Robust Algorithms for Reinforcement Learning with Adversarial Rewards

Sreejeet Maity Aritra Mitra
North Carolina State University, Raleigh

Abstract

Recently, there has been a growing interest in understanding the non-asymptotic performance of model-free reinforcement learning algorithms. Despite this, the behavior of these algorithms in real-world settings, particularly in the presence of adversarial or corrupted rewards, remains poorly understood. In this work, we introduce a novel robust Q-learning algorithm that achieves near-optimal performance, with an unavoidable error term of $O(\varepsilon)$.

Vulnerability of the Q-Learning Algorithm

Are existing model-free RL algorithms such as Q-Learning robust to reward perturbations ?

Vanilla Q-Learning Algorithm

$$Q_{t+1}(s,a) = (1-\alpha_t)Q_t(s,a) + \alpha_t\left[r_t(s,a) + \gamma \max_{a' \in \mathcal{A}} Q(s_t(s,a),a')\right]. \tag{1}$$

Strong Contamination Attack Model. We consider an adversary with full knowledge of the MDP and the learner's observations at each iteration. The adversary can arbitrarily perturb the entire reward set $\{r_t(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sim \mathcal{R}(s,a)$ in each iteration t, with an attack budget $\varepsilon \in [0,1/2)$, allowing corruption in at most an ε-fraction of the first t iterations. When corrupted, the observed reward for state-action pair (s,a) in iteration t becomes $r_t^c(s,a) \sim \mathcal{R}_c(s,a)$, and it equals to $r_t(s,a) \sim \mathcal{R}(s,a)$ when there is no corruption. To accommodate both scenarios, we define $y_t(s,a)$ as the observed reward, which can either be $r_t^c(s,a)$ or $r_t(s,a)$, depending on whether an attack is present or not.

Theorem 1

Consider the vanilla synchronous Q-learning algorithm in Eq. (1) *with rewards perturbed based on the attack model described above. If the step-size sequence satisfies $\alpha_t \in (0,1)$ with $\sum_{t=1}^{\infty}\alpha_t = \infty$ and $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$, then with probability 1, $Q_t \to \tilde{Q}_c^*$, where $\tilde{Q}_c^*$ is the unique fixed point of the perturbed Bellman operator $\bar{\mathcal{T}}_c^* : \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ defined by*

$$(\bar{\mathcal{T}}_c^* Q)(s,a) = \tilde{R}_c(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q(s',a')\right], \tag{2}$$
$$\texttt{Given:}\ \bar{\mathcal{T}}_c^* Q_t \triangleq (\bar{\mathcal{T}}_c^* Q)(s,a) \triangleq y_t(s,a) + \gamma \max_{a'\in\mathcal{A}} Q(s_t(s,a),a').$$

Problem of Interest

Given the strong-contamination reward attack model and a failure probability $\delta \in (0,1)$, our goal is to generate a robust estimate Q_t of the optimal value function Q^* such that with probability at least $1-\delta$, the ℓ_∞-error $\|Q_t - Q^*\|_\infty$ is bounded from above by an error-function $e(t,\varepsilon)$ that has optimal dependence on the number of iterations t and corruption fraction ε.

Design of `TRIM` Filter

Algorithm 1 Univariate Trimmed-Mean Estimator

Require: Corrupted data set $Z_1,\dots,Z_{M/2}$, $\tilde{Z}_1,\dots\tilde{Z}_{M/2}$, corruption fraction ε, and confidence level δ.

1: Set $\zeta = 8\varepsilon + 24\frac{\log(4/\delta)}{M}$.
2: Let $Z_1^* \le Z_2^* \le \cdots \le Z_{M/2}^*$ represent a non-decreasing arrangement of $\{Z_i\}_{i\in[M/2]}$. Compute quantiles: $\gamma = Z^*_{\zeta M/2}$ and $\beta = Z^*_{(1-\zeta)M/2}$.
3: Compute robust mean estimate $\hat{\mu}_M$ as follows:

$$\hat{\mu}_T = \frac{2}{M}\sum_{i=1}^{M/2}\phi_{\gamma,\beta}(\tilde{Z}_i); \phi_{\gamma,\beta}(x) = \begin{cases}\beta & x > \beta \\ x & x \in [\gamma,\beta] \\ \gamma & x < \gamma\end{cases}$$

Theorem 2

Consider the trimmed mean estimator in Algorithm 1. Suppose $\varepsilon \in [0,1/16)$, and let $\delta \in (0,1)$ be such that $\delta \ge 4e^{-M/2}$. Then, there exists an universal constant C, such that with probability at least $1-\delta$,

$$|\hat{\mu}_Z - \mu_Z| \le C\sigma_Z\left(\sqrt{\varepsilon} + \sqrt{\frac{\log(4/\delta)}{M}}\right). \tag{3}$$

