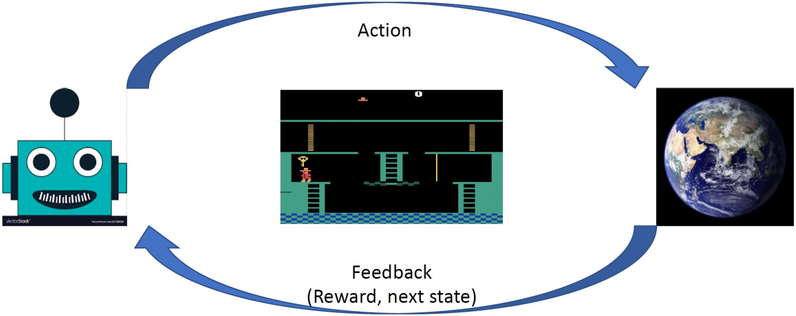
Provably Robust Offline Reinforcement Learning via Smoothed Policy Iteration

Tengyang Xie

tx10@illinois.edu

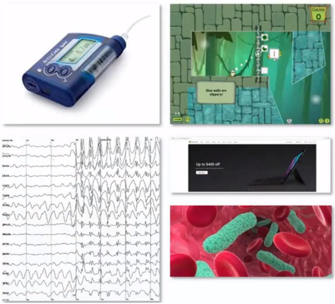
Reinforcement Learning



Learning to Make Good Decisions under Uncertainty

Motivations

- RL has been used in many *safety-critical* applications (e.g., medical treatment design, autonomous driving).



Motivations

- RL has been used in many *safety-critical* applications (e.g., medical treatment design, autonomous driving).
- Observation from real-world applications contains unavoidable perturbation
  (e.g., sensor errors or adversarial attack).

76% it is a

45 MPH Sign

Motivations

- RL has been used in many *safety-critical* applications (e.g., medical treatment design, autonomous driving).
- Observation from real-world applications contains unavoidable perturbation
  (e.g., sensor errors or adversarial attack).

Goal: Robust RL against (adversarial) perturbations on state observations.

Problem Setup

- Typical RL: Observe state s, take action $a \sim \pi(\cdot|s)$, receive reward r and next state s'.
- This paper: Observe perturbed state $s+\nu(s)$, take action $a \sim \pi(\cdot|s+\nu(s))$, receive reward r and next perturbed state $s'+\nu(s')$.

($\nu(s)$ — arbitrary L^2 bounded perturbation)

Challenge

For any given π, what we actually executed is $\pi \circ \nu$.

Because the observation can be always perturbed like this —

Proposed Method

Policy Iteration + Lipschitz

Why policy iteration (PI)?

PI is the prototype of all policy-based methods, e.g., actor-critic, TRPO, PPO...

Basic framework of PI: At every iteration t, repeat

(1) Estimate Q^{π_t}.

(2) Set the greedy policy of Q^{π_t} as π_{t+1}.

Proposed Method

Policy Iteration + Lipschitz

Why Lipschitz?

Lipschitz continuity measures the robustness for supervised learning [Salman et al., 2019; Bubeck et al., 2020].

Open Question:

Can this principle (Lipschitz $\Rightarrow$ robustness) apply to RL? How?

Our Algorithm

Given perturbed batch data $\mathcal{D} = \{s_i + \nu(s_i), a_i, s_i' + \nu(s_i'), r_i\}_{i=1}^n$, where $r_i \sim \mathcal{R}(s, a)$ and $s' \sim \mathcal{P}(\cdot|s, a)$.

1. Initialize policy as π_1.
2. For $t = 1, 2, \ldots, T$, do
    2.1 Estimate Q^{π_t} as f_t using $\mathcal{D}$, where f_t satisfies i) $\|f_t - Q^{\pi_t \circ \nu}\|_\infty = \varepsilon_1$ and, ii) f_t is L-Lipschitz.
    2.2 $\pi_{t+1} \leftarrow$ greedy policy of f_t.
3. Output π_{T+1}

Robust Policy Improvement

Theorem (Robust Policy Improvement). Assume (Q, π) pair satisfies:

(i) $\|Q - Q^{\pi}\|_{\infty} = \varepsilon_1$.

(ii) Q is L-Lipschitz.

Let $\pi_Q \circ \nu_{\varepsilon_2}$ be the perturbed greedy policy w.r.t. Q and an ε_2-bounded perturbation ν_{ε_2}, then for any $s \in \mathcal{S}$,

$$V^{\pi_Q \circ \nu_{\varepsilon_2}}(s) \geq V^{\pi}(s) - \frac{2\varepsilon_1 + 2L\varepsilon_2}{1-\gamma},$$

where $\pi_Q \circ \nu_{\varepsilon_2}(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q(\nu_{\varepsilon_2}(s), a)$.

Performance Bound

Theorem (Performance Bound). Let π_{T+1} be the out put of our algorithm. Then,

$$
\begin{aligned}
& J(\pi^\star) - J(\pi_{T+1}) \\
& \leq \gamma^T V_{\max} + \frac{1-\gamma^T}{1-\gamma} \left(2L\varepsilon_2 + 2\varepsilon_1 + \frac{\gamma(2\varepsilon_1 + 2L\varepsilon_2)}{1-\gamma} \right),
\end{aligned}
$$

where ε_1 is the estimation error in the policy evaluation step ($\varepsilon_1 = \varepsilon_Q + O(n^{-1/2})$ or $\varepsilon_1 = O(n^{-1/d})$ depends on the evaluation algorithm), and ε_2 is the perturbation scale.

