Robust Q-Learning under Corrupted Rewards

Sreejeet Maity   Aritra Mitra
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh

NC STATE UNIVERSITY Electrical and Computer Engineering

63rd IEEE Conference on Decision and Control 2024

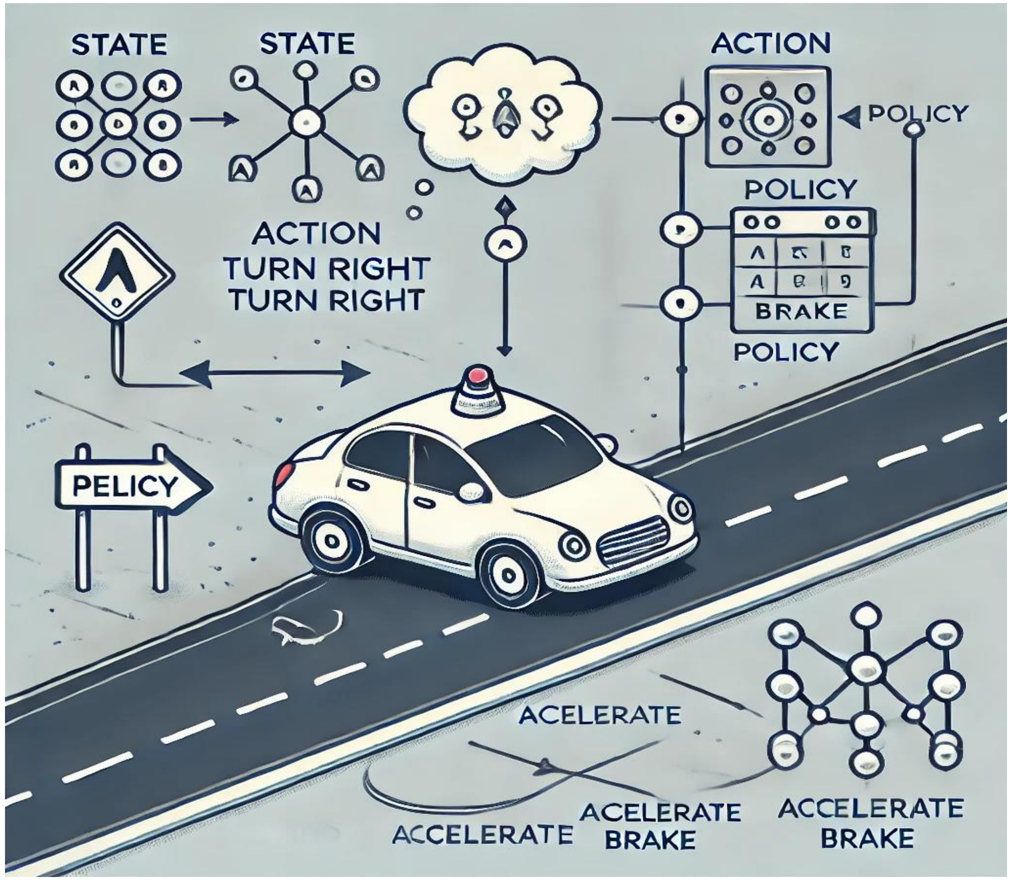
CDC·2024 MILANO

Are RL Algorithms always trustworthy ?

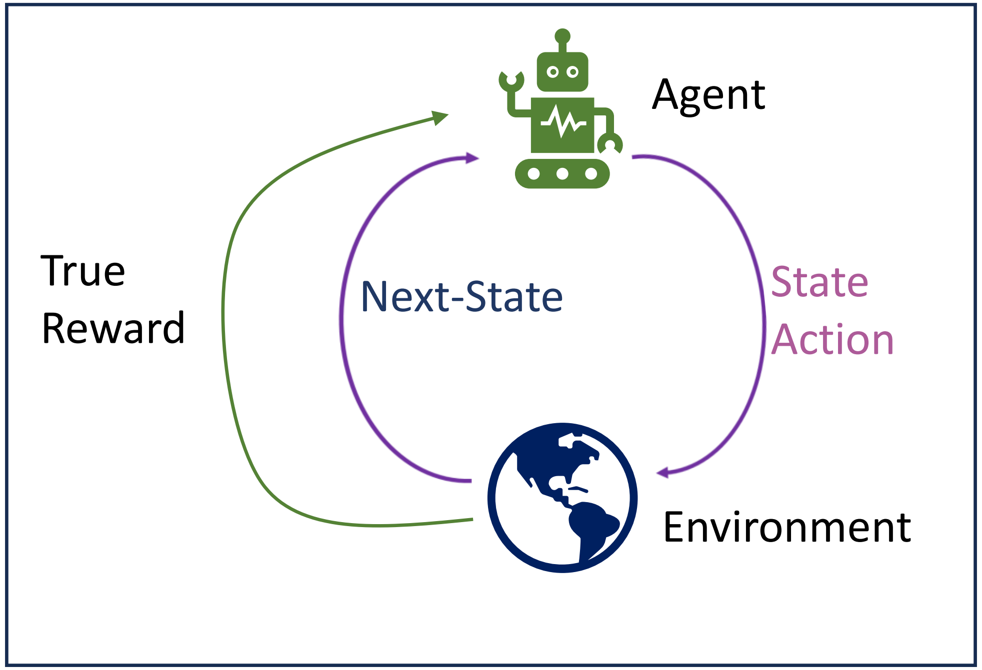
- How 'reliable' are RL Algorithms in real-world environments ?

