Distributionally Robust Reinforcement Learning:

Formulations, Model-free Algorithms, and Sample Complexities

Shengbo Wang

MS&E — Stanford University

June 1, 2023

Joint work with Nian Si, Jose Blanchet, Zhengyuan Zhou

Outline

Introduction

Existing RL algorithms often make the implicit assumption that the training environment (usually a simulator) is the same as the deploying environment.

- Simulator can be be mis-specified.
- Even if a policy is trained directly in a real environment, the deployment environment may be different.

Distrbutionally robust (DR) RL is a framework that learns a more robust policy using the worst case value over some uncertainty set of probability measures.

Outline

Tabular RL

Finite MDP formulation

- State and action space $|S|, |A| < \infty$.
- Transition kernel $\mathcal{P}_0 = \left\{p^0_{s,a} \in \mathcal{P}(S)\right\}$.
- History dependent and randomized policy class Π
- Optimal infinite horizon discounted reward:

$$v^*(s) = \sup_{\pi \in \Pi} E^\pi_s \sum_{k=0}^{\infty} \gamma^k r(S_k, A_k)$$

Bellman equation and deterministic Markov optimality

$$v^*(s) = \sup_{a \in A} r(s, a) + \gamma p_{s,a}[v^*]$$

Robust MDP

MDP with transition kernel $\mathcal{P}_0 = \left\{ p^0_{s,a} \in \mathcal{P}(S) \right\}$ could be **inaccurate**.

DR optimal value function:

$$v^*(s, \Pi, \mathrm{K}_C) = \sup_{\pi \in \Pi} \inf_{\kappa \in \mathrm{K}_C} E_s^{\pi,\kappa} \left[\sum_{k=0}^{\infty} \gamma^k r(S_k, A_k) \right].$$

Adversarial environment:

$$\kappa = (\kappa_1, \kappa_2, \dots); \quad \kappa_t(\cdot | s_0, a_0, \dots, s_t, a_t); \quad \kappa_t(\cdot | s_t, a_t).$$

Bellman equation? Markov optimal for both the controller and the adversary?

Markov Optimality and DR Bellman Equation

Any marginal uncertainty sets: $\{\mathcal{P}_{s,a} \subset \mathcal{P}(\mathcal{S}) : s, a \in S \times A\}$.
SA-rectangularity: at time t and state S_t, after observing the history

$$H_t = (S_0, A_0, \ldots A_{t-1}, S_t)$$

and controller's next action A_t, the adversary freely chooses $p \in \mathcal{P}_{S_t, A_t}$.

González-Trejo et. al(2003): under *SA-rectangularity*, $v^*(s, \Pi, \mathrm{K}_{\mathrm{SA}})$ uniquely solves

$$v(s) = \sup_{a \in A} \inf_{p \in \mathcal{P}_{s,a},} r(s, a) + \gamma p[v].$$

Markov optimality for both players given by the sup and inf.

S-rectangularity (Wiesemann et al. 2013): The adversary cannot see the realization of the next action A_t. Markov optimality for both players.

Incomplete List of Literature

SA-rectanuglar:
History dependent adversary: González-Trejo et. al (2003).
Markov adversary: Nilim et al. (2005), Iyengar (2005).

S-rectangular:
Xu and Mannor (2010), Wiesemann et al. (2013).

General multistage stochastic program:
Shapiro (2022).

Outline

Outline

Model-based and model-free Approaches

Two principles, namely model-based and model-free, have motivated distinct algorithmic designs.

Model-based approach: Gather a dataset to construct an empirical version of the underlying MDP. Then, solve it using dynamic programming.

Model-free approach:

- Maintain only lower-dimensional statistics of the transition data, which are iteratively updated.
- E.g. Q-learning, V-learning, policy gradient.
- Memory and computation efficient, easily generalized to continuum space settings.

Outline

Robust q-function

We assume SA-rectangularity, a reference kernel $\left\{p^0_{s,a}\right\}$, and Kullback-Leibler divergence marginal uncertainty sets:

$$\mathcal{P}_{s,a}(\delta) := \left\{p : D_{\mathrm{KL}}\left(p \| p^0_{s,a}\right) \leq \delta\right\}.$$

The optimal DR q-function is the unique solution

$$\begin{aligned} q^*_\delta(s,a) &= r(s,a) + \gamma \inf_{p \in \mathcal{P}_{s,a}(\delta)} p[\sup_{a \in A} q^*_\delta(\cdot, a)] \\ &=: \mathcal{T}_\delta(q^*_\delta)(s,a). \end{aligned}$$

$\mathcal{T}_0$ recovers the Bellman operator for non-robust MDPs.

