}) + \tilde{T}(\overline{Q}) \right)(s, a)
\]

use $\overline{Q}$ to help reduce variability

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

\[
T_t(Q)(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a')
\]

\[
\tilde{T}(Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[ \max_{a'} Q(s', a') \right]
\]
An epoch-based stochastic algorithm

— inspired by Johnson & Zhang ’13

\begin{itemize}
\item update variance-reduced $\tilde{Q}$ and $\tilde{T}(\tilde{Q})$ (which stay fixed in the rest of the epoch)
\item run variance-reduced Q-learning updates iteratively
\end{itemize}

for each epoch

1. update $\tilde{Q}$ and $\tilde{T}(\tilde{Q})$ (which stay fixed in the rest of the epoch)
2. run variance-reduced Q-learning updates iteratively
Theorem 4 (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

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

- allows for more aggressive learning rates
Sample complexity of variance-reduced Q-learning

Theorem 4 (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

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

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

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

**Observed:** \( \{s_t, a_t, r_t\}_{t \geq 0} \) generated by behavior policy \( \pi_b \)

**Goal:** learn optimal value \( V^* \) and \( Q^* \) based on sample trajectory
Markovian samples and behavior policy

observed:

\[ s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} s_4 \xrightarrow{a_4} s_5 \]

\[ \pi_b(s_0) \quad \pi_b(s_1) \quad \pi_b(s_2) \quad \pi_b(s_3) \quad \pi_b(s_4) \quad \pi_b(s_5) \]

learn:

\[ s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} s_4 \xrightarrow{a_4} s_5 \]

\[ \pi^*(s_0) \quad \pi^*(s_1) \quad \pi^*(s_2) \quad \pi^*(s_3) \quad \pi^*(s_4) \quad \pi^*(s_5) \]

Key quantities of sample trajectory

- minimum state-action occupancy probability (uniform coverage)

\[ \mu_{\text{min}} := \min \mu_{\pi_b}(s, a) \in [0, \frac{1}{|S||A|}] \]

- mixing time: \( t_{\text{mix}} \)
Q-learning on Markovian samples

\[ Q_{t+1}(s_t, a_t) = (1 - \eta_t)Q_t(s_t, a_t) + \eta_t \mathcal{T}_t(Q_t)(s_t, a_t), \quad t \geq 0 \]

only update \((s_t, a_t)\)-th entry
Q-learning on Markovian samples

\[
Q_{t+1}(s_t, a_t) = (1 - \eta_t)Q_t(s_t, a_t) + \eta_t \tau_t(Q_t)(s_t, a_t), \quad t \geq 0
\]

*only update \((s_t, a_t)\)-th entry*

\[
\tau_t(Q)(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a')
\]
Q-learning on Markovian samples

- **asynchronous**: only a single entry is updated each iteration
Q-learning on Markovian samples

- asynchronous: only a single entry is updated each iteration
- off-policy: target policy $\pi^* \neq$ behavior policy $\pi_b$
Sample complexity of asynchronous Q-learning

**Theorem 5 (Li, Cai, Chen, Wei, Chi ’21)**

For any $0 < \varepsilon \leq \frac{1}{1-\gamma}$, sample complexity of async Q-learning to yield $\|\hat{Q} - Q^*\|_\infty \leq \varepsilon$ with high prob. (or $\mathbb{E}[\|\hat{Q} - Q^*\|_\infty] \leq \varepsilon$) is at most

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

(up to log factor)
Theorem 5 (Li, Cai, Chen, Wei, Chi ’21)

For any $0 < \varepsilon \leq \frac{1}{1 - \gamma}$, sample complexity of async Q-learning to yield $\|\hat{Q} - Q^*\|_\infty \leq \varepsilon$ with high prob. (or $\mathbb{E}[\|\hat{Q} - Q^*\|_\infty] \leq \varepsilon$) is at most

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

<table>
<thead>
<tr>
<th>other papers</th>
<th>sample complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Even-Dar, Mansour ’03</td>
<td>$(t_{\text{cover}})^{\frac{1}{1 - \gamma}} \frac{1}{(1 - \gamma)^4 \varepsilon^2}$</td>
</tr>
<tr>
<td>Even-Dar, Mansour ’03</td>
<td>$(\frac{1}{t_{\text{cover}}^{1+3\omega}})^{\frac{1}{1 - \gamma}} \frac{1}{(1 - \gamma)^4 \varepsilon^2} + (t_{\text{cover}})^{\frac{1}{1 - \gamma}} \frac{1}{1 - \omega}, \omega \in (\frac{1}{2}, 1)$</td>
</tr>
<tr>
<td>Beck &amp; Srikant ’12</td>
<td>$t_{\text{cover}}^3</td>
</tr>
<tr>
<td>Qu &amp; Wierman ’20</td>
<td>$t_{\text{mix}} \frac{t_{\text{mix}}}{\mu_{\min}(1 - \gamma)^5 \varepsilon^2}$</td>
</tr>
<tr>
<td>Li, Wei, Chi, Gu, Chen ’20</td>
<td>$\frac{1}{\mu_{\min}(1 - \gamma)^5 \varepsilon^2} + t_{\text{mix}} \frac{t_{\text{mix}}}{\mu_{\min}(1 - \gamma)}$</td>
</tr>
<tr>
<td>Chen, Maguluri, Shakkottai, Shanmugam ’21</td>
<td>$\frac{1}{\mu_{\min}(1 - \gamma)^5 \varepsilon^2} + \text{other-term}(t_{\text{mix}})$</td>
</tr>
</tbody>
</table>
Linear dependency on $1/\mu_{\text{min}}$

