First-order Policy Optimization for Robust Markov Decision Process

Yan Li

Georgia Institute of Technology

Joint work with George Lan, Tuo Zhao

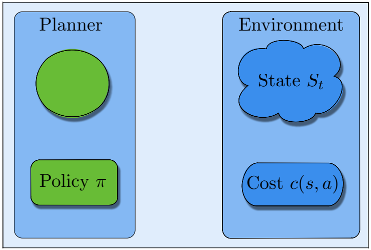
Markov Decision Process & Policy Optimization

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$

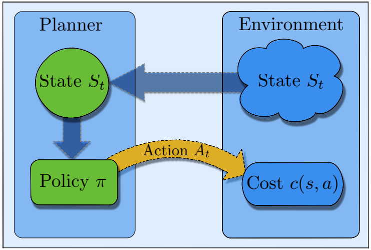


Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Decision making:

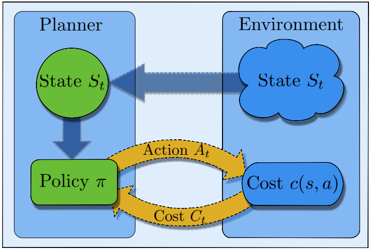
1. Observe current state S_t and feed into policy
2. Make A_t following distribution $\pi(\cdot|S_t)$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



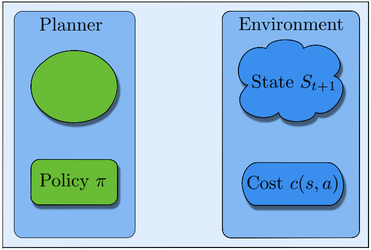
Observing loss: $C_t = c(S_t, A_t) \in [0, 1]$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



State transition: S_{t+1} follows distribution $\mathbb{P}(\cdot|S_t, A_t)$

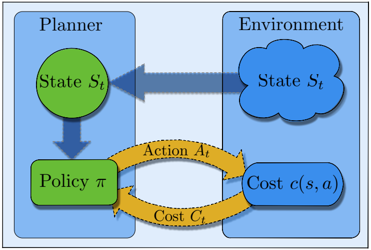
Repeat decision process ..

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Trajectory:

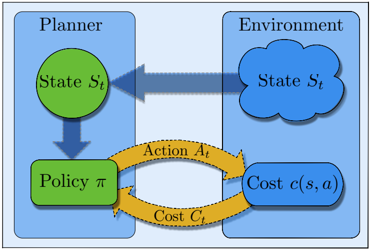
$$\{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_t, A_t, C_t), \ldots\}$$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Performance (value function):

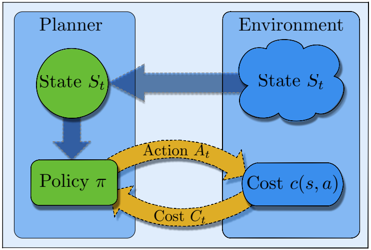
$$V_{\mathbb{P}}^{\pi}(s) = \mathbb{E}_{\mathbb{P}}^{\pi}\Big[\sum_{t=0}^{\infty} \underbrace{\gamma^t C_t}_{\text{discounting future}} \Big| S_0 = s\Big]$$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Planning: find the optimal policy of

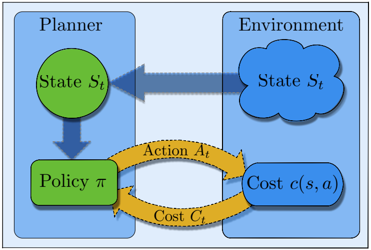
$$\min_{\pi} V_{\mathbb{P}}^{\pi}(s), \ \forall s \in \mathcal{S}$$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Planning with an equivalent objective:

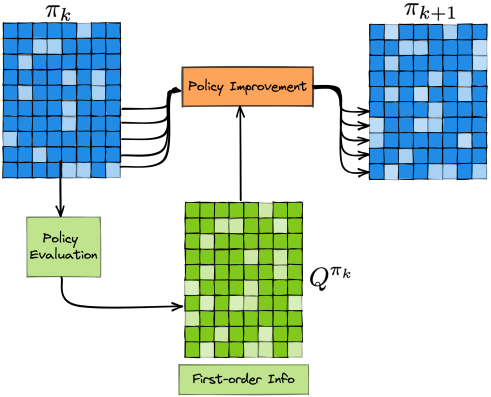
$$\min_{\pi} f_\rho(\pi) = \sum_{s\in\mathcal{S}} \rho(s) V_{\mathbb{P}}^{\pi}(s) \quad \Rightarrow \quad \text{Non-convex}$$

Planning Methods for MDP

1. Linear programming based methods
    - stochastic primal-dual methods
2. Dynamic programming based methods
    - stochastic value iteration or Q-Learning
    - can diverge even with linear approximation
3. Nonlinear programming based methods
    - **policy gradient methods**
    - much more friendly to function approximation
    - Only until very recently, these methods were shown to exhibit comparable or even superior performance guarantees than alternative methods

Policy Gradients – Overview

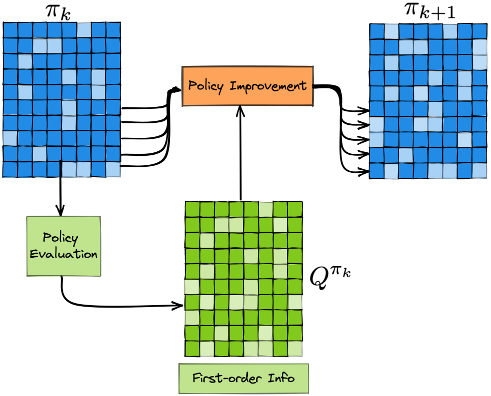
Policy Gradients - A Basic Skeleton



First-order policy optimization:

1. $\mathrm{Eval}(\pi_k) \rightarrow Q_{\mathbb{P}}^{\pi_k}$
2. Construct gradient information G_k
3. $\mathrm{Update}(\pi_k, G_k) \rightarrow \pi_{k+1}$
4. Repeat ..