- How critical is the assumption of "perfect" feedback in practical scenarios?

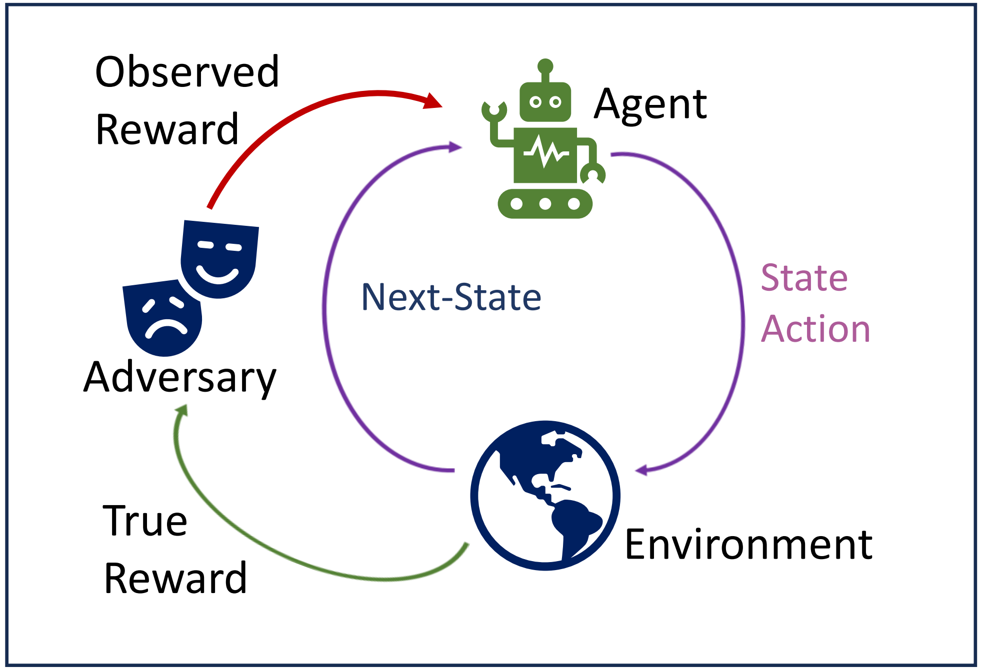
- Consequences of having blind faith in the correctness of feedback in safety-critical applications ?



Reinforcement Learning with Corrupted Rewards



Standard RL Pipeline



RL Pipeline with Corrupted Rewards

Brief Outline of the Talk

- **Contribution 1:** Vulnerability of the classical Q-Learning Algorithm (*Watkins et al.*, Machine Learning, Vol 8, 1992) with adversarial corruptions in rewards.

- **Contribution 2:** Design of a novel, Robust Q-Learning Algorithm to safeguard against corrupted rewards (adversary corrupts a fraction of rewards).

- **Contribution 3:** Finite-time analysis ***achieving near-optimal bounds*** with a small additive error proportional to corruption fraction.

Basic RL Setup

- We consider an MDP $\mathcal{M} = (S, A, P, R, \gamma)$ with finite state and action spaces.
- $R(s, a)$ is the immediate expected reward at state-action pair (s, a).
- $P(s'|s, a)$ is the probability of transitioning from s to s' under action a.
- A deterministic policy $\pi: S \mapsto A$.