Greedy policy $\pi^*_\delta(s) = \arg\max_{a \in A} q^*_\delta(s,a)$ is optimal.

Goal: Learn the q^*_δ function.

The Q-learning

A simulator that take $(s, a) \in S \times A$ and return a new state $s' \sim p^0_{s,a}$.

Non-robust q-function

$$\begin{aligned} q^*_0(s, a) &= \mathcal{T}_0(q^*_0)(s, a) \\ &= r(s, a) + \gamma p[\sup_{a \in A} q^*_0(\cdot, a)] \end{aligned}$$

Non-robust Q-Learning: for all (s, a), sample $s' \sim p^0_{s,a}$ and update

$$\begin{aligned} Q_{k+1}(s, a) &= (1 - \alpha_k) Q_k(s, a) + \alpha_k (r + \gamma \max_{b \in A} Q_k(s', b)) \\ &= (1 - \alpha_k) Q_k(s, a) + \alpha_k \hat{\mathcal{T}}_{k+1}(Q_k). \end{aligned}$$

Unbiasedness: $E_{s' \sim p^0} \hat{\mathcal{T}}_{k+1}(q) = \mathcal{T}_0(q)$.

Stochastic Approximations

Fixed point equation induced by contraction mapping

$$q_0^* = \mathcal{T}_0(q_0^*).$$

I.i.d. sequence $\left\{\hat{\mathcal{T}}_k\right\}$ s.t. $E\hat{\mathcal{T}}_{k+1}(q) = \mathcal{T}_0(q)$, then iterations of

$$Q_{k+1}(s,a) = (1-\alpha_k)Q_k(s,a) + \alpha_k \hat{\mathcal{T}}_{k+1}(Q_k)$$

converges to q_0^* under mild assumptions. Chen et al. (2020): finite time convergence guarantees.

Estimator of $\mathcal{T}_\delta$

Recall the DR Bellman operator (for the q function)

$$\mathcal{T}_\delta(q)(s,a) = r(s,a) + \gamma \inf_{p \in \mathcal{P}_{s,a}(\delta)} p[v(q)]$$

compare to

$$\mathcal{T}_0(q)(s,a) = r(s,a) + \gamma p^0_{s,a}[v(q)].$$

where $v(q) = \sup_{a \in A} q(\cdot, a)$. Strong duality:

$$\inf_{p \in \mathcal{P}_{s,a}(\delta)} p[v(q)] = \sup_{\alpha \geq 0} -\alpha \log p^0_{s,a}[\exp(-v(q)/\alpha)] - \alpha\delta.$$

Non-parametric estimator: use $p^0_{n,s,a}$ for $p^0_{s,a}$

$$\mathbf{T}_{n,\delta}(q)(s,a) := r(s,a) + \sup_{\alpha \geq 0} -\alpha \log p^0_{n,s,a}[\exp(-v(q)/\alpha)] - \alpha\delta.$$

Typically biased.

Two Designs for DR Q-learning

Idea 1: Construct unbiased estimator $\hat{\mathcal{T}}_\delta$.

$$Q_{k+1}(s,a) = (1-\alpha_k)Q_k(s,a) + \alpha_k \hat{\mathcal{T}}_{\delta,k+1}(Q_k).$$

Liu et al. (2022) proposed randomized antithetic Multilevel Monte Carlo (MLMC) estimator introduced in [Blanchet and Glynn, 2015].
We improved their design and get finite variance estimator (W et al. 2023a).

Idea 2: Use biased estimator $\mathbf{T}_{n,\delta}$ and control the bias.

$$Q_{k+1}(s,a) = (1-\beta_k)Q_k(s,a) + \beta_k \mathbf{T}_{n,\delta,k+1}(Q_k).$$

Challenging to get tight bound on the bias. W et al. 2023b: Balance the systematic error caused by the bias and the statistical error.

Comparison of the Algorithms

Unbiased MLMC DRQL $Q_{k+1}(s,a) = (1-\alpha_k)Q_k(s,a) + \alpha_k \hat{\mathcal{T}}_{\delta,k+1}(Q_k)$.

- $\hat{\mathcal{T}}_\delta(q)$ has finite variance but infinite exponential moment.
- The random operator $\hat{\mathcal{T}}_\delta(\cdot)$ is not a contraction.
- The number of simulator calls N used to produce $\hat{\mathcal{T}}_\delta(q)$ is **random** with $EN = \Theta(1)$.