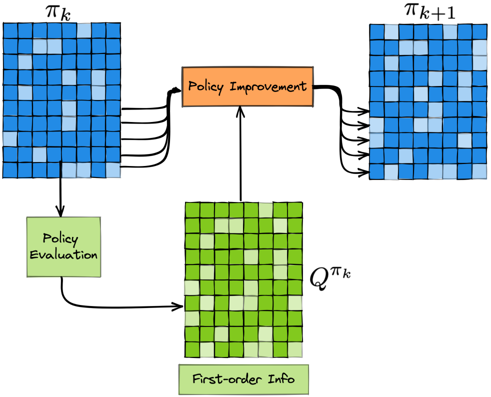
Policy Gradients - A Basic Skeleton



Q-function:

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty}\gamma^t c(S_t,A_t)\middle| S_0 = s, A_0 = a\right]$$

Policy Gradients - A Basic Skeleton



⋆ **Challenges:**

- Non-convex landscape
- Transition $\mathbb{P}$ and cost $c(\cdot)$ can be unknown

Policy Gradients – Existing Development

1. Deterministic setting: exact first-order information:
   - Even-Dar, Kakade, Mansour '09: $\mathcal{O}(1/\sqrt{T})$
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/T)$
   - Cen et. al. '20: linear for entropy regularized MDPs

Policy Gradients – Existing Development

1. Deterministic setting: exact first-order information:
   - Even-Dar, Kakade, Mansour '09: $\mathcal{O}(1/\sqrt{T})$
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/T)$
   - Cen et. al. '20: linear for entropy regularized MDPs
2. Stochastic setting – sample complexity
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/\epsilon^4)$
   - Shani, Efroni, Mannor '20: $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^3)$ for entropy regularized MDPs

Policy Gradients – Existing Development

1. Deterministic setting: exact first-order information:
   - Even-Dar, Kakade, Mansour '09: $\mathcal{O}(1/\sqrt{T})$
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/T)$
   - Cen et. al. '20: linear for entropy regularized MDPs
2. Stochastic setting – sample complexity
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/\epsilon^4)$
   - Shani, Efroni, Mannor '20: $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^3)$ for entropy regularized MDPs
3. Policy mirror descent (Lan, '21)
   - Deterministic: linear for both regularized and un-regularized
   - Stochastic: $\mathcal{O}(1/\epsilon^2)$ un-regularized; $\mathcal{O}(1/\epsilon)$ regularized

Robust Markov Decision Process

Motivating Examples

I: Planning with Pre-collected Data $\mathcal{D}$

Direct approach

1. Estimate transition kernel $\widehat{\mathbb{P}} \approx \mathbb{P}$ from $\mathcal{D}$
2. Planning with estimated $\widehat{\mathbb{P}}$

Motivating Examples

I: Planning with Pre-collected Data $\mathcal{D}$

Direct approach

1. Estimate transition kernel $\widehat{\mathbb{P}} \approx \mathbb{P}$ from $\mathcal{D}$
2. Planning with estimated $\widehat{\mathbb{P}}$

Subject to randomness/error in data collection

Motivating Examples

I: Planning with Pre-collected Data $\mathcal{D}$

Direct approach

1. Estimate transition kernel $\widehat{\mathbb{P}} \approx \mathbb{P}$ from $\mathcal{D}$
2. Planning with estimated $\widehat{\mathbb{P}}$

Subject to randomness/error in data collection

Robust approach

1. Construct $\mathcal{P}$ s.t. $\mathbb{P} \in \mathcal{P}$ with high confidence
2. Planning within $\mathcal{P}$ to hedge against randomness

Motivating Examples

II: Sim-to-real Transition (Robotics)

- Training environment (simulation) has $\mathbb{P}_{\text{sim}}$
- Deployment (real-life) environment has $\mathbb{P}_{\text{real}} \approx \mathbb{P}_{\text{sim}}$, but not equal
- Ultimate goal is to perform well for $\mathbb{P}_{\text{real}}$

Motivating Examples

II: Sim-to-real Transition (Robotics)

- Training environment (simulation) has $\mathbb{P}_{\text{sim}}$
- Deployment (real-life) environment has $\mathbb{P}_{\text{real}} \approx \mathbb{P}_{\text{sim}}$, but not equal
- Ultimate goal is to perform well for $\mathbb{P}_{\text{real}}$

Robust approach

1. Construct $\mathcal{P}$ based on robustness preference
   - ϵ-contamination model (Huber, '64):
     $$\mathcal{P} = \{(1-\epsilon)\mathbb{P}_{\text{sim}} + \epsilon\mathbb{Q} : \mathbb{Q} \in \mathcal{Q} \text{ (pre-specified)}\}$$
   - KL-divergence based:
     $$\mathcal{P} = \{\mathbb{P} : \mathrm{KL}(\mathbb{P}(\cdot|s,a) \| \mathbb{P}_{\text{sim}}(\cdot|s,a)) \leq \epsilon\}$$
   - Large ϵ yields stronger robustness
2. Planning within $\mathcal{P}$ to hedge against environment changes
   - Can only samples from interaction with $\mathbb{P}_{\text{sim}}$