- The "goodness" of a policy π is captured by the value function $V_\pi: S \mapsto \mathbb{R}$:

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t) | s_0 = s\right]$$

Goal: Find optimal policy (π^*) that maximizes value function (without the knowledge of MDP). *One of most popular algorithm for this is Q-Learning.*

State-Action Value Function (Q-Function)

- The state-action value function $Q_\pi : S \times A \mapsto \mathbb{R}$:

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t) | s_0 = s\,, a_0 = a\right]$$

Let $Q_{\pi^*} = Q^*$ be the optimal state-value function.

- Then Q^* is the unique fixed point of the Bellman optimality operator $\mathcal{T}^*$

$$\boxed{(\mathcal{T}^* Q)(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)\, max_{a' \in A} Q(s', a')}$$

Also, $||\mathcal{T}^* Q_1 - \mathcal{T}^* Q_2||_\infty \leq \gamma ||Q_1 - Q_2||_\infty$ for $Q_1, Q_2 \in \mathbb{R}^{|S||A|}$, $0 < \gamma < 1$.

Synchronous Q-Learning

At each iteration $t \in [T]$, for all (s, a), we observe:

- A new state $s_t(s, a) \sim P(.|s, a)$ (drawn independently).
- A stochastic reward $r_t(s, a) \sim \mathcal{R}(s, a)$ (drawn independently).

Reward Model

- Unbiased and Light Tailed.

Update Rule:

$$\boxed{Q_{t+1}(s, a) = (1-\alpha)Q_t(s, a) + \alpha\,[r_t(s, a) + \gamma \max_{a' \in A} Q_t(s_t(s, a), a')]}$$

Related Works

Asymptotic Analysis:

1) *Vivek S Borkar,* Springer, 2009.
2) *John N Tsitsiklis,* Machine Learning, 1994.
3) *Csaba Szepesvári,* NeuRIPS, 1997.

Finite-time Analysis (Our setting) :

1) *J Wainwright,* Arxiv, 2019.
2) *Wiermann and Qu,* PMLR, 2020.

Note: All the works assume *rewards* are sampled from the true distribution.

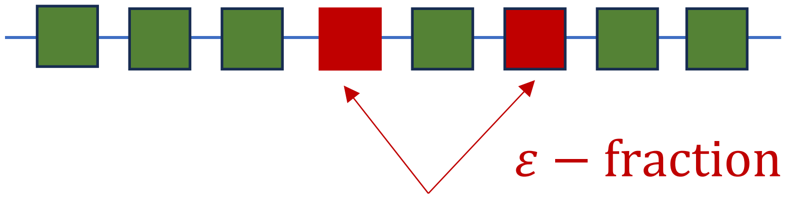
Heavy-Tailed Distribution $\mathcal{R}(s, a)$

In our setting, we further relaxed the standard assumptions on the true reward distribution. Assuming only the finiteness of second-moment, the infinite support reward distribution $\mathcal{R}(s, a)$ can potentially be

- Heavy-Tailed!

Strong Adversarial Corruption in Rewards

- The adversary observes the entire reward set $\{r_t(s,a)\}_{(s,a)\in S\times A}$ in each iteration t.
- Perturb $\varepsilon \in \left[0, \frac{1}{2}\right)$ fraction of these observed rewards up to t.



Strong Adversary:

- Has complete access to the MDP.
- Has access to all the observations.

Vulnerability of Q-Learning

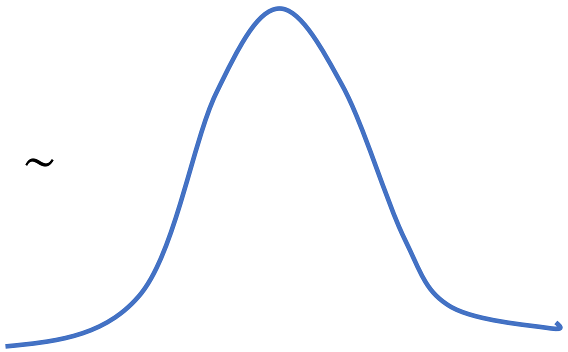
Assume an adversary *corrupts the reward at each iteration with probability ε* (Huber Contamination).