Biased DRQL $Q_{k+1}(s,a) = (1-\beta_k)Q_k(s,a) + \beta_k \mathbf{T}_{n,\delta,k+1}(Q_k)$:

- $\mathbf{T}_{n,\delta,k+1}(q)$ is bounded, hence sub-Gaussian for any $n \geq 1$.
- The random operator $\mathbf{T}_{n,\delta,k+1}(\cdot)$ is a γ-contraction.
- Need to choose $n = \Omega((1-\gamma)^{-1}\epsilon^{-1})$ to get a target error ϵ.

Comparison of the Algorithms

Unbiased MLMC DRQL $Q_{k+1}(s,a) = (1-\alpha_k)Q_k(s,a) + \alpha_k \hat{\mathcal{T}}_{\delta,k+1}(Q_k)$.

- $\hat{\mathcal{T}}_\delta(q)$ has finite variance but infinite exponential moment.
- The random operator $\hat{\mathcal{T}}_\delta(\cdot)$ is not a contraction.
- The number of simulator calls N used to produce $\hat{\mathcal{T}}_\delta(q)$ is **random** with $EN = \Theta(1)$.

Biased DRQL $Q_{k+1}(s,a) = (1-\beta_k)Q_k(s,a) + \beta_k \mathbf{T}_{n,\delta,k+1}(Q_k)$:

- $\mathbf{T}_{n,\delta,k+1}(q)$ is bounded, hence sub-Gaussian for any $n \geq 1$.
- The random operator $\mathbf{T}_{n,\delta,k+1}(\cdot)$ is a γ-contraction.
- Need to choose $n = \Omega((1-\gamma)^{-1}\epsilon^{-1})$ to get a target error ϵ.

Variance-reduced DRQL

Wainwright (2019): Variance reduction using a epoch structure.

Variance-reduced DRQL
At epoch $l \le l_{\mathrm{vr}}$, do

$$Q_{l,k+1} = (1-\lambda_k)Q_{l,k} + \lambda_k \left(\mathbf{T}_{l,k+1}(Q_{l,k}) - \mathbf{T}_{l,k+1}(\hat{Q}_{l-1}) + \widetilde{\mathbf{T}}_l(\hat{Q}_{l-1})\right)$$

for $k = 0, 1 \ldots, k_{\mathrm{vr}}$.
Assign $\hat{Q}_l = Q_{l,k_{\mathrm{vr}}+1}$.

Geometric pathwise convergence:

$$P\left(\|\hat{Q}_l - q^*_\delta\| \le \frac{2^{-l}}{1-\gamma}, \forall l \le l_{\mathrm{vr}}\right) \ge 1-\eta$$

Outline

Outline

Model-based Algorithms

KL uncertainty sets DR-RL:

Algorithm	Sample Complexity	Origin
DRVI	$\frac{\lvert S\rvert^2\lvert A\rvert}{e^{O(1-\gamma)}(1-\gamma)^4\epsilon^2\delta^2}$	Zhou et al. 2021
REVI/DRVI	$\frac{\lvert S\rvert^2\lvert A\rvert}{e^{O(1-\gamma)}(1-\gamma)^4\epsilon^2\delta^2}$	Panaganti and Kalathil 2021
DRVI	$\frac{\lvert S\rvert^2\lvert A\rvert}{(1-\gamma)^4\epsilon^2 p_\wedge^2\delta^2}$	Yang et al. 2021
DRVI-LCB	$\frac{\lvert S\rvert\lvert A\rvert}{(1-\gamma)^4\epsilon^2 p_\wedge\delta^2}$	Shi and Chi 2022

where

- ϵ: target error.
- δ: radius of the uncertainty set.
- $p_\wedge$: minimal support probability.

All complexity bounds has $\tilde{O}(\delta^{-2})$ dependence as $\delta \downarrow 0$

Model-free Algorithms

For $\delta \leq \tilde{O}(p_\wedge)$ and KL uncertainty sets,

Algorithm	Sample Complexity
MLMC DRQL	$\lvert S\rvert\lvert A\rvert(1-\gamma)^{-5}\epsilon^{-2}p_\wedge^{-6}\delta^{-4}$
DRQL	$\lvert S\rvert\lvert A\rvert(1-\gamma)^{-5}\epsilon^{-2}p_\wedge^{-3}$
Variance-reduced DRQL	$\lvert S\rvert\lvert A\rvert(1-\gamma)^{-4}\epsilon^{-2}p_\wedge^{-3}$

Our methods can be easily generalized to other ϕ-divergence uncersainty sets DRRL. (KL is the hard one)

ϕ-diveregence, strongly convex:

Algorithm	Sample Complexity	Origin
Model-free DR-RL	$\lvert S\rvert\lvert A\rvert\epsilon^{-4}poly(1-\gamma)^{-1}$	Yang et al. 2023

Outline

The Bias and Variance

To get the correct ϵ^{-2} dependence, it is necessary that the bias of $\mathbf{T}_{n,\delta}$ is of order n^{-1} and the variance of $\hat{\mathcal{T}}_{\delta}$ is uniformly bounded.