Robust Markov Decision Process

▷ **Robust Objective:**

$$\min_{\pi} \left\{ f_r(\pi) := \sum_{s\in\mathcal{S}} \rho(s) \underbrace{\max_{\mathbb{P}\in\mathcal{P}} V_{\mathbb{P}}^{\pi}(s)}_{V_r^{\pi}(s)} \right\}$$

- $\mathcal{P} = \{\mathbb{P} : \mathbb{P}(\cdot|s,a) = \mathbb{P}_{\mathrm{N}}(\cdot|s,a) + u(\cdot|s,a),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A}, u \in \mathcal{U}\}$.
- $\mathbb{P}_{\mathrm{N}}$: nominal transition kernel
- $\mathcal{U}$: set of possible perturbations
- Non-convex, non-smooth in π

Robust Markov Decision Process

▷ **Robust Objective:**

$$\min_{\pi} \left\{ f_r(\pi) := \sum_{s\in\mathcal{S}} \rho(s) \underbrace{\max_{\mathbb{P}\in\mathcal{P}} V^{\pi}_{\mathbb{P}}(s)}_{V^{\pi}_{r}(s)} \right\}$$

- $\mathcal{P} = \{\mathbb{P} : \mathbb{P}(\cdot|s,a) = \mathbb{P}_{\mathrm{N}}(\cdot|s,a) + u(\cdot|s,a),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A}, u \in \mathcal{U}\}$.
- $\mathbb{P}_{\mathrm{N}}$: nominal transition kernel
- $\mathcal{U}$: set of possible perturbations
- Non-convex, non-smooth in π

▷ **Structure of Ambiguity Set:**

1. (s,a)-rectangularity [our focus]:

$$\mathcal{P} = \Pi_{(s,a)\in\mathcal{S}\times\mathcal{A}}\, \mathcal{P}_{s,a},\ \ \mathcal{P}_{s,a} \subseteq \Delta_{\mathcal{S}}$$

   - No coupling of uncertainties for different state-action pair
   - Equivalence to nested robust formulation

Robust Markov Decision Process

▷ **Robust Objective:**

$$\min_{\pi} \left\{ f_r(\pi) := \textstyle\sum_{s\in\mathcal{S}} \rho(s) \underbrace{\max_{\mathbb{P}\in\mathcal{P}} V_{\mathbb{P}}^{\pi}(s)}_{V_r^{\pi}(s)} \right\}$$

- $\mathcal{P} = \{\mathbb{P} : \mathbb{P}(\cdot|s,a) = \mathbb{P}_{\mathrm{N}}(\cdot|s,a) + u(\cdot|s,a),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A}, u \in \mathcal{U}\}.$
- $\mathbb{P}_{\mathrm{N}}$: nominal transition kernel
- $\mathcal{U}$: set of possible perturbations
- Non-convex, non-smooth in π

▷ **Structure of Ambiguity Set:**

1. (s,a)-rectangularity [our focus]:

$$\mathcal{P} = \Pi_{(s,a)\in\mathcal{S}\times\mathcal{A}}\, \mathcal{P}_{s,a},\ \ \mathcal{P}_{s,a} \subseteq \Delta_{\mathcal{S}}$$

   - No coupling of uncertainties for different state-action pair
   - Equivalence to nested robust formulation
2. Popular alternatives: s-rectangularity (Wiesemann et al., '13), r-rectangularity (Goyal and Grand-Clement, '23)
   - See Li and Shapiro for a unified treatment
3. General cases: NP hard

Robust Markov Decision Process

Can we learn robust policy, while given different levels of access to $\mathcal{P}$?

▷ **"Access of $\mathcal{P}$"**

1. Deterministic: Both $\mathbb{P}_{\mathrm{N}}$ and $\mathcal{U}$ are known
2. Stochastic: can only draw samples/trajectories from $\mathbb{P}_{\mathrm{N}}$

Robust Markov Decision Process

Can we learn robust policy, while given different levels of access to $\mathcal{P}$?

▷ **"Access of $\mathcal{P}$"**

1. Deterministic: Both $\mathbb{P}_{\mathrm{N}}$ and $\mathcal{U}$ are known
2. Stochastic: can only draw samples/trajectories from $\mathbb{P}_{\mathrm{N}}$

▷ **Existing Development**

1. Value based methods (vast majority):
   - Tamar et. al, '14; Roy et. al, '17; Liu et. al, '22; many others
2. Policy gradient methods (relatively few):
   - Wang and Zou, '22: smoothing argument
     - $\mathcal{O}(1/\epsilon^3)$ iterations in deterministic setting
     - $\mathcal{O}(1/\epsilon^7)$ samples in stochastic setting
     - Tailors to special (s,a)-rectangular set
   - Wang et al., '23: smoothing argument
     - $\mathcal{O}(1/\epsilon^4)$ iterations in deterministic setting
   - Non-optimal (even $\mathcal{U} = \{\mathbf{0}\}$)

Robust Policy Mirror Descent: Preview

Preview of Results

▷ Robust Policy Mirror Descent

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P}\in\mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

Preview of Results

▷ Robust Policy Mirror Descent

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P}\in\mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

▷ Parameters and Variants

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^p(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s)\rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$

Preview of Results

▷ Robust Policy Mirror Descent

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P}\in\mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