If we take $\mu_{\text{min}} \approx \frac{1}{|S||A|}$, $t_{\text{cover}} \approx \frac{t_{\text{mix}}}{\mu_{\text{min}}}$.
Effect of mixing time on sample complexity

\[ \frac{1}{\mu_{\text{min}} (1 - \gamma)^4 \varepsilon^2} + \frac{t_{\text{mix}}}{\mu_{\text{min}} (1 - \gamma)} \]

- reflects cost taken to reach steady state
Effect of mixing time on sample complexity

\[
\frac{1}{\mu_{\min}(1 - \gamma)^4 \varepsilon^2} + \frac{t_{\text{mix}}}{\mu_{\min}(1 - \gamma)}
\]

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

— prior art: \(\frac{t_{\text{mix}}}{\mu_{\min}^2 (1 - \gamma)^5 \varepsilon^2}\) (Qu & Wierman '20)
Model-free RL

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

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

\[
s \sim \rho^b, \quad a \sim \pi^b(\cdot | s), \quad s' \sim P(\cdot | s, a)
\]

for some state distribution \( \rho^b \) and behavior policy \( \pi^b \)
Recap: offline RL / batch RL

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

$$s \sim \rho^b, \quad a \sim \pi^b(\cdot | s), \quad s' \sim P(\cdot | s, a)$$

for some state distribution $\rho^b$ and behavior policy $\pi^b$

Single-policy concentrability

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

where $d^\pi$: occupancy distribution under $\pi$

- captures distributional shift
- allows for partial coverage
How to design offline model-free algorithms with optimal sample efficiency?
How to design offline model-free algorithms with optimal sample efficiency?

Q-learning $\rightarrow$ LCB-Q $\rightarrow$ LCB-Q-Advantage

pessimism (low confidence bounds) variance reduction

Our algorithms
LCB-Q: Q-learning with LCB penalty

\[ Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) + \eta_t T_t(Q_t)(s_t, a_t) - \eta_t b_t(s_t, a_t) \]

\( Q_{t+1}(s_t, a_t) \) is the updated Q-value.

- **Classical Q-learning**:
  - \((1 - \eta_t)Q_t(s_t, a_t) + \eta_t T_t(Q_t)(s_t, a_t)\)

- **LCB penalty**:
  - \(-\eta_t b_t(s_t, a_t)\)

---

Shi et al. ’22, Yan et al. ’22
LCB-Q: Q-learning with LCB penalty

— Shi et al. ’22, Yan et al. ’22

\[
Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) + \eta_t \mathcal{T}_t(Q_t)(s_t, a_t) - \eta_t b_t(s_t, a_t)
\]

- \(b_t(s, a)\): Hoeffding-style confidence bound
- pessimism in the face of uncertainty
LCB-Q: Q-learning with LCB penalty

\[ Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) + \eta_t T_t(Q_t)(s_t, a_t) - \eta_t b_t(s_t, a_t) \]

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

\[
\tilde{O}\left(\frac{SC^*}{(1-\gamma)^5\varepsilon^2}\right) \quad \Rightarrow \quad \text{sub-optimal by a factor of } \frac{1}{(1-\gamma)^2}
\]

**Issue:** large variability in stochastic update rules
Q-learning with LCB and variance reduction

— Shi et al. '22, Yan et al. '22

\[ Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) - \eta_t b_t(s_t, a_t) \]

\[ + \eta_t \left( T_t(Q_t) - T_t(\hat{Q}) + \hat{T}(\hat{Q}) \right)(s_t, a_t) \]

LCB penalty

advantage

reference

\[ \rho = 1 - \gamma \]

Theorem 6 (Yan, Li, Chen, Fan '22, Shi, Li, Wei, Chen, Chi '22)
Q-learning with LCB and variance reduction

\[ Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) - \eta_t b_t(s_t, a_t) \]

\[ + \eta_t \left( T_t(Q_t) - T_t(\overline{Q}) + \hat{T}(\overline{Q}) \right)(s_t, a_t) \]