The bias is $O(n^{-1})$ if the functional $p_{n,s,a} \to \mathbf{T}_{n,\delta}(q)$ is smooth in $p_{n,s,a}$.

The Dual Functional

Recall that the estimator:

$$\mathbf{T}_{n,\delta}(q)(s,a) := r(s,a) + \sup_{\alpha \geq 0} -\alpha \log p^0_{n,s,a}[\exp(-v(q)/\alpha)] - \alpha\delta.$$

Fix function v, define the dual functionals

$$g_v(p) := \sup_{\alpha \geq 0} -\alpha \log p[\exp(-v/\alpha)] - \alpha\delta =: \sup_{\alpha \geq 0} f_v(p,\alpha).$$

If $g_v(p)$ **is infinitely differentiable**, bias expansion:

$$Eg_v(p_n) - g_v(p) \approx \cancel{E(p_n - p)[Dg_v(p)]} + E(p_n - p)D^2 g_v(p)(p_n - p) + O(n^{-3}).$$

Turns out that the variance of $\hat{\mathcal{T}}_\delta(q)$ is also closely related to the coefficient of the second order term.

Differentiablity of the Dual Functional

Recall that

$$g_v(p) := \sup_{\alpha \geq 0} -\alpha \log p[\exp(-v/\alpha)] - \alpha\delta =: \sup_{\alpha \geq 0} f_v(p, \alpha).$$

If dual optimizer α^* and α_n^* of $f_v(p, \cdot)$ and $f_v(p_n, \cdot)$ are all positive, then they are the unique solution to the first order optimality condition for $q = p, p_n$

$$0 = d_\alpha f(q, \alpha) = -\log q[\exp(-v/\alpha)] - \delta - \frac{q[v \exp(-v/\alpha)]}{\alpha q[\exp(-v/\alpha)]}.$$

Implicit function theorem implies that $\alpha^*(p)$ is a smooth function of p. Therefore, $g_v(\cdot)$ is differentiable (C^∞).

Bias and Variance Bounds

$$g_v(p) = \sup_{\alpha \geq 0} -\alpha \log p[\exp(-v/\alpha)] - \alpha\delta$$

By bounding the $D^2 g_v$, we get

Proposition: bias and variance bounds

If $\delta \leq \tilde{O}(p_\wedge)$, then exist c, c' s.t.

$$\| E\mathbf{T}_{n,\delta}\delta(Q) - \mathcal{T}_\delta(Q)\|_\infty \leq \frac{c\tilde{l}}{p_\wedge^3 n}\left(r_{\max} + \|Q\|_\infty\right).$$

and

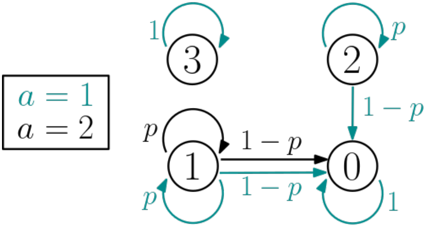
$$E\|\hat{\mathcal{T}}_\delta(Q) - \mathcal{T}_\delta(Q)\|_\infty^2 \leq \frac{c\tilde{l}}{p_\wedge^6}\left(r_{\max}^2 + \|Q\|_\infty^2\right).$$

where $\tilde{l}$ is some log-order term.

Outline

A Hard MDP

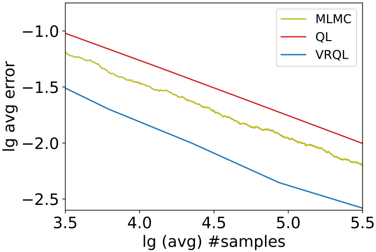
The following hard MDP is constructed by Li et al, (2021).

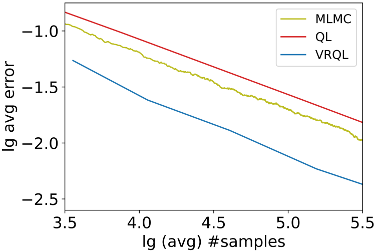


They proved that when $p = (4\gamma - 1)/3\gamma$, the Q-learning algorithm on this MDP has sample complexity $\tilde{\Theta}(\epsilon^{-2}(1-\gamma)^{-4})$.

Performance of the Algorithms

Log of averaged error $\|Q_k - Q^*\|_\infty$ is plotted against the log number samples





Thanks for listening!