▷ Parameters and Variants

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^{p}(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s)\rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$
  3. natural policy gradient: $w(p) = \sum_{a\in\mathcal{A}} p_a \log(p_a)$:

$$\pi_{k+1}(a|s) \propto \pi_k(a|s)\exp\left(-\eta_k Q_r^{\pi_k}(s,a)\right)$$

Preview of Results

▷ Robust Policy Mirror Descent

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P} \in \mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s, \cdot), p \rangle + \mathcal{D}_{\pi_k}^p(s)$$

▷ Parameters and Variants

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^p(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s) \rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$
  3. natural policy gradient: $w(p) = \sum_{a \in \mathcal{A}} p_a \log(p_a)$:
     $$\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(-\eta_k Q_r^{\pi_k}(s, a)\right)$$
  4. Tsallis divergence with index $q \in (0, 1)$: $w(p) = -\sum_{a \in \mathcal{A}} p_a^p$
     - π_{k+1} can be computed using simple bisection (Li and Lan, '23)

Preview of Results

▷ **Robust Policy Mirror Descent**

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P}\in\mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$
Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

1. **Versatile:** recovers PMD for non-robust MDP (Lan, '21)

Preview of Results

▷ **Robust Policy Mirror Descent**

Algorithm RPMD update: $\pi_k \to \pi_{k+1}$

Input: Compute robust $Q_r^{\pi_k} := \max_{\mathbb{P}\in\mathcal{P}} Q_{\mathbb{P}}^{\pi_k}$
Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

1. **Versatile:** recovers PMD for non-robust MDP (Lan, '21)
2. **Efficient:**
   - Deterministic setting (exact $Q_r^{\pi_k}$): $\mathcal{O}(\log(1/\epsilon))$ iterations
   - Stochastic setting (estimated $Q_r^{\pi_k}$): $\mathcal{O}(1/\epsilon^2)$ samples
   - Optimal dependence on ϵ

First-order Viewpoint and Intuitions

Issues with Policy Gradients

▷ **Not-so-friendly Landscape**

1. $V_r^\pi(s)$ is only almost everywhere (Hausdorff sense) differentiable
2. Need to handle potential non-smoothness/non-differentiability

Issues with Policy Gradients

▷ **Not-so-friendly Landscape**

1. $V_r^\pi(s)$ is only almost everywhere (Hausdorff sense) differentiable
2. Need to handle potential non-smoothness/non-differentiability

▷ **Additional Issues**

1. The analytic form of gradient (if exists):

$$\nabla f_r(\pi)[s,a] = \frac{1}{1-\gamma} d_\rho^{\pi,\mathbb{P}_\pi}(s) Q_r^\pi(s,a)$$

- $d_\rho^{\pi,\mathbb{P}_\pi}(s) \coloneqq (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=0}^{\infty}\gamma^t \rho(s')\mathrm{Prob}^{\pi,\mathbb{P}_\pi}(S_t = s | S_0 = s')$
- needs worst kernel $\mathbb{P}_\pi$ of π – difficult to compute/estimate

Issues with Policy Gradients

▷ Not-so-friendly Landscape

1. $V_r^\pi(s)$ is only almost everywhere (Hausdorff sense) differentiable
2. Need to handle potential non-smoothness/non-differentiability

▷ Additional Issues

1. The analytic form of gradient (if exists):

$$\nabla f_r(\pi)[s,a] = \tfrac{1}{1-\gamma} d_\rho^{\pi,\mathbb{P}_\pi}(s) Q_r^\pi(s,a)$$

   - $d_\rho^{\pi,\mathbb{P}_\pi}(s) := (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=0}^\infty \gamma^t \rho(s')\mathrm{Prob}^{\pi,\mathbb{P}_\pi}(S_t = s|S_0 = s')$
   - needs worst kernel $\mathbb{P}_\pi$ of π – difficult to compute/estimate

2. Going from gradient stationarity to global optimality is indirect
   - Need additional smoothing (Wang and Zou, '22, Wang et al., '23)
   - Local-to-global conversion already non-optimal in non-robust case

Issues with Policy Gradients

▷ Not-so-friendly Landscape

1. $V_r^\pi(s)$ is only almost everywhere (Hausdorff sense) differentiable
2. Need to handle potential non-smoothness/non-differentiability

▷ Additional Issues

1. The analytic form of gradient (if exists):

$$\nabla f_r(\pi)[s,a] = \frac{1}{1-\gamma} d_\rho^{\pi,\mathbb{P}_\pi}(s) Q_r^\pi(s,a)$$

   - $d_\rho^{\pi,\mathbb{P}_\pi}(s) := (1-\gamma)\sum_{s'\in\mathcal{S}}\sum_{t=0}^\infty \gamma^t \rho(s')\mathrm{Prob}^{\pi,\mathbb{P}_\pi}(S_t = s | S_0 = s')$
   - needs worst kernel $\mathbb{P}_\pi$ of π – difficult to compute/estimate

2. Going from gradient stationarity to global optimality is indirect
   - Need additional smoothing (Wang and Zou, '22, Wang et al., '23)
   - Local-to-global conversion already non-optimal in non-robust case

⋆ **Need alternative first-order information** ⋆

"Useful" First-order Information

⋆ Robust Q-function as "Subgradient" ⋆

▷ **Local Improvement**

$$V_r^{\pi'}(s) - V_r^{\pi}(s) \leq \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d_s^{\pi', \mathbb{P}_{\pi'}}} \left\langle Q_r^{\pi}, \pi' - \pi \right\rangle_{s'}$$