Observed Rewards

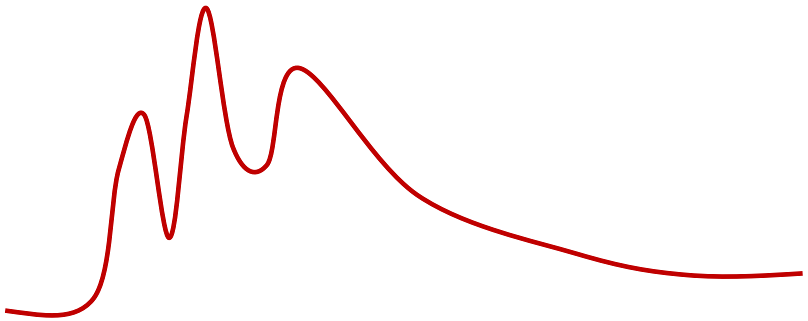
$$y_t(s,a) \sim (1-\varepsilon)\mathcal{R}(s,a) + \varepsilon\, \mathcal{C}(s,a)$$

$$\mathbb{E}\,[y_t(s,a)] = R_c(s,a)$$

$\sim$



$\oplus$



True Reward with prob. (1- ϵ)
$\mathcal{R}(s,a)$

Corrupted Reward with prob. (ϵ)
$\mathcal{C}(s,a)$

Vulnerability of Q-Learning

Under a weaker corruption model, assume an adversary *corrupts the reward at each iteration with probability ε* (Huber Contamination).

Theorem 1: Under Huber contamination, and a suitable step size α, with probability 1, $Q_t \to \widetilde{Q_c^*}$, where $\widetilde{Q_c^*}$ is the unique fixed point of the perturbed Bellman operator $\mathcal{T}_c^*$, satisfying:

$$(\mathcal{T}_c^* Q)(s,a) = R_c(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)\, max_{a' \in A} Q(s',a')$$

$$R_c(s,a) = (1-\varepsilon) R(s,a) + \varepsilon\, C(s,a)$$

This is explicitly controlled by the adversary!

Vulnerability of Q-Learning (contd.)

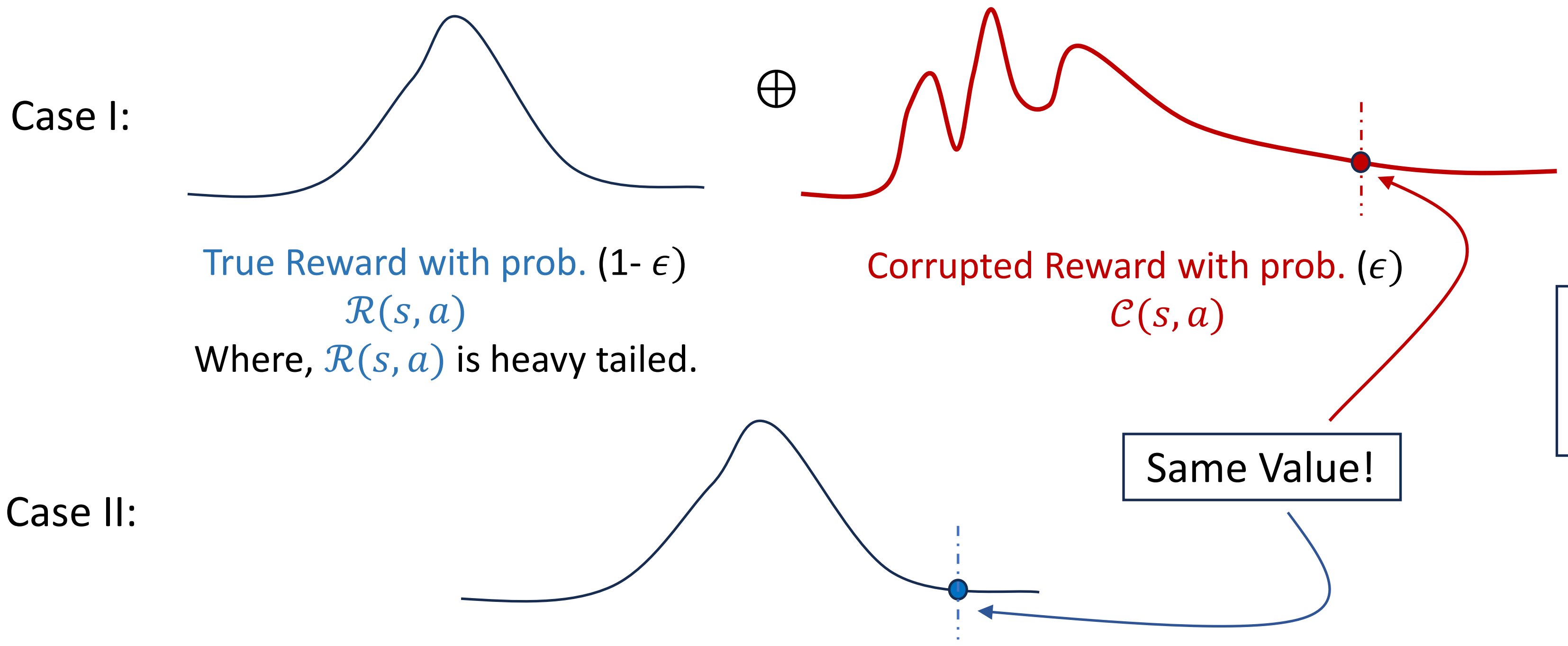
Can the gap between the true optimal Q^* and $\widetilde{Q_c^*}$ be arbitrarily large?