Policy Evaluation Subroutine

Why Lipschitz policy evaluation (PE) is doable?

Lipschitz MDP $\Longrightarrow$ Lipschitz Q-Function [Asadi et al., 2018]

Options for us:

(i) Non-parametric policy evaluation over a Lipschitz function class.

(ii) General function class + randomized smoothing.

(i) Non-Parametric Policy Evaluation over a Lipschitz Function Class

Inspired by [Tang et al., 2020].

Algorithm:

Given batch data $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{n}$ (all the states are perturbed by ν_{ε_2}) and target policy π.

1. Set $q_{0,i} = r_i - \gamma L d(s_i, s_i'), \forall i \in [i]$
2. For $t = 1, 2, \ldots, T$
   - 2.1 $Q_t(s, a) = \max_{i \in [n]: a_i = a} (q_{t-1,i} - Ld(s, s_i) - L\varepsilon_2), \ \forall (s, a) \in \mathcal{S} \times \mathcal{A}$
   - 2.2 $q_{t,i} = r_i + \gamma [\widehat{T}^{\pi} Q_t](s_i, a_i), \forall i \in [i]$

(i) Non-Parametric Policy Evaluation over a Lipschitz Function Class

Theorem

Let $Q = Q_T$, $T \to \infty$, *we have*

$$\|Q_T - Q^\pi\|_\infty \leq \frac{2L(\varepsilon_{\mathcal{S}} + \varepsilon_2)}{1-\gamma},$$

where $\varepsilon_{\mathcal{S}} \coloneqq \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \min_{i\in[n]:a_i=a} d(s, s_i)$, *and* ε_2 *is the perturbation bound.*

Open Issue: $\varepsilon_{\mathcal{S}} = O(1/n^{1/d})$ — inevitably curse of dimensionality for non-parametric approaches

(ii) General Function Class + Randomized Smoothing

Lipschitz constant is controlled by $O(V_{\max}/\sigma)$.

Theorem. For any function $\mathcal{F}$, if $f(s, a) \in [0, V_{\max}]$ for any $f \in \mathcal{F}$ and any $(s, a) \in \mathcal{S} \times \mathcal{A}$, then $f \circ \mathcal{N}(0, \sigma I)$ is $\frac{V_{\max}}{\sigma}\sqrt{2/\pi}$-Lipschitz.

Approximation error scales with $O(\sigma)$.

(ii) General Function Class + Randomized Smoothing

Lipschitz constant is controlled by $O(V_{\max}/\sigma)$.

Approximation error scales with $O(\sigma)$.

Theorem. Let the smoothed function class $\widetilde{\mathcal{F}}$ be defined by

$$\widetilde{\mathcal{F}} = \{f \circ \mathcal{N}(0, \sigma I); f \in \mathcal{F}\},$$

If $Q^\pi \in \mathcal{F}$, then

$$\min_{\widetilde{f} \in \widetilde{\mathcal{F}}} \left\| \widetilde{f} - \mathcal{T}^\pi \widetilde{f} \right\|_\infty \leq L_R \sigma \sqrt{\frac{2}{\pi}} + (L_{\mathcal{P}} + 1) L_Q \sigma \sqrt{\frac{2}{\pi}}.$$

(ii) General Function Class + Randomized Smoothing

Lipschitz constant is controlled by $O(V_{\max}/\sigma)$.

Approximation error scales with $O(\sigma)$.

Theorem[a] (informal). Let π_{T+1} be the output of the algorithm and the policy evaluation subroutine uses the randomized smoothed function class.
If $\sigma = \Theta(\sqrt{V_{\max}\varepsilon_2})$ (ε_2 is the perturbation bound), then,

$$J(\pi^\star) - J(\pi_{T+1}) \le O\left(n^{-1/2} + \sqrt{V_{\max}\varepsilon_2}\right).$$

[a] proof uncompleted due to limited time

Conclusion

We provides a novel RL algorithm that:

- Provably robust against against perturbations on state observations.
- The performance is guaranteed with a finite-sample error bounds.

Messages from our results:

- The principle of Lipschitz $\Rightarrow$ robustness also applies to RL indeed.
- Randomized smoothing still works well in RL against perturbations on state observations.

Next Plan

- Complete the current randomized smoothing part.
- The theoretical analysis can be improved by $\|\cdot\|_\infty \to C\|\cdot\|_\mu$[1], which is more practical for capturing rich observation with function approximation.
- Experiments.

[1] C is the concentrability coefficient for capturing the distribution shift.

Reference I

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273, 2018.

Sébastien Bubeck, Yuanzhi Li, and Dheeraj Nagaraj. A law of robustness for two-layers neural networks. *arXiv preprint arXiv:2009.14444*, 2020.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

Reference II

Ziyang Tang, Yihao Feng, Na Zhang, Jian Peng, and Qiang Liu. Off-policy interval estimation with lipschitz value iteration. *Advances in Neural Information Processing Systems*, 33, 2020.