Distributionally Robust RL with Interactive Data Collection

Fundamental Hardness and Near-Optimal Algorithm

Miao Lu

Stanford University

INFORMS 2024

Joint work with

Han Zhong
PKU

Jose Blanchet
Stanford

Tong Zhang
UIUC

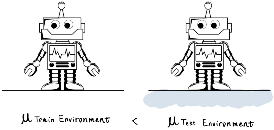
Lu, M., Zhong, H., Zhang, T., & Blanchet, J. (2024). Distributionally Robust Reinforcement Learning with Interactive Data Collection: Fundamental Hardness and Near-Optimal Algorithm. *Preliminary version at 37th Conference on Neural Information Processing Systems (NeurIPS), 2024*

Work supported by grants NSF 2118199, 2229012, 2312204 and FA9550-20-1-0397.

Outline

RL and Sim-to-Real Gaps

- RL achieves tremenduous success in:
  - Robotics
  - Healthcare
  - Recommendation systems
  - Autonomous driving
  - Training large language models
- Challenge: sim-to-real gap
  - Training env. $\neq$ Testing env.
  - Cause degeneration of performance

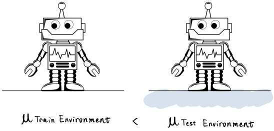


Basics of RL

- MDP: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{R_h\}_{h=1}^H)$.
- $\mathcal{S}$: state space, $\mathcal{A}$: action space.
- $R : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$: reward function
- H: Horizon length.
- $P_h^\star$: transition distribution of training Env.
- P_h': transition distribution of testing Env.

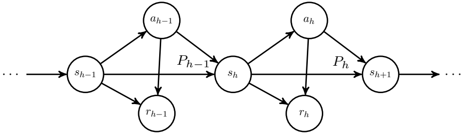
RL and Sim-to-Real Gaps

- RL achieves tremenduous success in:
  - Robotics
  - Healthcare
  - Recommendation systems
  - Autonomous driving
  - Training large language models
- Challenge: sim-to-real gap
  - Training env. $\neq$ Testing env.
  - Cause degeneration of performance



Basics of RL

- MDP: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{R_h\}_{h=1}^H)$.
- $\mathcal{S}$: state space, $\mathcal{A}$: action space.
- $R : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$: reward function
- H: Horizon length.
- $P_h^\star$: transition distribution of training Env.
- P'_h: transition distribution of testing Env.



Robust Markov Decision Processes

- Goal: using data collected from training Env. $P^\star = \{P_h^\star\}_{h=1}^H$, find a policy $\pi^\star$ that

$$\pi^\star := \arg\sup_{\pi\in\Pi} V_{1,P^\star,\mathbf{\Phi}}^{\pi}(s) := \arg\sup_{\pi\in\Pi} \inf_{\substack{P_h\in\mathbf{\Phi}(P_h^\star)\\ 1\le h\le H}} \mathbb{E}_P^\pi\left[\sum_{h=1}^H R_h(s_h,a_h)\middle| s_1 = s\right].$$

- Robust set Φ: set of testing environment distributions
  - This work: we focus on Total Variation (TV) distance robust set:

$$\Phi(P) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_\rho(s,a;P),$$

$$\mathcal{P}_\rho(s,a;P) := \left\{\widetilde{P}(\cdot)\in\Delta(\mathcal{S}) : D_{\mathrm{TV}}\left(\widetilde{P}(\cdot)\middle\|P(\cdot|s,a)\right)\le\rho\right\}.$$

  - $\rho\in[0,1)$ is the level of robustness.
- Robust Bellman optimal equation:

$$V_{h,P^\star,\Phi}^\star(s) = \max_{a\in\mathcal{A}} Q_{h,P^\star,\Phi}^\star(s,a), \quad Q_{h,P^\star,\Phi}^\star(s,a) = R_h(s,a) + \inf_{P_h\in\Phi(P_h^\star)} \mathbb{E}_P^\pi\left[V_{h+1,P^\star,\Phi}^\star\right].$$