Yes!

Theorem 2: There exists an MDP with finite state-action spaces for which the gap $\left\| Q^* - \widetilde{Q_c^*} \right\|_\infty$ can be arbitrarily large under the Huber Corruption Model.

Proof: Details of MDP construction in our paper.

What makes Robust Q-Learning hard ?



Can we always differentiate between $\bullet$, and $\bullet$?

Proposed Robust Q-Learning Update Rule

- Vanilla Q-Learning is provably susceptible against adversarial corruptions (*Theorem* 1,2).

- Proposed Robust Q-Learning Update Rule:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha[\tilde{r}_t(s,a) + \gamma \max_{a' \epsilon A} Q_t(s_t(s,a),a')]$$

Proposed Reward Proxy

Robust Q-Learning Algorithm

- Brief outline of our Algorithm is as follows (details in paper):

Iteratively, for all $(s, a) \in S \times A$, and t ∈ [T]

Inputs

- Q_0.
- Step Size α.
- Conf. Par δ.
- Corr. Fraction ε
- Iterations T.

- Sample $s_t(s,a) \sim P(.|s,a)$.
- Observe reward $y_t(s,a)$.
- $\widetilde{r}_t(s,a)$ = TRIM$[\{y_k(s,a)\}_{k\in[t]}, \delta_1, \varepsilon]$

Where, $\delta_1 = \delta/(2|S||A|T)$.

$\widetilde{r}_t(s,a)$

Threshold? (Yes/No)

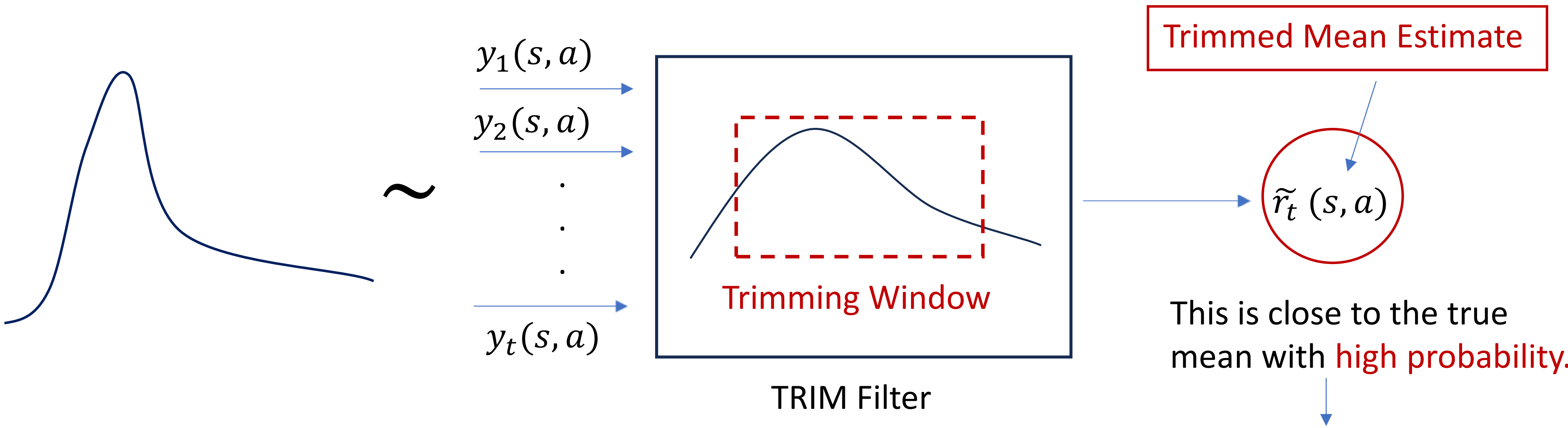
$\widetilde{r}_t(s,a)$

Update Eq. (1)