- incorporates variance reduction into LCB-Q

For \( \epsilon \) Theorem 6 (Yan, Li, Chen, Fan '22, Shi, Li, Wei, Chen, Chi '22)

\[ O((1 - \gamma)^\star(\rho - \epsilon)) \]

\[ \pi(\epsilon) \]
Q-learning with LCB and variance reduction

— Shi et al.'22, Yan et al.'22

\[ Q_{t+1}(s_t, a_t) \leftarrow (1 - \eta_t)Q_t(s_t, a_t) - \eta_t b_t(s_t, a_t) \]

\[ + \eta_t \left( \mathcal{T}_t(Q_t) - \mathcal{T}_t(\overline{Q}) + \hat{\mathcal{T}(\overline{Q})} \right)(s_t, a_t) \]

- incorporates variance reduction into LCB-Q

---

**Theorem 6** (Yan, Li, Chen, Fan '22, Shi, Li, Wei, Chen, Chi '22)

For \( \varepsilon \in (0, 1 - \gamma] \), LCB-Q-Advantage achieves \( V^*(\rho) - V^{\hat{\pi}}(\rho) \leq \varepsilon \) with optimal sample complexity \( \tilde{O}\left(\frac{SC^*}{(1-\gamma)^3\varepsilon^2}\right) \)
Model-free offline RL attains sample optimality too! — with some burn-in cost though …
Model-free RL

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

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

\[ \{s^1_h, a^1_h, r^1_h\}^{H}_{h=1} \]
Sequentially execute MDP for $K$ episodes, each consisting of $H$ steps.
Online RL: interacting with real environments

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

- **Episode 1**: Execute $\pi^1$ followed by the sequence $\{s_h^1, a_h^1, r_h^1\}_{h=1}^H$
- **Episode 2**: Execute $\pi^2$ followed by the sequence $\{s_h^2, a_h^2, r_h^2\}_{h=1}^H$
- **Episode $K$**: Execute $\pi^K$ followed by the sequence $\{s_h^K, a_h^K, r_h^K\}_{h=1}^H$
Online RL: interacting with real environments

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

--- sample size: $T = KH$

**Exploration** (exploring unknowns) vs. **exploitation** (exploiting learned info)
Regret: gap between learned policy & optimal policy

adversary               learner

initial state \( s_1 \)  \( \rightarrow \)  execute policy \( \pi^1 \)

episode 1
Regret: gap between learned policy & optimal policy

Initial state $s_1^1$ \rightarrow\text{execute policy } \pi^1 \rightarrow \ldots \rightarrow \text{initial state } s_K^1 \rightarrow\text{execute policy } \pi^K

Episode 1

Episode $K$
Regret: gap between learned policy & optimal policy

Performance metric: given initial states $\{s^k_1\}^{K}_{k=1}$, define

\[
\text{Regret}(T) := \sum_{k=1}^{K} \left( V_1^*(s^k_1) - V_1^{\pi_k}(s^k_1) \right)
\]
Lower bound
(Domingues et al. ’21)
\[ \text{Regret}(T) \gtrsim \sqrt{H^2 \text{SAT}} \]

Existing algorithms

- UCB-VI: Azar et al. ’17
- UBEV: Dann et al. ’17
- UCB-Q-Hoeffding: Jin et al. ’18
- **UCB-Q-Bernstein**: Jin et al. ’18
- UCB2-Q-Bernstein: Bai et al. ’19
- EULER: Zanette et al. ’19
- UCB-Q-Advantage: Zhang et al. ’20
- UCB-M-Q: Menard et al. ’21
- Q-EarlySettled-Advantage: Li et al. ’21
Which model-free algorithms are sample-efficient for online RL?
Which model-free algorithms are sample-efficient for online RL?

- Q-learning
- UCB-Q
- UCB-Q-Advantage
- Q-EarlySettled-Advantage

UCB exploration
variance reduction
early-settled variance reduction

Jin et al. '18
Zhang et al. '20
Li et al. '21
Q-learning with UCB exploration (Jin et al., 2018)

\[ Q_h(s_h, a_h) \leftarrow (1 - \eta_k)Q_h(s_h, a_h) + \eta_k T_k(Q_{h+1})(s_h, a_h) + \eta_k b_h(s_h, a_h) \]

- classical Q-learning
- exploration bonus

Issue: large variability in stochastic update rules
Q-learning with UCB exploration (Jin et al., 2018)

\[ Q_h(s_h, a_h) \leftarrow (1 - \eta_k)Q_h(s_h, a_h) + \eta_k T_k(Q_{h+1})(s_h, a_h) + \eta_k b_h(s_h, a_h) \]

- \( b_h(s, a) \): upper confidence bound; encourage exploration — *optimism in the face of uncertainty*
- inspired by UCB bandit algorithm (Lai, Robbins ’85)
Q-learning with UCB exploration (Jin et al., 2018)

\[ Q_h(s_h, a_h) \leftarrow (1 - \eta_k)Q_h(s_h, a_h) + \eta_k T_k (Q_{h+1}) (s_h, a_h) + \eta_k b_h(s_h, a_h) \]