- Following $-Q_r^{\pi}$ improves the value

"Useful" First-order Information

⋆ **Robust Q-function as "Subgradient"** ⋆

▷ **Local Improvement**

$$V_r^{\pi'}(s) - V_r^{\pi}(s) \leq \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d_s^{\pi', \mathbb{P}_{\pi'}}} \left\langle Q_r^{\pi}, \pi' - \pi \right\rangle_{s'}$$

- Following $-Q_r^{\pi}$ improves the value

▷ **Global Convergence**

$$\mathbb{E}_{s' \sim d_s^{\pi^*, \mathbb{P}_{\pi}}} \left[\langle Q_r^{\pi}, \pi - \pi^* \rangle_{s'}\right] \geq (1-\gamma)\left(V_r^{\pi}(s) - V_r^{\pi^*}(s)\right)$$

- Q_r^{π} provides enough information on optimality gap
  - ⋆ Proper state aggregation is required

"Useful" First-order Information

⋆ **Robust Q-function as "Subgradient"** ⋆

▷ **Local Improvement**

$$V_r^{\pi'}(s) - V_r^{\pi}(s) \leq \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d_s^{\pi'}, \mathbb{P}_{\pi'}} \left\langle Q_r^{\pi}, \pi' - \pi \right\rangle_{s'}$$

- Following $-Q_r^{\pi}$ improves the value

▷ **Global Convergence**

$$\mathbb{E}_{s' \sim d_s^{\pi^*}, \mathbb{P}_{\pi}} \left[\langle Q_r^{\pi}, \pi - \pi^* \rangle_{s'} \right] \geq (1-\gamma)\left(V_r^{\pi}(s) - V_r^{\pi^*}(s) \right)$$

- Q_r^{π} provides enough information on optimality gap
  - ⋆ Proper state aggregation is required

▷ Q_r^{π} bears great similarities of subgradients for convex problems

Robust Policy Mirror Descent: Deterministic Setting

Convergence Characterization

Theorem

Let $M = \sup_{\mathbb{P} \in \mathcal{P}} \| d_\rho^{\pi^*, \mathbb{P}} / \rho \|_\infty$ *and* $M' = \sup_{\mathbb{P}, \mathbb{P}' \in \mathcal{P}} \| d_\rho^{\pi^*, \mathbb{P}} / d_\rho^{\pi^*, \mathbb{P}'} \|_\infty$*. In RPMD, choosing* $\eta_k \geq \eta_{k-1} \left(1 - \frac{1-\gamma}{M}\right)^{-1} M'$ *yields*

$$f_\rho(\pi_k) - f_\rho(\pi^*) \leq \left(1 - \frac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\textit{from initialization}}$$

1. First linear rate for first-order policy based method

Convergence Characterization

Theorem

Let $M = \sup_{\mathbb{P}\in\mathcal{P}} \|d_\rho^{\pi^*,\mathbb{P}}/\rho\|_\infty$ *and* $M' = \sup_{\mathbb{P},\mathbb{P}'\in\mathcal{P}} \|d_\rho^{\pi^*,\mathbb{P}}/d_\rho^{\pi^*,\mathbb{P}'}\|_\infty$. *In RPMD, choosing* $\eta_k \geq \eta_{k-1}\left(1-\frac{1-\gamma}{M}\right)^{-1} M'$ *yields*

$$f_\rho(\pi_k) - f_\rho(\pi^*) \leq \left(1-\frac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\textit{from initialization}}$$

1. First linear rate for first-order policy based method
2. Subsumes the special case of non-robust MDPs

$$M = \|d_\rho^{\pi^*}/\rho\|_\infty,\ M' = 1.$$

Convergence Characterization

Theorem

Let $M = \sup_{\mathbb{P}\in\mathcal{P}} \|d_\rho^{\pi^*,\mathbb{P}}/\rho\|_\infty$ *and* $M' = \sup_{\mathbb{P},\mathbb{P}'\in\mathcal{P}} \|d_\rho^{\pi^*,\mathbb{P}}/d_\rho^{\pi^*,\mathbb{P}'}\|_\infty$*. In RPMD, choosing* $\eta_k \geq \eta_{k-1}\left(1-\frac{1-\gamma}{M}\right)^{-1} M'$ *yields*

$$f_\rho(\pi_k) - f_\rho(\pi^*) \leq \left(1-\frac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\text{from initialization}}$$

1. First linear rate for first-order policy based method
2. Subsumes the special case of non-robust MDPs

$$M = \|d_\rho^{\pi^*}/\rho\|_\infty, \ M' = 1.$$

3. Unclear whether dependence on M is tight
   - Appears also for non-robust MDP with linear rate
   - Seems removable with a sublinear rate

Robust Policy Mirror Descent: Stochastic Setting

Stochastic Robust Policy Mirror Descent

Algorithm SRPMD update: $\pi_k \to \pi_{k+1}$

Input: Evaluate $\widehat{Q}_r^{\pi_k,\xi_k} \approx Q_r^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k,\xi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

Stochastic Robust Policy Mirror Descent

Algorithm SRPMD update: $\pi_k \to \pi_{k+1}$

Input: Evaluate $\widehat{Q}_r^{\pi_k,\xi_k} \approx Q_r^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k,\xi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

Theorem

With the same stepsize as RPMD, if $\mathbb{E}_{\xi_k}\|Q_r^{\pi_k,\xi_k} - Q_r^{\pi_k}\|_\infty \leq e$ *for all* $k \geq 0$, *then*

$$\mathbb{E}\left[f_\rho(\pi_k) - f_\rho(\pi^*)\right] \leq \left(1 - \frac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\text{from initialization}} + \frac{4Me}{(1-\gamma)^2}$$