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha[\widetilde{r}_t(s,a) + \gamma \, max_{a' \epsilon A} Q_t(s_t(s,a), a')] \quad (1)$$

Key Algorithmic Components : TRIM

- $\widetilde{\mathrm{r}}_{\mathrm{t}}(\mathrm{s}, \mathrm{a})$ is the output of the `TRIM` Filter.
- For each set of reward obs. $[y_k\ (s, a)]_{k\in[t]}$, compute the Trimmed Mean Estimate from *Lugosi et al*, The Annals of Statistics, 2021 .



Is Robust Mean enough to ensure the convergence of Robust Q-Learning ? No.

Key Algorithmic Components: Thresholding

Is Trimmed Mean enough to ensure the convergence of Robust Q-Learning ? No.

- The TRIM output is close to the *true mean* with high probability.
- But what happens on extreme events ? Output of TRIM can be unbounded!
- For finite-time analysis, the iterates $\{Q_t\}_{t\in[T]}$ need to be uniformly bounded.
- Hence, we need to ensure the reward proxy to be bounded.

How to ensure that?

- First, we will design a novel conditional threshold for the TRIM output .
- In the "good" events (TRIM Output < Threshold), we won't threshold the TRIM output .
- Rather, we only need to threshold the TRIM output in "extreme" events.

Key Algorithmic Components: Thresholding

- The robust mean estimator works for $t \geq T_{lim} = \left\lceil 2\, log\left(\frac{4}{\delta_1}\right)\right\rceil$.
- Also, the guarantee holds with high probability.

How to design the threshold for the extreme events ?

Upper Bound on Reward Mean

Where, $|R(s,a)| \leq \mu$
$\forall\ (s,a) \in S \times A$, and $\mu \geq 1$

We design a curated deterministic threshold for all such outliers. Setting, $\mathcal{R}$ = max{ μ , σ }.

The threshold is motivated by the guarantees of *Trimmed Mean* described before.

$$G_t = \begin{cases} 2\,\mathcal{R}\,, & 0 \leq t \leq T_{lim} \\ C\mathcal{R}\left(\sqrt{\frac{log\left(\frac{4}{\delta_1}\right)}{t}}\right) + \mathcal{R}, & t \geq T_{lim} + 1 \end{cases}$$

Upper Bound on Reward SD

$\mathbb{E}_{r(s,a)\sim\mathcal{R}(s,a)}\left[\left(r(s,a) - R(s,a)\right)^2\right] \leq \sigma^2$
$\forall(s,a) \in S \times A$

Main Result

Theorem 3: With corruption fraction $\varepsilon \in [0, 1/16)$, failure probability $\delta \in (0, 1)$, and a suitable step-size α, the output of our Algorithm satisfies with probability at least 1- δ:

Corruption Fraction

$$||Q_T - Q^*||_\infty \le \underbrace{\frac{||Q_0 - Q^*||_\infty}{T} + O\left(\frac{\mathcal{R}}{(1-\gamma)^{\frac{5}{2}}} \frac{\log T}{\sqrt{T}} \sqrt{\log\left(\frac{|S||A|T}{\delta}\right)}\right.}_{\text{Matches the bound in Wainwright, Qu}} \left. + \underbrace{\frac{\mathcal{R}\sqrt{\varepsilon}}{1-\gamma}}_{\text{Additive Error Term}} \right)$$

Key Takeaways:

- Despite strong corruption, our proposed Algorithm achieves near-optimal, high probability l_∞ – error rate of $\tilde{O}\left(\frac{1}{\sqrt{T}}\right) + O(\sqrt{\varepsilon})$.
- In absence of corruption, it matches the prior established results.

Summary

- The standard Q-Learning is vulnerable to adversarial reward corruption (*Theorem 1, Theorem 2*).
- We propose a provably robust Q-Learning algorithm (Algorithm 2) with theoretical guarantees (*Theorem 3*).
- Our algorithm only requires the existence of second moments, allowing for rewards with infinite support and heavy tails!
- We conjecture that the additive error, dependent on corruption fraction, is unavoidable.

Future Directions:

- Extension to Asynchronous Q-Learning Algorithm.
- Lower Bounds to show the dependence on ε is unavoidable.
- Can we extend the algorithm when there is limited knowledge about the underlying reward distribution.
- What are some other possible adversaries ? State Adversaries.