- \( b_h(s, a) \): upper confidence bound; encourage exploration
  — _optimism in the face of uncertainty_
- inspired by UCB bandit algorithm (Lai, Robbins’85)

\[ \text{Regret}(T) \lesssim \sqrt{H^3 SAT} \quad \Longrightarrow \quad \text{sub-optimal by a factor of } \sqrt{H} \]
Q-learning with UCB exploration (Jin et al., 2018)

\[ Q_h(s_h, a_h) \leftarrow (1 - \eta_k)Q_h(s_h, a_h) + \eta_k T_k(Q_{h+1})(s_h, a_h) + \eta_k b_h(s_h, a_h) \]

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

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

**Issue:** large variability in stochastic update rules
UCB Q-learning with UCB and variance reduction

Incorporates \textit{variance reduction} into UCB-Q: \hfill — Zhang, Zhou, Ji ’20

- asymptotically regret-optimal
UCB Q-learning with UCB and variance reduction

Incorporates variance reduction into UCB-Q: — Zhang, Zhou, Ji ’20

• asymptotically regret-optimal

• **Issue:** high burn-in cost $O(S^6 A^4 H^{28})$
UCB Q-learning with UCB and variance reduction

Incorporates variance reduction into UCB-Q: — Zhang, Zhou, Ji ’20

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

One additional idea: early settlement of reference updates — Li, Shi, Chen, Chi ’23
UCB Q-learning with UCB and variance reduction

Incorporates variance reduction into UCB-Q: — Zhang, Zhou, Ji ’20

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

One additional idea: early settlement of reference updates — Li, Shi, Chen, Chi ’23

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

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

— with some burn-in cost though
Reference I

- "When can we learn general-sum Markov games with a large number of players sample-efficiently?" Z. Song, S. Mei, Y. Bai, ICLR 2022
- "Minimax-optimal multi-agent RL in markov games with a generative model," G. Li, Y. Chi, Y. Wei, Y. Chen, NeurIPS, 2022
- "The complexity of Markov equilibrium in stochastic games,” C. Daskalakis, N. Golowich, K. Zhang, COLT, 2023

"Learning from delayed rewards," C. Watkins, 1989


Reference III


- "Is Q-Learning minimax optimal? A tight sample complexity analysis," G. Li, Y. Wei, Y. Chi, Y. Chen, accepted to Operations Research, 2023


Reference IV


Non-asymptotic Analysis for Reinforcement Learning (Part 3)

Yuejie Chi

Carnegie Mellon University

Sigmetrics Tutorial
June 2023
A triad of RL approaches

— Figure credit: D. Silver
Policy optimization in practice

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

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

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

Our goal:

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

- **Backgrounds and basics**
  - policy gradient method

- **Convergence guarantees of single-agent policy optimization**
  - (natural) policy gradient methods
  - finite-time rate of global convergence
  - entropy regularization and beyond

- **Multi-agent policy optimization: two-player zero-sum games**
  - Matrix game
  - Markov game

- **Concluding remarks and further pointers**
Backgrounds: policy optimization in tabular Markov decision processes
Searching for the optimal policy

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

- optimal value/Q function: $V^* := V^{\pi^*}$, $Q^* := Q^{\pi^*}$
Policy gradient methods

Given an initial state distribution \( s \sim \rho \), find policy \( \pi \) such that

\[
\maximize_{\pi} \quad V^{\pi}(\rho) := \mathbb{E}_{s \sim \rho}[V^{\pi}(s)]
\]

Parameterization:

\[
\pi := \pi_{\theta}
\]

\[
\maximize_{\theta} \quad V^{\pi_{\theta}}(\rho) := \mathbb{E}_{s \sim \rho}[V^{\pi_{\theta}}(s)]
\]

Policy gradient method (Sutton et al., 2000)

For \( t = 0, 1, \cdots \)

\[
\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_{\theta} V^{\pi^{(t)}_{\theta}}(\rho)
\]

where \( \eta \) is the learning rate.
Softmax policy gradient methods

Given an initial state distribution \( s \sim \rho \), find policy \( \pi \) such that

\[
\text{maximize}_{\pi} \quad V^{\pi}(\rho) := \mathbb{E}_{s \sim \rho} [V^{\pi}(s)]
\]

softmax parameterization:

\[
\pi_{\theta}(a|s) \propto \exp(\theta(s, a))
\]

\[
\text{maximize}_{\theta} \quad V^{\pi_{\theta}}(\rho) := \mathbb{E}_{s \sim \rho} [V^{\pi_{\theta}}(s)]
\]

Policy gradient method (Sutton et al., 2000)

For \( t = 0, 1, \ldots \)

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

where \( \eta \) is the learning rate.
Finite-time global convergence guarantees
Global convergence of the PG method?

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

• (Mei et al., 2020) Softmax PG converges to global opt in

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

Is the rate of PG good, bad or ugly?
A negative message

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

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

\[
\frac{1}{\eta} \left| S \right|^{2 \Theta \left( \frac{1}{1-\gamma} \right)} \text{ iterations}
\]

to achieve \( \| V(t) - V^* \|_{\infty} \leq 0.15 \).