Reward-Filtering: Instead of directly using the observed reward $y_t(s,a)$, we compute a robust estimate of the expected reward $R(s,a)$ using a trimmed-mean estimator. The output, $\tilde{r}_t(s,a)$, represents the robust reward estimate. In our setting, we have assumed that there is an upper bound, denoted by $\mathcal{R}$, on both the mean and variance of the rewards for all state-action pairs. *Apart from that, we do not assume anything else on the reward distributions.*

Reward-Thresholding: If $\tilde{r}_t(s,a)$ exceeds G_t, it is constrained to maintain boundedness. This thresholding is critical to ensuring that $\tilde{r}_t(s,a)$ closely follows the true expected rewards with high probability, enabling tight performance guarantees.

$$T_{lim} \triangleq \left\lceil 2\log\left(\frac{4}{\delta_1}\right)\right\rceil. \tag{4}$$

However, the guarantee is not deterministic; instead, it only holds with high probability. For technical reasons that will become apparent later in our analysis, we need the sequence $\{Q_t\}$ of iterates generated by our algorithm to be uniformly bounded deterministically. However, the above discussion suggests that simply using the output of the `TRIM` function to perform updates will not suffice to achieve this goal. As such, we employ a second layer of safety by carefully defining a threshold function as follows:

$$G_t = \begin{cases} 2\mathcal{R} & 0 \le t \le T_{lim} \\ C\mathcal{R}\left(\sqrt{\frac{\log\left(\frac{4}{\delta_1}\right)}{t}} + \sqrt{\varepsilon}\right) + \bar{r}, & t \ge T_{lim}+1 \end{cases} \tag{9}$$

where C is the universal constant from our analysis. We use a thresholding operation to control the output of the `TRIM` function when it exceeds G_t, ensuring bounded iterates. For $t \le T_{lim}$, we use a simple bound $\mathcal{R}$ due to insufficient data. Once enough samples are collected, `TRIM` is expected to approximate $R(s,a)$ closely, and for $t \ge T_{lim}+1$, G_t is set based on this approximation. This ensures that the conservative estimate is rarely needed, allowing $\tilde{r}_t(s,a)$ to be used effectively, which is crucial for achieving accurate performance rates.

ε–`Robust` Q-Learning Algorithm

Algorithm 2 The ε-`Robust` Q-Learning Algorithm

Input: Initial estimate $Q_0 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, step-size α, corruption fraction ε, confidence level δ, total iterations T
for $t = 0,1,\dots T$ **do**
  for $(s,a) \in \mathcal{S}\times\mathcal{A}$ **do**
    Sample $s_t(s,a) \sim P(\cdot|s,a)$.
    Observe $y_t(s,a)$.
    $\tilde{r}_t(s,a) \leftarrow \texttt{TRIM}\left[\{y_k(s,a)\}_{0\le k\le t}, \varepsilon, \delta_1\right]$, where $\delta_1 = \delta/(2|\mathcal{S}||\mathcal{A}|T)$.
    if $|\tilde{r}_t(s,a)| > G_t$ **then**
      Set $\tilde{r}_t(s,a) \leftarrow \texttt{SIGN}(\tilde{r}_t(s,a)) \times G_t$, where G_t is the threshold function in Eq. (9).
    end if
    Update $Q_t(s,a)$ as per Eq. (10).
  end for
end for

Update Rule of ε–`Robust` Q-Learning Algorithm

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left(\tilde{r}_t(s,a) + \gamma \max_{a'\in\mathcal{A}} Q_t(s_t(s,a),a')\right). \tag{10}$$

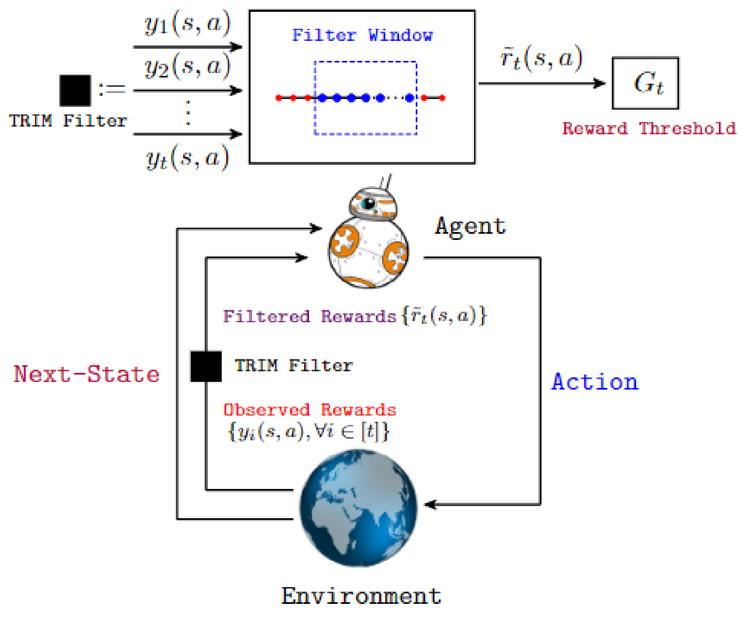

Figure 1: Schematic Diagram of ε-`Robust` Q-Learning Algorithm at time instant $t \in [T]$.