Robust Markov Decision Processes

- Goal: using data collected from training Env. $P^\star = \{P_h^\star\}_{h=1}^H$, find a policy $\pi^\star$ that

$$\pi^\star := \arg\sup_{\pi \in \Pi} V_{1,P^\star,\Phi}^{\pi}(s) := \arg\sup_{\pi \in \Pi} \inf_{\substack{P_h \in \Phi(P_h^\star) \\ 1 \le h \le H}} \mathbb{E}_P^\pi \left[\sum_{h=1}^H R_h(s_h, a_h) \middle| s_1 = s \right].$$

- Robust set $\mathbf{\Phi}$: set of testing environment distributions
  - This work: we focus on Total Variation (TV) distance robust set:

$$\mathbf{\Phi}(P) = \bigotimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_\rho(s,a;P),$$

$$\mathcal{P}_\rho(s,a;P) := \left\{ \widetilde{P}(\cdot) \in \Delta(\mathcal{S}) : D_{\mathrm{TV}}\left(\widetilde{P}(\cdot) \middle\| P(\cdot|s,a)\right) \le \rho \right\}.$$

  - $\rho \in [0,1)$ is the level of robustness.

- Robust Bellman optimal equation:

$$V_{h,P^\star,\Phi}^\star(s) = \max_{a \in \mathcal{A}} Q_{h,P^\star,\Phi}^\star(s,a), \quad Q_{h,P^\star,\Phi}^\star(s,a) = R_h(s,a) + \inf_{P_h \in \Phi(P_h^\star)} \mathbb{E}_P^\pi\left[V_{h+1,P^\star,\Phi}^\star\right].$$

Robust Markov Decision Processes

- Goal: using data collected from training Env. $P^\star = \{P_h^\star\}_{h=1}^H$, find a policy $\pi^\star$ that

$$\pi^\star := \arg\sup_{\pi\in\Pi} V^\pi_{1,P^\star,\mathbf{\Phi}}(s) := \arg\sup_{\pi\in\Pi} \inf_{\substack{P_h\in\mathbf{\Phi}(P_h^\star)\\ 1\le h\le H}} \mathbb{E}_P^\pi\left[\sum_{h=1}^H R_h(s_h,a_h)\middle| s_1 = s\right].$$

- Robust set $\mathbf{\Phi}$: set of testing environment distributions
  - This work: we focus on Total Variation (TV) distance robust set:

$$\mathbf{\Phi}(P) = \bigotimes_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{P}_\rho(s,a;P),$$

$$\mathcal{P}_\rho(s,a;P) := \left\{\widetilde{P}(\cdot)\in\Delta(\mathcal{S}) : D_{\mathrm{TV}}\left(\widetilde{P}(\cdot)\middle\| P(\cdot|s,a)\right)\le\rho\right\}.$$

  - $\rho\in[0,1)$ is the level of robustness.
- Robust Bellman optimal equation:

$$V^\star_{h,P^\star,\mathbf{\Phi}}(s) = \max_{a\in\mathcal{A}} Q^\star_{h,P^\star,\mathbf{\Phi}}(s,a), \quad Q^\star_{h,P^\star,\mathbf{\Phi}}(s,a) = R_h(s,a) + \inf_{P_h\in\mathbf{\Phi}(P_h^\star)} \mathbb{E}_P^\pi\left[V^\star_{h+1,P^\star,\mathbf{\Phi}}\right].$$

Recap: *Learn the optimal robust policy* $\pi^\star$ *using data of training Env.* $P^\star$

What kind of data do we have?

	Mechanism	Explanation
Prior Work	Generative Model	can query any (s, a, h) to obtain $s' \sim P_h^\star(\cdot \mid s, a)$
	Offline Learning	pre-collected data $\{(s_h^i, a_h^i, s_{h+1}^i)\}_{i=1}^N \sim \mu_h(s, a) \otimes P_h^\star(s' \mid s, a)$

This work	Interactive data collection	*interact with $P^\star$ by algorithm-dependent policy!*

	Mechanism	Explanation
Prior Work	Generative Model	can query any (s, a, h) to obtain $s' \sim P_h^\star(\cdot\|s, a)$
	Offline Learning	pre-collected data $\{(s_h^i, a_h^i, s_{h+1}^i)\}_{i=1}^N \sim \mu_h(s, a) \otimes P_h^\star(s'\|s, a)$

This work	Interactive data collection	*interact with $P^\star$ by algorithm-dependent policy!*

Data Generation Mechanism

Interaction protocol:

- Interact with training Env. $P^\star$ for $K \in [N]$ episodes.
- In each episode $k \in [K]$, use policy π^k to collect data $\{(s_h^k, a_h^k, r_h^k, s_{h+1}^k)\}_{h=1}^H$.
- After episode k, use historical data to update policy to π^{k+1}

Metric:

- Online regret:

$$\mathrm{Regret}_\Phi(K) := \sum_{k=1}^{K} V_{1,P^\star,\Phi}^\star(s_1) - V_{1,P^\star,\Phi}^{\pi^k}(s_1).$$

- Sample complexity: # episodes of interaction suffice to learn ε-optimal robust policy, i.e.

$$V_{1,P^\star,\Phi}^\star(s_1) - V_{1,P^\star,\Phi}^{\widehat{\pi}}(s_1) \le \varepsilon \tag{1}$$

Question:

"Can we design a provably sample-efficient robust RL algorithm using interactive data collection in the training environment?"

Data Generation Mechanism

Interaction protocol:

- Interact with training Env. $P^\star$ for $K \in [N]$ episodes.
- In each episode $k \in [K]$, use policy π^k to collect data $\left\{(s_h^k, a_h^k, r_h^k, s_{h+1}^k)\right\}_{h=1}^H$.
- After episode k, use historical data to update policy to π^{k+1}

Metric:

- Online regret:

$$\mathrm{Regret}_{\mathbf{\Phi}}(K) := \sum_{k=1}^{K} V_{1,P^\star,\mathbf{\Phi}}^{\star}(s_1) - V_{1,P^\star,\mathbf{\Phi}}^{\pi^k}(s_1).$$

- Sample complexity: # episodes of interaction suffice to learn ε-optimal robust policy, i.e.

$$V_{1,P^\star,\mathbf{\Phi}}^{\star}(s_1) - V_{1,P^\star,\mathbf{\Phi}}^{\widehat{\pi}}(s_1) \leq \varepsilon \tag{1}$$

Question:

"Can we design a provably sample-efficient robust RL algorithm using interactive data collection in the training environment?"

Data Generation Mechanism

Interaction protocol:

- Interact with training Env. $P^\star$ for $K \in [N]$ episodes.
- In each episode $k \in [K]$, use policy π^k to collect data $\{(s_h^k, a_h^k, r_h^k, s_{h+1}^k)\}_{h=1}^H$.
- After episode k, use historical data to update policy to π^{k+1}

Metric:

- Online regret:

$$\mathrm{Regret}_{\mathbf{\Phi}}(K) := \sum_{k=1}^{K} V_{1,P^\star,\mathbf{\Phi}}^{\star}(s_1) - V_{1,P^\star,\mathbf{\Phi}}^{\pi^k}(s_1).$$

- Sample complexity: # episodes of interaction suffice to learn ε-optimal robust policy, i.e.

$$V_{1,P^\star,\mathbf{\Phi}}^{\star}(s_1) - V_{1,P^\star,\mathbf{\Phi}}^{\widehat{\pi}}(s_1) \leq \varepsilon \quad (1)$$

Question:

"Can we design a provably sample-efficient robust RL algorithm using interactive data collection in the training environment?"

Outline

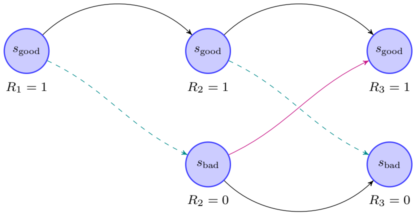
Hardness Result

- For RMDP with total variation (TV) robust set, robust RL with interactive data collection inevitably incurs a regret of $\mathrm{Regret}(K) \geq \Omega(\rho \cdot HK)$, where $\rho =$ size of robust set.
- Hard example: a simple class of two RMDPs: $\mathcal{S} = \{s_{\mathrm{good}}, s_{\mathrm{bad}}\}$, $\mathcal{A} = \{0, 1\}$.

  - Solid lines: transitions of the nominal transition kernel $P^{\star}$.
  - Dashed lines: transitions of the worst case transition in the robust set.
  - Red solid line: the transition where the two RMDP instances differ in that different action leads to higher transition probability from s_{bad} to s_{good}.
- When starting from $s_1 = s_{\mathrm{good}}$, the nominal transition kernel keeps the agent at s_{good} and no information at s_{bad} is revealed!