- Softmax PG can take (super)-exponential time to converge (in problems w/ large state space & long effective horizon)!
- Also hold for average sub-opt gap \( \frac{1}{|S|} \sum_{s \in S} [V(t)(s) - V^*(s)] \).
Key ingredients: for $3 \leq s \leq H \approx \frac{1}{1-\gamma}$,

- $\pi(t)(a_{opt} | s)$ keeps decreasing until $\pi(t)(a_{opt} | s - 2) \approx 1$
What is happening in our constructed MDP?

Convergence time for state $s$ grows geometrically as $s$ increases

$$\text{convergence-time}(s) \gtrsim (\text{convergence-time}(s - 2))^{1.5}$$
“Seriously, lady, at this hour you’d make a lot better time taking the subway.”
Booster #1: natural policy gradient

Natural policy gradient (NPG) method (Kakade, 2002)

For $t = 0, 1, \ldots$

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

where $\eta$ is the learning rate and $\mathcal{F}_\rho^\theta$ is the Fisher information matrix:

$$\mathcal{F}_\rho^\theta := \mathbb{E} \left[ (\nabla_\theta \log \pi_\theta(a|s)) (\nabla_\theta \log \pi_\theta(a|s))^\top \right].$$
TRPO/PPO (Schulman et al., 2015; 2017) are popular heuristics in training RL algorithms, with KL regularization

\[
\text{KL}(\pi^{(t)}_\theta \| \pi_\theta) \approx \frac{1}{2} (\theta - \theta^{(t)})^\top \mathcal{F}_\rho^\theta (\theta - \theta^{(t)})
\]

via constrained or proximal terms:

\[
\theta^{(t+1)} = \arg\max_{\theta} V^{\pi^{(t)}_\theta} (\rho) + (\theta - \theta^{(t)})^\top \nabla_\theta V^{\pi^{(t)}_\theta} (\rho) - \eta \text{KL}(\pi^{(t)}_\theta \| \pi_\theta)
\]

\[
\approx \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi^{(t)}_\theta} (\rho),
\]

leading to exactly NPG!

NPG \approx \text{TRPO/PPO!}
NPG in the tabular setting

Natural policy gradient (NPG) method (Tabular setting)

For $t = 0, 1, \cdots$, NPG updates the policy via

$$
\pi^{(t+1)}(\cdot|s) \propto \pi^{(t)}(\cdot|s) \exp \left( \eta Q^{(t)}(s, \cdot) \left/ \frac{1 - \gamma}{1 - \gamma} \right. \right)
$$

where $Q^{(t)} := Q^{\pi^{(t)}}$ is the Q-function of $\pi^{(t)}$, and $\eta > 0$.

- invariant with the choice of $\rho$
- Reduces to policy iteration (PI) when $\eta = \infty$. 

Global convergence of NPG

**Theorem (Agarwal et al., 2019)**

Set $\pi^{(0)}$ as a uniform policy. For all $t \geq 0$, we have

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

**Implication:** set $\eta \geq (1 - \gamma)^2 \log |A|$, we find an $\epsilon$-optimal policy within at most

$$\frac{2}{(1 - \gamma)^2 \epsilon}$$

iterations.

Global convergence at a sublinear rate independent of $|S|, |A|$!
Booster #2: entropy regularization

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

\[
\forall s \in S : \quad V^\pi_\tau(s) := \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t (r_t + \tau \mathcal{H}(\pi(\cdot|s_t))) \mid s_0 = s \right]
\]

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

\[
\text{maximize}_{\theta} \quad V^{{\pi}_{\theta}}_\tau(\rho) := \mathbb{E}_{s \sim \rho} [V^{{\pi}_{\theta}}_\tau(s)]
\]
Entropy-regularized natural gradient helps!

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

Can we justify the efficacy of entropy-regularized NPG?

<table>
<thead>
<tr>
<th>Policy Gradient</th>
<th>Natural Policy Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Policy Gradient" /></td>
<td><img src="image2" alt="Natural Policy Gradient" /></td>
</tr>
</tbody>
</table>

Increase regularization
Entropy-regularized NPG in the tabular setting

For $t = 0, 1, \cdots$, the policy is updated via

$$
\pi^{(t+1)}(\cdot|s) \propto \pi^{(t)}(\cdot|s)^{1 - \frac{\eta \tau}{1 - \gamma}} \exp\left(Q^{(t)}_\tau(s, \cdot)/\tau\right)^{\frac{\eta \tau}{1 - \gamma}}
$$

where $Q^{(t)}_\tau := Q^{\pi^{(t)}}_\tau$ is the soft $Q$-function of $\pi^{(t)}$, and $0 < \eta \leq \frac{1 - \gamma}{\tau}$.