Stochastic Robust Policy Mirror Descent

Algorithm SRPMD update: $\pi_k \to \pi_{k+1}$

Input: Evaluate $\widehat{Q}_r^{\pi_k,\xi_k} \approx Q_r^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k,\xi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

Theorem

With the same stepsize as RPMD, if $\mathbb{E}_{\xi_k} \|Q_r^{\pi_k,\xi_k} - Q_r^{\pi_k}\|_\infty \leq e$ *for all* $k \geq 0$, *then*

$$\mathbb{E}\left[f_\rho(\pi_k) - f_\rho(\pi^*)\right] \leq \left(1 - \tfrac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\text{from initialization}} + \tfrac{4Me}{(1-\gamma)^2}$$

▷ Converges up to the noise level

Stochastic Robust Policy Mirror Descent

Algorithm SRPMD update: $\pi_k \to \pi_{k+1}$

Input: Evaluate $\widehat{Q}_r^{\pi_k,\xi_k} \approx Q_r^{\pi_k}$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \operatorname{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle Q_r^{\pi_k,\xi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^p(s)$$

Theorem

With the same stepsize as RPMD, if $\mathbb{E}_{\xi_k}\|Q_r^{\pi_k,\xi_k} - Q_r^{\pi_k}\|_\infty \le e$ *for all* $k \ge 0$*, then*

$$\mathbb{E}\left[f_\rho(\pi_k) - f_\rho(\pi^*)\right] \le \left(1 - \tfrac{1-\gamma}{M}\right)^k \cdot \underbrace{\mathcal{O}(1)}_{\text{from initialization}} + \tfrac{4Me}{(1-\gamma)^2}$$

- Converges up to the noise level
- Need to interact with $\mathbb{P}_{\mathrm{N}}$ to learn robust Q-function

Learning the Robust Q-function

Exploiting Access to $\mathbb{P}_{\mathrm{N}}$

▷ **when $\mathcal{U}$ is known**

Algorithm Robust Temporal Difference Learning: $\pi \to Q_r^{\pi,\xi}$

for $t = 0, 1, \ldots$ **do**

Collect $s_{t+1} \sim \mathbb{P}_{\mathrm{N}}(\cdot|s_t, a_t)$, and make action $a_{t+1} \sim \pi(\cdot|s_{t+1})$

Update:

$$\theta_{t+1} = \theta_t + \alpha_t \big[c(s_t, a_t) + \gamma\theta_t(s_{t+1}, a_{t+1}) + \sigma_{\mathcal{U}_{s_t,a_t}}(M(\pi, \theta_t)) - \theta_t(s_t, a_t)\big] e(s_t, a_t)$$

end for

- $\sigma_X(\cdot)$ is the support function of X, $[M(\pi, x)](s) = \sum_{a\in\mathcal{A}} \pi(a|s)x(s, a)$

Learning the Robust Q-function

Exploiting Access to $\mathbb{P}_{\mathrm{N}}$

▷ **when $\mathcal{U}$ is known**

Algorithm Robust Temporal Difference Learning: $\pi \to Q_r^{\pi,\xi}$

for $t = 0, 1, \ldots$ **do**

Collect $s_{t+1} \sim \mathbb{P}_{\mathrm{N}}(\cdot|s_t, a_t)$, and make action $a_{t+1} \sim \pi(\cdot|s_{t+1})$

Update:

$$\theta_{t+1} = \theta_t + \alpha_t \big[c(s_t, a_t) + \gamma\theta_t(s_{t+1}, a_{t+1}) + \sigma_{\mathcal{U}_{s_t,a_t}}(M(\pi, \theta_t)) - \theta_t(s_t, a_t)\big] e(s_t, a_t)$$

end for

- $\sigma_X(\cdot)$ is the support function of X, $[M(\pi, x)](s) = \sum_{a\in\mathcal{A}} \pi(a|s)x(s, a)$
- When $\mathcal{U} = \{\mathbf{0}\}$, reduces to standard TD

Learning the Robust Q-function

Exploiting Access to $\mathbb{P}_{\mathrm{N}}$

▷ **when $\mathcal{U}$ is known**

Algorithm Robust Temporal Difference Learning: $\pi \to Q_r^{\pi,\xi}$

for $t = 0, 1, \ldots$ **do**

Collect $s_{t+1} \sim \mathbb{P}_{\mathrm{N}}(\cdot|s_t, a_t)$, and make action $a_{t+1} \sim \pi(\cdot|s_{t+1})$

Update:

$$\theta_{t+1} = \theta_t + \alpha_t \Big[c(s_t, a_t) + \gamma \theta_t(s_{t+1}, a_{t+1}) + \sigma_{\mathcal{U}_{s_t, a_t}}(M(\pi, \theta_t)) - \theta_t(s_t, a_t) \Big] e(s_t, a_t)$$

end for

- $\sigma_X(\cdot)$ is the support function of X, $[M(\pi, x)](s) = \sum_{a \in \mathcal{A}} \pi(a|s) x(s, a)$
- When $\mathcal{U} = \{\mathbf{0}\}$, reduces to standard TD

▷ **when $\mathcal{U}$ is unknown**

1. Trivially extends to ϵ-contamination model
   - Unbiased robust Bellman evaluation operator is available
2. Can be extended to KL-divergence based $\mathcal{P}$
   - Dual representation + multi-level Monte Carlo (Liu et al. '22, Wang et al., '23)

Sample Complexity of RTD and SRPMD

▷ **Sample complexity of Robust TD**

Proposition

For any $\epsilon > 0$, with properly chosen $\{\alpha_t\}$, the RTD method needs at most

$$T = \widetilde{\mathcal{O}}\left(\frac{\log^2(1/\epsilon)}{(1-\gamma)^5\epsilon^2}\right)$$

iterations to find an estimate θ_T satisfying $\mathbb{E}_\xi\|\theta_T - Q_r^\pi\|_\infty \leq \epsilon$.