Hardness Result

- For RMDP with total variation (TV) robust set, robust RL with interactive data collection inevitably incurs a regret of $\mathrm{Regret}(K) \geq \Omega(\rho \cdot HK)$, where $\rho =$ size of robust set.
- Hard example: a simple class of two RMDPs: $\mathcal{S} = \{s_{\mathrm{good}}, s_{\mathrm{bad}}\}$, $\mathcal{A} = \{0, 1\}$.



  - Solid lines: transitions of the nominal transition kernel $P^\star$.
  - Dashed lines: transitions of the worst case transition in the robust set.
  - Red solid line: the transition where the two RMDP instances differ in that different action leads to higher transition probability from s_{bad} to s_{good}.
- When starting from $s_1 = s_{\mathrm{good}}$, the nominal transition kernel keeps the agent at s_{good} and no information at s_{bad} is revealed!

Outline

Combating Hard Instance: Our Assumption

- We identify the obstacle as the curse of support shift: the disjointedness of the distributional support between training / testing environments.
  - A broader understanding: the states often appearing in testing Env. are extremely hard to arrive in the training Env. Conjecture: imply hardness for other ϕ-divergence robust set.
- To rule out such instances, we propose the Vanishing Minimal Value (VMV) assumption:

Assumption 1 (Vanishing Minimal Value).

The optimal robust value is zero at a specific state, i.e.,

$$\min_{s \in \mathcal{S}} V^{\star}_{1,P^{\star},\Phi}(s) = 0.$$

- Rules out the above hard instances (An equivalent form of robust Bellman equation where no explicit support shift occurs)
- Example (fail state assumption): $\exists s_f \in \mathcal{S}$, s.t. $R_h(s_f, \cdot) = 0$ and $P^{\star}_h(s_f|s_f, \cdot) = 1$.

Combating Hard Instance: Our Assumption

- We identify the obstacle as the curse of support shift: the disjointedness of the distributional support between training / testing environments.
  - A broader understanding: the states often appearing in testing Env. are extremely hard to arrive in the training Env. Conjecture: imply hardness for other ϕ-divergence robust set.
- To rule out such instances, we propose the Vanishing Minimal Value (VMV) assumption:

Assumption 1 (Vanishing Minimal Value).

The optimal robust value is zero at a specific state, i.e.,

$$\min_{s \in \mathcal{S}} V^{\star}_{1,P^{\star},\mathbf{\Phi}}(s) = 0.$$

- Rules out the above hard instances (An equivalent form of robust Bellman equation where no explicit support shift occurs)
- Example (fail state assumption): $\exists s_f \in \mathcal{S}$, s.t. $R_h(s_f, \cdot) = 0$ and $P^{\star}_h(s_f | s_f, \cdot) = 1$.

Algorithm Design

Our algorithm: **<u>OP</u>**timistic **<u>RO</u>**bust **<u>V</u>**alue **<u>I</u>**teration for **<u>TV</u>** Robust Set (`OPROVI-TV`).

In each episode $k \in [K]$, it has three stages:

- (Stage 1: Training env. model estimation) estimate training env. $P^\star$ as $\widehat{P}$
- (Stage 2: Optimistic robust planning) solve the optimal robust policy π^k for the estimated model $\widehat{P}$ based on a sophisticated joint consideration of:
  - <u>Robust optimal Bellman equation</u> to ensure exploitation and distributional robustness
  - <u>Optimistic robust value estimation</u> to encourage exploration
- (Stage 3: Interactive data collection) use policy π^k to collect data.

Algorithm Design

Our algorithm: **OP**timistic **RO**bust **V**alue **I**teration for **TV** Robust Set (`OPROVI-TV`).

In each episode $k \in [K]$, it has three stages:

- (Stage 1: Training env. model estimation) estimate training env. $P^\star$ as $\widehat{P}$
- (Stage 2: Optimistic robust planning) solve the optimal robust policy π^k for the estimated model $\widehat{P}$ based on a sophisticated joint consideration of:
  - Robust optimal Bellman equation to ensure exploitation and distributional robustness
  - Optimistic robust value estimation to encourage exploration
- (Stage 3: Interactive data collection) use policy π^k to collect data.

Algorithm Design

Our algorithm: **OP**timistic **RO**bust **V**alue **I**teration for **TV** Robust Set (OPROVI-TV).