- invariant with the choice of $\rho$
- Reduces to soft policy iteration (SPI) when $\eta = \frac{1 - \gamma}{\tau}$. 
Linear convergence with exact gradient

**Exact oracle:** perfect evaluation of $Q^\pi_\tau(t)$ given $\pi(t)$;

—Read the paper for the inexact case

---

**Theorem (Cen, Cheng, Chen, Wei, Chi, 2020)**

For any learning rate $0 < \eta \leq (1 - \gamma)/\tau$, the entropy-regularized NPG updates satisfy

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

$$
\|Q^\star_\tau - Q^{(t+1)}_\tau\|_\infty \leq C_1 \gamma (1 - \eta \tau)^t
$$

for all $t \geq 0$, where $Q^\star_\tau$ is the optimal soft Q-function, and

$$
C_1 = \|Q^\star_\tau - Q^{(0)}_\tau\|_\infty + 2\tau \left(1 - \frac{\eta \tau}{1 - \gamma}\right) \|\log \pi^\star_\tau - \log \pi^{(0)}_\tau\|_\infty.
$$
Implications

To reach \( \| Q^*_\tau - Q^{(t+1)}_\tau \|_\infty \leq \epsilon \), the iteration complexity is at most

- **General learning rates** \((0 < \eta < \frac{1-\gamma}{\tau})\):

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

- **Soft policy iteration** \((\eta = \frac{1-\gamma}{\tau})\):

\[
\frac{1}{1-\gamma} \log \left( \frac{\| Q^*_\tau - Q^{(0)}_\tau \|_\infty}{\epsilon} \right)
\]

Global linear convergence of entropy-regularized NPG at a rate independent of \(|S|, |A|\)!
Comparisons with entropy-regularized PG

(Mei et al., 2020) showed entropy-regularized PG achieves

\[ V^*_\tau(\rho) - V^{(t)}_\tau(\rho) \leq \left( V^*_\tau(\rho) - V^{(0)}_\tau(\rho) \right) \]

\[ \cdot \exp \left( -\frac{(1 - \gamma)^4 t}{(8/\tau + 4 + 8 \log |A|)|S|} \left\| d_{\rho}^{\pi^*_\tau} \right\|^{-1}_\infty \min_{s} \rho(s) \left( \inf_{0 \leq k \leq t-1} \min_{s,a} \pi^{(k)}(a|s) \right)^2 \right) \]

can be exponential in |S| and \( \frac{1}{1 - \gamma} \)

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

Regularized NPG
\( \tau = 0.001 \)

Vanilla NPG
\( \tau = 0 \)

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

Sublinear rate: \( \frac{1}{\min\{\eta, (1-\gamma)^2\} \epsilon} \)
(Agarwal et al. 2019)

Entropy regularization enables fast convergence!
A key operator: soft Bellman operator

**Soft Bellman operator**

\[ T_\tau(Q)(s, a) := r(s, a) \]

immediate reward

\[ + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} \left[ \max_{a' \sim \pi(\cdot|s')} \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[ Q(s', a') - \tau \log \pi(a'|s') \right] \right], \]

next state’s value

entropy

**Soft Bellman equation:** \( Q_\tau^* \) is unique solution to

\[ T_\tau(Q_\tau^*) = Q_\tau^* \]

\( \gamma \)-contraction of soft Bellman operator:

\[ \| T_\tau(Q_1) - T_\tau(Q_2) \|_\infty \leq \gamma \| Q_1 - Q_2 \|_\infty \]
Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

Policy iteration

Soft policy iteration

Bellman operator

Soft Bellman operator
A key linear system: general learning rates

Let

\[ x_t := \begin{bmatrix} \|Q^* - Q^{(t)}_\tau\|_\infty \\ \|Q^*_\tau - \tau \log \xi^{(t)}_\tau\|_\infty \end{bmatrix} \]

and \( y := \begin{bmatrix} \|Q^{(0)}_\tau - \tau \log \xi^{(0)}_\tau\|_\infty \\ 0 \end{bmatrix} \),

where \( \xi^{(t)} \propto \pi^{(t)} \) is an auxiliary sequence, then

\[ x_{t+1} \leq Ax_t + \gamma \left( 1 - \frac{\eta \tau}{1 - \gamma} \right)^{t+1} y, \]

where

\[ A := \begin{bmatrix} \gamma \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\eta \tau}{1-\gamma} & 1 - \frac{\eta \tau}{1-\gamma} \end{bmatrix} \]

is a rank-1 matrix with a non-zero eigenvalue \( 1 - \eta \tau \) contract rate!
Beyond entropy regularization

Leverage regularization to promote structural properties of the learned policy.