Finite Time Analysis of our proposed algorithm

Theorem 3

(Robust Q-learning bound) *Suppose the corruption fraction satisfies $\varepsilon \in [0,1/16)$. Then, given any $\delta \in (0,1)$, the output of Algorithm 2 with step-size $\alpha = \frac{\log T}{(1-\gamma)T}$ satisfies the following bound with probability at least $1-\delta$:*

$$\|Q_T - Q^*\| \le \frac{\|Q_0 - Q^*\|}{T} + O\left(\frac{\mathcal{R}}{(1-\gamma)^{\frac{5}{2}}}\frac{\log T}{\sqrt{T}}\sqrt{\log\left(\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}\right)} + \frac{\mathcal{R}\sqrt{\varepsilon}}{1-\gamma}\right).$$

Numerical Results

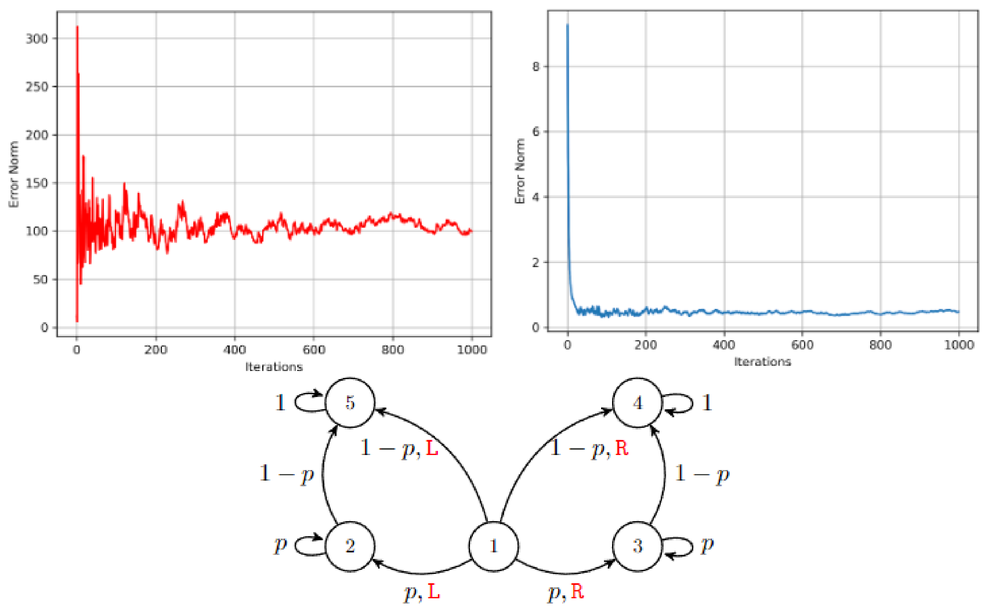

Figure 2: Consider the MDP depicted in the figure. In the absence of attacks, state 1 yields a reward of d when taking action $a = L$ and a reward of $-d$ for action $a = R$, where $d > 0$. States 2 and 3 provide a reward of 1, while states 4 and 5 provide a reward of 0. Under a Huber attack model, the rewards in state $s = 1$ are perturbed: for the state-action pair $(1,L)$, the reward is d with probability $1-\varepsilon$ and $-C$ with probability ε; for $(1,R)$, the reward is $-d$ with probability $1-\varepsilon$ and C with probability ε. The corruption signal C is defined as $((2-\varepsilon)d+\kappa)\,\varepsilon^{-1}$, where $\kappa > 0$ is a tunable parameter. The error norm $\|Q_t - Q^*\|_\infty$ is plotted for both the Vanilla Q-Learning Algorithm under corruption with $\varepsilon = 0.01$ (shown in red) and our proposed ε-Robust TD Learning Algorithm under the same corruption level (shown in blue), highlighting the significant improvement of our approach.

Conclusions

We studied model-free synchronous Q-learning under a strong-contamination reward attack model, and developed a robust algorithm that comes with near-optimal guarantees. To our knowledge, these results are the first of their kind in the context of adversarial robustness of model-free reinforcement learning algorithms.