In each episode $k \in [K]$, it has three stages:

- (Stage 1: Training env. model estimation) estimate training env. $P^\star$ as $\widehat{P}$
- (Stage 2: Optimistic robust planning) solve the optimal robust policy π^k for the estimated model $\widehat{P}$ based on a sophisticated joint consideration of:
  - Robust optimal Bellman equation to ensure exploitation and distributional robustness
  - Optimistic robust value estimation to encourage exploration
- (Stage 3: Interactive data collection) use policy π^k to collect data.

Algorithm Design

Our algorithm: **OP**timistic **RO**bust **V**alue **I**teration for **TV** Robust Set (`OPROVI-TV`).

In each episode $k \in [K]$, it has three stages:

- (Stage 1: Training env. model estimation) estimate training env. $P^\star$ as $\widehat{P}$
- (Stage 2: Optimistic robust planning) solve the optimal robust policy π^k for the estimated model $\widehat{P}$ based on a sophisticated joint consideration of:
  - Robust optimal Bellman equation to ensure exploitation and distributional robustness
  - Optimistic robust value estimation to encourage exploration
- (Stage 3: Interactive data collection) use policy π^k to collect data.

Theoretical Results

Theorem 1 (Online Regret of OPROVI-TV).

Under the VMV assumption, for any $\rho \in [0, 1)$, `OPROVI-TV` has an online regret of

$$\mathrm{Regret}(K) \leq \widetilde{\mathcal{O}}\left(\sqrt{\min\left\{H, \rho^{-1}\right\} \cdot H^2 SAK}\right).$$

As a corollary, `OPROVI-TV` is capable of finding an ε-optimal robust policy within

$$\widetilde{\mathcal{O}}\left(\min\{H, \rho^{-1}\} \cdot \frac{H^2 SA}{\varepsilon^2}\right)$$

interactive samples. $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors.

- First result of this kind
- Matching the sample complexity lower bound of generative model case [Shi et al., 2023]
- Requires less sample as the robust set size ρ increases

Theoretical Results

Theorem 1 (Online Regret of OPROVI-TV).

Under the VMV assumption, for any $\rho \in [0, 1)$, `OPROVI-TV` has an online regret of

$$\mathrm{Regret}(K) \leq \widetilde{\mathcal{O}}\left(\sqrt{\min\left\{H, \rho^{-1}\right\} \cdot H^2 SAK}\right).$$

As a corollary, `OPROVI-TV` is capable of finding an ε-optimal robust policy within

$$\widetilde{\mathcal{O}}\left(\min\{H, \rho^{-1}\} \cdot \frac{H^2 SA}{\varepsilon^2}\right)$$

interactive samples. $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors.

- First result of this kind
- Matching the sample complexity lower bound of generative model case [Shi et al., 2023]
- Requires less sample as the robust set size ρ increases

Outline

- Function approximations
  - Existing works have preliminary results on linear function approximations
  - General function approximations: identifying general learnability principle like Bellman rank, bilinear class, Bellman-Eluder dimension, generalized Eluder coefficient for standard online RL.
- Other types of robust set
  - KL divergence? It is even unknown whether robust RL with interactive data collection is possible in this case.
- Robust Markov games

Remark

Provably sample-efficient algorithms exist for generative model/offline learning setup [Blanchet et al., 2023], but remain unknown for interactive data collection!

- Function approximations
  - Existing works have preliminary results on linear function approximations
  - General function approximations: identifying general learnability principle like Bellman rank, bilinear class, Bellman-Eluder dimension, generalized Eluder coefficient for standard online RL.
- Other types of robust set
  - KL divergence? It is even unknown whether robust RL with interactive data collection is possible in this case.
- Robust Markov games

Remark

Provably sample-efficient algorithms exist for generative model/offline learning setup [Blanchet et al., 2023], but remain unknown for interactive data collection!

Thanks for your attention!

References I

J. Blanchet, M. Lu, T. Zhang, and H. Zhong. Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. Advances in Neural Information Processing Systems, 36, 2023.

L. Shi, G. Li, Y. Wei, Y. Chen, M. Geist, and Y. Chi. The curious price of distributional robustness in reinforcement learning with a generative model. Advances in Neural Information Processing Systems, 36, 2023.