- **cost-sensitive RL**
  - weighted 1-norm
- **sparse exploration**
  - Tsallis entropy
- **constrained and safe RL**
  - log-barrier

For further details, see: (Lan, PMD 2021) and (Zhan et al, GPMD 2021)
Policy optimization for games
Zero-sum two-player Markov game

Given an initial state distribution $s \sim \rho$, find policy $\pi$ such that

$$\max_{\mu \in \Delta(A)} \min_{\nu \in \Delta(B)} V^{\mu,\nu}(\rho) := \mathbb{E}_{s \sim \rho}[V^{\mu,\nu}(s)]$$

Can we design a policy optimization method that guarantees fast last-iterate convergence?
Entropy regularization in MARL

Promote the stochasticity of the policy pair using the “soft” value function \((\text{Williams and Peng, 1991; Cen et al., 2020})\):

\[
V_{\mu,\nu}^\tau(s) := \mathbb{E} \left[ \sum_{h=1}^{H} (r_h + \tau \mathcal{H}(\mu_h(\cdot|s_h)) - \tau \mathcal{H}(\nu_h(\cdot|s_h))) \right| s_0 = s
\]

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

\[
\max_{\mu \in \Delta(A)} \min_{\nu \in \Delta(B)} V_{\mu,\nu}^\tau(\rho)
\]
The quantal response equilibrium (QRE) is the policy pair \((\mu^*_\tau, \nu^*_\tau)\) that is the unique solution to

\[
\max_{\mu \in \Delta(A)^{|S|}} \min_{\nu \in \Delta(B)^{|S|}} V_{\tau}^{\mu, \nu}(\rho).
\]

- Unlike NE, QRE assumes bounded rationality: action probability follows the logit function.

Translating to an \(\epsilon\)-NE: setting \(\tau \asymp \tilde{O}(\epsilon/H)\).
**Soft value iteration**: for \( h = H, \ldots, 1 \)

\[
Q_h(s, a, b) \leftarrow r_h(s, a, b) + \\
\cdot \mathbb{E}_{s' \sim P_h(\cdot|s,a,b)} \left[ \max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s')\nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s')) \right],
\]

where \( Q_h(s) = [Q_h(s, \cdot, \cdot)] \in \mathbb{R}^{A \times B} \).

**Entropy-regularized matrix game**

\[
\max_{\mu \in \Delta(A)} \min_{\nu \in \Delta(B)} \mu^\top A\nu + \tau \mathcal{H}(\mu) - \tau \mathcal{H}(\nu)
\]
Failure of NPG/MWU methods

$$\max_{\mu \in \Delta(A)} \min_{\nu \in \Delta(B)} f_{\tau}(\mu, \nu) := \mu^\top A\nu + \tau \mathcal{H}(\mu) - \tau \mathcal{H}(\nu)$$

- Multiplicative Weights Update (MWU):
  $$\begin{align*}
  \mu^{(t+1)}(a) &\propto \mu^{(t)}(a)^{1-\eta\tau} \exp \left( \eta [A\nu^{(t)}]_a \right) \\
  \nu^{(t+1)}(b) &\propto \nu^{(t)}(b)^{1-\eta\tau} \exp \left( -\eta [A^\top \mu^{(t)}]_b \right)
  \end{align*}$$

- $\eta > 0$: step size;

- The trajectory may cycle/diverge!
Motivation: an implicit update method

Implicit update (IU) method

For \( t = 0, 1, \cdots \),

\[
\begin{align*}
\mu^{(t+1)} &\propto [\mu^{(t)}]^{1-\eta \tau} \exp \left( \frac{[A \nu^{(t+1)}]/\tau}{\eta \tau} \right) \\
\nu^{(t+1)} &\propto [\nu^{(t)}]^{1-\eta \tau} \exp \left( -\frac{[A^\top \mu^{(t+1)}]/\tau}{\eta \tau} \right)
\end{align*}
\]

Theorem (Cen, Wei, Chi, 2021)

Suppose that \( 0 < \eta \leq 1/\tau \), then for all \( t \geq 0 \),

\[
\text{KL} (\hat{\zeta}^*_{\tau} \| \hat{\zeta}^{(t)}) \leq (1 - \eta \tau)^t \text{KL} (\hat{\zeta}^*_{\tau} \| \hat{\zeta}^{(0)}) ,
\]

where \( \text{KL} (\hat{\zeta}^*_{\tau} \| \hat{\zeta}^{(t)}) = \text{KL} (\hat{\mu}^*_{\tau} \| \mu^{(t)}) + \text{KL} (\hat{\nu}^*_{\tau} \| \nu^{(t)}) \).

Can we make this practical?
From implicit updates to policy extragradient methods

Optimistic multiplicative weights update (OMWU) method
(Related to OMD, Rakhlin and Sridharan, 2013): for \( t = 0, 1, \ldots \),

predict:

\[
\begin{align*}
\bar{\mu}(t+1) &\propto [\mu(t)]^{1-\eta \tau} \exp \left( \frac{[A\bar{\nu}(t)]}{\tau} \right)^{\eta \tau} \\
\bar{\nu}(t+1) &\propto [\nu(t)]^{1-\eta \tau} \exp \left( -\frac{[A^\top \bar{\mu}(t)]}{\tau} \right)^{\eta \tau}
\end{align*}
\]

update:

\[
\begin{align*}
\mu(t+1) &\propto [\mu(t)]^{1-\eta \tau} \exp \left( \frac{[A\bar{\nu}(t+1)]}{\tau} \right)^{\eta \tau} \\
\nu(t+1) &\propto [\nu(t)]^{1-\eta \tau} \exp \left( -\frac{[A^\top \bar{\mu}(t+1)]}{\tau} \right)^{\eta \tau}
\end{align*}
\]

Theorem (Cen, Wei, Chi, 2021)

Suppose that \( \eta \leq \min \left\{ \frac{1}{2\tau+2\|A\|_\infty}, \frac{1}{4\|A\|_\infty} \right\} \), then for all \( t \geq 0 \), the last-iterate converges to \( \epsilon \)-QRE within \( \tilde{O} \left( \frac{1}{\eta \tau} \log \frac{1}{\epsilon} \right) \) iterations.