Sample Complexity of RTD and SRPMD

▷ **Sample complexity of Robust TD**

Proposition

For any $\epsilon > 0$, with properly chosen $\{\alpha_t\}$, the RTD method needs at most

$$T = \widetilde{\mathcal{O}}\left(\frac{\log^2(1/\epsilon)}{(1-\gamma)^5\epsilon^2}\right)$$

iterations to find an estimate θ_T satisfying $\mathbb{E}_\xi\|\theta_T - Q_r^\pi\|_\infty \leq \epsilon$.

▷ **Sample complexity of SRPMD**

Theorem

With the same stepsize chosen as before, total number of samples required by SRPMD for finding an ϵ-optimal policy can be bounded by

$$\widetilde{\mathcal{O}}\left(\frac{M^3\log^2\left(4M/(\epsilon(1-\gamma)^2)\right)}{(1-\gamma)^{10}\epsilon^2}\right).$$

- We believe the dependence on $(1-\gamma)^{-1}$ can be improved

Robust Policy Mirror Descent: (Linear) Function Approximation

Preview of Linear Approximation

▷ **The essential target:** Find θ^π so that

$$\|\underbrace{\phi(\cdot,\cdot)^\top\theta^\pi}_{Q^\pi_{\theta^\pi}} - Q^\pi_r(\cdot,\cdot)\|_\infty$$

can be controlled.

Isn't linear function approximation easy?

Preview of Linear Approximation

▷ **The essential target:** Find θ^π so that

$$\| \underbrace{\phi(\cdot,\cdot)^\top \theta^\pi}_{Q^\pi_{\theta^\pi}} - Q^\pi_r(\cdot,\cdot)\|_\infty$$

can be controlled.

Isn't linear function approximation easy?

1. Fixed-point (contraction) based:

$$Q^\pi_\theta = \Pi_{\phi,\nu}\mathcal{T}^\pi Q^\pi_\theta \;\rightarrow\; \theta^\pi$$

- $\mathcal{T}^\pi$ – Robust Bellman operator of Q^π_r
- $\Pi_{\phi,\nu}$ – the projection onto $\mathrm{span}(\Psi)$ in $\|\cdot\|_\nu$
- $\Pi_{\phi,\nu}\mathcal{T}^\pi$ – a contraction
- Roots of TD and many variants

Preview of Linear Approximation

▷ **The essential target:** Find θ^π so that

$$\| \underbrace{\phi(\cdot,\cdot)^\top \theta^\pi}_{Q^\pi_{\theta^\pi}} - Q^\pi_r(\cdot,\cdot)\|_\infty$$

can be controlled.

Isn't linear function approximation easy?

1. Fixed-point (contraction) based:

$$Q^\pi_\theta = \Pi_{\phi,\nu}\mathcal{T}^\pi Q^\pi_\theta \ \rightarrow \ \theta^\pi$$

- $\mathcal{T}^\pi$ – Robust Bellman operator of Q^π_r
- $\Pi_{\phi,\nu}$ – the projection onto $\mathrm{span}(\Psi)$ in $\|\cdot\|_\nu$
- $\Pi_{\phi,\nu}\mathcal{T}^\pi$ – a contraction
- Roots of TD and many variants

2. Minimize Bellman residual:

$$\min_\theta \|Q^\pi_\theta(\cdot,\cdot) - \mathcal{T}^\pi Q^\pi_\theta(\cdot,\cdot)\|_2^2 \ \rightarrow \ \theta^\pi$$

- Easily combined and nonlinear approximations (e.g., NNs)

Difficulties of Linear Approximation

Why is linear function approximation difficult (for robust evaluation)?

Difficulties of Linear Approximation

Why is linear function approximation difficult (for robust evaluation)?

1. Fixed-point (contraction) based:

$$Q^{\pi}_{\theta} = \Pi_{\phi,\nu}\mathcal{T}^{\pi}_{\text{robust}}Q^{\pi}_{\theta} \not\to \theta^{\pi}$$

- $\mathcal{T}^{\pi}_{\text{robust}}$ – Bellman operator of Q^{π}
- $\Pi_{\phi,\nu}\mathcal{T}^{\pi}_{\text{robust}}$ – NOT a contraction
- Does not even have a solution
- Robust TD diverges with linear approximation

Difficulties of Linear Approximation

Why is linear function approximation difficult (for robust evaluation)?

1. Fixed-point (contraction) based:

$$Q_\theta^\pi = \Pi_{\phi,\nu} \mathcal{T}_{\text{robust}}^\pi Q_\theta^\pi \not\to \theta^\pi$$

- $\mathcal{T}_{\text{robust}}^\pi$ – Bellman operator of Q^π
- $\Pi_{\phi,\nu} \mathcal{T}_{\text{robust}}^\pi$ – NOT a contraction
- Does not even have a solution
- Robust TD diverges with linear approximation

2. Minimize Bellman residual:

$$\min_\theta \|Q_\theta^\pi(\cdot,\cdot) - \mathcal{T}_{\text{robust}}^\pi Q_\theta^\pi(\cdot,\cdot)\|_2^2 \not\to \theta^\pi$$

- Non-convex in θ

Difficulties of Linear Approximation

Why is linear function approximation difficult (for robust evaluation)?