Linear, last-iterate convergence to the QRE!
Soft value iteration via nested-loop OMWU

**Soft value iteration:** for $h = H, \ldots, 1$

$$Q_h(s, a, b) \leftarrow r_h(s, a, b) +$$

$$\cdot \mathbb{E}_{s' \sim P_h(\cdot | s, a, b)} \left[ \max_{\mu(s')} \min_{\nu(s')} \mu(s')^\top Q_{h+1}(s') \nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s')) \right] ,$$

where $Q_h(s) = [Q_h(s, \cdot, \cdot)] \in \mathbb{R}^{A \times B}$.

**Nested-loop approach:**

$(\mu_h(t), \nu_h(t)) \leftarrow \text{OMWU}(Q_h)$

However, not easy to use in online settings...
A two-timescale single-loop approach?

**Soft value iteration:** for \( h = H, \ldots, 1 \)

\[
Q_h(s, a, b) \leftarrow r_h(s, a, b) + \\
\cdot \mathbb{E}_{s' \sim P_h(\cdot|s,a,b)}\left[ \max_{\mu} \min_{\nu} \mu(s')^\top Q_{h+1}(s')\nu(s') + \tau \mathcal{H}(\mu(s')) - \tau \mathcal{H}(\nu(s')) \right],
\]

where \( Q_h(s) = [Q_h(s, \cdot, \cdot)] \in \mathbb{R}^{A \times B} \).

**Single-loop, two-timescale approach:**

- **Smooth value update**
  \( Q^{(t+1)} \leftarrow (1 - \alpha)Q^{(t)} + \alpha \cdot \text{lookahead} \)

- **Policy update via OMWU**
  \( (\mu^{(t+1)}, \nu^{(t+1)}) \leftarrow \text{OMWU}(Q^{(t)}) \)
Main result: episodic setting

**Theorem (Cen, Chi, Du, Xiao, 2022)**

The last-iterate of the two-timescale single-loop algorithm finds an $\epsilon$-QRE in

$$\tilde{O}\left(\frac{H^2}{\tau} \log \frac{1}{\epsilon}\right)$$

iterations, corresponding to $\tilde{O}\left(\frac{H^3}{\epsilon}\right)$ iterations for finding an $\epsilon$-NE.

- First last-iterate convergence result for the episodic setting.
- **Almost dimension-free**: independent of the size of the state-action space.
Main result: discounted setting

**Theorem (Cen, Chi, Du, Xiao, 2022)**

For the infinite-horizon $\gamma$-discounted setting, the last-iterate of the single-loop algorithm finds an $\epsilon$-QRE in

$$\tilde{O} \left( \frac{S}{(1 - \gamma)^4 \tau} \log \frac{1}{\epsilon} \right)$$

iterations, and in $\tilde{O} \left( \frac{S}{(1 - \gamma)^5 \epsilon} \right)$ iterations for finding an $\epsilon$-NE.

- This significantly improves upon the prior art $\tilde{O} \left( \frac{S^5 (A+B)^{1/2}}{(1-\gamma)^{16} c^4 \epsilon^2} \right)$ of (Wei et al., 2021) and $\tilde{O} \left( \frac{S^2 \| 1/\rho \|_5^5}{(1-\gamma)^{14} c^4 \epsilon^3} \right)$ of (Zeng et al., 2022) in all parameter dependencies.
Concluding Remarks
Concluding remarks

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

Promising directions:

• function approximation
• multi-agent/federated RL
• hybrid RL
• many more...
Beyond the tabular setting

Figure credit: (Silver et al., 2016)

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

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

- **Competitive setting:** finding Nash equilibria for Markov games

- **Collaborative setting:** multiple agents jointly optimize the policy to maximize the total reward

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

**Online RL**
- interact with environment
- actively collect new data

**Offline/Batch RL**
- no interaction
- data is given

Can we achieve the best of both worlds?
(Wagenmaker and Pacchiano, 2022; Song et al., 2022; Li et al., 2023)
Federated reinforcement learning enables multiple agents to collaboratively learn a global model without sharing datasets.

Can we achieve linear speedup via federated learning?
(Khodadadian et al., 2022; Woo et al., 2023)
Disclaimer: this straw-man list is by no means exhaustive (in fact, it is quite the opposite given the fast pace of the field), and biased towards materials most related to this tutorial; readers are invited to further delve into the references therein to gain a more complete picture.

Books and monographs:

Policy optimization:


Bibliography III


**Additional ad-hoc pointers:**


Thanks!

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