1. Fixed-point (contraction) based:

$$Q^{\pi}_{\theta} = \Pi_{\phi,\nu}\mathcal{T}^{\pi}_{\text{robust}}Q^{\pi}_{\theta} \not\to \theta^{\pi}$$

- $\mathcal{T}^{\pi}_{\text{robust}}$ – Bellman operator of Q^{π}
- $\Pi_{\phi,\nu}\mathcal{T}^{\pi}_{\text{robust}}$ – NOT a contraction
- Does not even have a solution
- Robust TD diverges with linear approximation

2. Minimize Bellman residual:

$$\min_{\theta} \|Q^{\pi}_{\theta}(\cdot,\cdot) - \mathcal{T}^{\pi}_{\text{robust}}Q^{\pi}_{\theta}(\cdot,\cdot)\|_2^2 \not\to \theta^{\pi}$$

- Non-convex in θ

Current Development

No assumption-free convergent method for robust policy evaluation with linear approximation even in the deterministic setting

Robust Evaluation as Policy Optimization

▷ **MDP of Nature:**

- State space: $\mathcal{S} \times \mathcal{A}$
- Action space: $\mathcal{P}_{s,a}$ for each (s, a)
- Transition: transition of $\{(s_t, a_t)\}$ generated by π deployed in $\mathbb{P}$, where $\mathbb{P}$ is determined by nature's policy
- Cost: $-c(s, a)$

Robust Evaluation as Policy Optimization

▷ **MDP of Nature:**

- State space: $\mathcal{S} \times \mathcal{A}$
- Action space: $\mathcal{P}_{s,a}$ for each (s, a)
- Transition: transition of $\{(s_t, a_t)\}$ generated by π deployed in $\mathbb{P}$, where $\mathbb{P}$ is determined by nature's policy
- Cost: $-c(s, a)$

▷ **Observation:** optimal value function of nature equals to $-Q^{\pi}_{r}(s, a)$

Question: can we optimize nature's MDP efficiently?

Robust Evaluation as Policy Optimization

▷ **MDP of Nature:**

- State space: $\mathcal{S} \times \mathcal{A}$
- Action space: $\mathcal{P}_{s,a}$ for each (s, a)
- Transition: transition of $\{(s_t, a_t)\}$ generated by π deployed in $\mathbb{P}$, where $\mathbb{P}$ is determined by nature's policy
- Cost: $-c(s, a)$

▷ **Observation:** optimal value function of nature equals to $-Q_r^\pi(s, a)$

Question: can we optimize nature's MDP efficiently?

▷ **Computational challenges at the first sight:**

1. Continuous action space
2. We do not want to parameterize the policy
   - Essentially this requires saving the model (model-based)

Robust Evaluation as Policy Optimization

▷ **MDP of Nature:**

- State space: $\mathcal{S} \times \mathcal{A}$
- Action space: $\mathcal{P}_{s,a}$ for each (s, a)
- Transition: transition of $\{(s_t, a_t)\}$ generated by π deployed in $\mathbb{P}$, where $\mathbb{P}$ is determined by nature's policy
- Cost: $-c(s, a)$

▷ **Observation:** optimal value function of nature equals to $-Q_r^\pi(s, a)$

Question: can we optimize nature's MDP efficiently?

Yes, $\mathcal{O}(1/\epsilon^2)$ sample suffices, even with linear approximation.

Also can be incorporated with NNs.

The method does not parameterize the policy of nature (model-free).

Summary

1. RPMD for robust MDP with (s, a)-rectangular ambiguity
   - Simple implementation
   - Subsumes planning of non-robust MDP
2. Deterministic setting: $\mathcal{O}(\log(1/\epsilon))$ iterations
3. Stochastic setting:
   - Convergence up to noise level
   - $\widetilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity
4. Evaluation with linear approximation:
   - $\mathcal{O}(1/\epsilon^2)$ sample complexity

Summary

1. RPMD for robust MDP with (s, a)-rectangular ambiguity
   - Simple implementation
   - Subsumes planning of non-robust MDP
2. Deterministic setting: $\mathcal{O}(\log(1/\epsilon))$ iterations
3. Stochastic setting:
   - Convergence up to noise level
   - $\widetilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity
4. Evaluation with linear approximation:
   - $\mathcal{O}(1/\epsilon^2)$ sample complexity
5. **Potential directions:**
   - Sample limit of policy gradients for robust MDP
     – dependence on the effective horizon (lower/upper bounds)
   - s- and r-rectangular ambiguity sets

Summary

1. RPMD for robust MDP with (s, a)-rectangular ambiguity
   - Simple implementation
   - Subsumes planning of non-robust MDP
2. Deterministic setting: $\mathcal{O}(\log(1/\epsilon))$ iterations
3. Stochastic setting:
   - Convergence up to noise level
   - $\widetilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity
4. Evaluation with linear approximation:
   - $\mathcal{O}(1/\epsilon^2)$ sample complexity
5. **Potential directions:**
   - Sample limit of policy gradients for robust MDP
     – dependence on the effective horizon (lower/upper bounds)
   - s- and r-rectangular ambiguity sets

Reference

- Li, Y., Lan, G, & Zhao, T. (2022). First-order policy optimization for robust Markov decision process. arXiv preprint arXiv:2209.10579.
- Li, Y., & Lan, G. (2023). First-order policy optimization for robust policy evaluation. arXiv preprint arXiv:2307.15890.