Improved Corruption Robust Algorithms for Episodic Reinforcement Learning

Yifang Chen, Simon Shaolei Du, Kevin Jamieson
University of Washington

Corrupted Episodic RL Protocol

- Unknown underlying model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$
- At episode $t = 1, 2, \ldots, T$,
    - The learner choose an non-stationary policy $\pi = \{\pi_h\}_{h=1}^{H}$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$
    - Based on $\mathcal{M}$, the policy π induces a random trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H, s_{H+1}$ where $a_h = \pi_h(s_h), r_h \sim R(s_h, a_h), s_{h+1} \sim P(\cdot|s_h, a_h)$.

Corrupted Episodic RL Protocol

- Unknown underlying model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$
- At episode $t = 1, 2, \ldots, T$,
  - The learner choose an non-stationary policy $\pi = \{\pi_h\}_{h=1}^{H}$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$
  - Based on $\mathcal{M}$, the policy π induces a random trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H, s_{H+1}$ where $a_h = \pi_h(s_h), r_h \sim R(s_h, a_h), s_{h+1} \sim P(\cdot|s_h, a_h)$.
- Regret: $\sum_{t=1}^{T} \left(\max_\pi \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi] - \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi_t] \right)$
- Goal: $\mathrm{Reg} \leq \widetilde{\mathcal{O}} \left(\min\{\sqrt{T}, \text{SomeGapComplexity}\} \text{poly}(|\mathcal{S}||\mathcal{A}|H) \right)$

Corrupted Episodic RL Protocol

- Unknown underlying model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$
- At episode $t = 1, 2, \ldots, T$,
  - The adversary choose an unknown corrupted model $\mathcal{M}_t = (\mathcal{S}, \mathcal{A}, P_t, R_t, H, s_1)$ based on the previous history.
  - The learner choose an non-stationary policy $\pi = \{\pi_h\}_{h=1}^{H}$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$
  - Based on $\mathcal{M}$, the policy π induces a random trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H, s_{H+1}$ where $a_h = \pi_h(s_h), r_h \sim R(s_h, a_h), s_{h+1} \sim P(\cdot|s_h, a_h)$.
- Regret: $\sum_{t=1}^{T} \left(\max_\pi \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi] - \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi_t] \right)$
- Goal: $\mathrm{Reg} \leq \widetilde{\mathcal{O}} \left(\min\{\sqrt{T}, \text{SomeGapComplexity}\} \text{poly}(|\mathcal{S}||\mathcal{A}|H) \right)$

Corrupted Episodic RL Protocol

- Unknown underlying model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$
- At episode $t = 1, 2, \ldots, T$,
  - The adversary choose an unknown corrupted model $\mathcal{M}_t = (\mathcal{S}, \mathcal{A}, P_t, R_t, H, s_1)$ based on the previous history.
  - The learner choose an non-stationary policy $\pi = \{\pi_h\}_{h=1}^{H}$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$
  - Based on $\mathcal{M}_t$, the policy π induces a random trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H, s_{H+1}$ where $a_h = \pi_h(s_h), r_h \sim R_t(s_h, a_h, h), s_{h+1} \sim P_t(\cdot | s_h, a_h, h)$.
- Regret: $\sum_{t=1}^{T} \left(\max_{\pi} \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi] - \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi_t] \right)$
- Goal: $\mathrm{Reg} \leq \widetilde{\mathcal{O}} \left(\min\{\sqrt{T}, \mathrm{SomeGapComplexity}\} \mathrm{poly}(|\mathcal{S}||\mathcal{A}|H) \right)$

Corrupted Episodic RL Protocol

- Unknown underlying model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$
- At episode $t = 1, 2, \ldots, T$,
  - The adversary choose an unknown corrupted model $\mathcal{M}_t = (\mathcal{S}, \mathcal{A}, P_t, R_t, H, s_1)$ based on the previous history.
  - The learner choose an non-stationary policy $\pi = \{\pi_h\}_{h=1}^{H}$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$
  - Based on $\mathcal{M}_t$, the policy π induces a random trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H, s_{H+1}$ where $a_h = \pi_h(s_h), r_h \sim R_t(s_h, a_h, h), s_{h+1} \sim P_t(\cdot | s_h, a_h, h)$.
- Regret: $\sum_{t=1}^{T} \left(\max_\pi \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi] - \mathbb{E}[\sum_{h=1}^{H} r_h \mid \mathcal{M}, \pi_t] \right)$
- Goal: $\mathrm{Reg} \leq \widetilde{\mathcal{O}} \Big(\min\{\sqrt{T}, \text{SomeGapComplexity}\} \text{poly}(|\mathcal{S}||\mathcal{A}|H) \Big)$
  $+$ Corruption Term

Formal definitions of corruption

- Corruption on rewards at episode t:

$$c_t^r = \sum_{h=1}^{H} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |R_t(s,a,h) - R(s,a)|, \qquad C^r = \sum_{t=1}^{T} c_t^r$$

- Corruption on transition functions episode t:

$$c_t^p = \sum_{h=1}^{H} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} \|P_t(\cdot|s,a,h) - P^*(\cdot|s,a)\|_1, \quad C^p = \sum_{t=1}^{T} c_t^p$$

Formal definitions of corruption

- Corruption on rewards at episode t:

$$c_t^r = \sum_{h=1}^{H} \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |R_t(s,a,h) - R(s,a)|, \qquad C^r = \sum_{t=1}^{T} c_t^r$$

- Corruption on transition functions episode t:

$$c_t^p = \sum_{h=1}^{H} \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \|P_t(\cdot|s,a,h) - P^*(\cdot|s,a)\|_1, \qquad C^p = \sum_{t=1}^{T} c_t^p$$

- The corruptions on transition functions make this problem harder than corrupted multi-arm bandits problem, which is a special case of tabular RL.

Our results

- Existing Results:
  - Corruptions only present on rewards: $\tilde{\mathcal{O}}(\min\{\sqrt{T}, \text{GapComplexity} + \sqrt{C^r \cdot \text{GapComplexity}}\})$ [JL20][JHL21]
  - Corruption term appear multiplicatively in the regret bound: $\tilde{\mathcal{O}}(C \min\{\sqrt{T}, \text{GapComplexity}\} + C^2)$, where C is the number of corrupted episodes [LSS20]

Our results

- Existing Results:
  - Corruptions only present on rewards: $\tilde{\mathcal{O}}(\min\{\sqrt{T}, \mathrm{GapComplexity} + \sqrt{C^r \cdot \mathrm{GapComplexity}}\})$ [JL20][JHL21]
  - Corruption term appear multiplicatively in the regret bound: $\tilde{\mathcal{O}}(C \min\{\sqrt{T}, \mathrm{GapComplexity}\} + C^2)$, where C is the number of corrupted episodes [LSS20]
- **Our results**:
  - $\tilde{\mathcal{O}}\big(\min\{\sqrt{T}, \mathrm{PolicyGapComplexity}\} + (1 + C^p)(C^p + C^r)\big)$

Our results

- Existing Results:
  - Corruptions only present on rewards: $\tilde{\mathcal{O}}(\min\{\sqrt{T}, \text{GapComplexity} + \sqrt{C^r \cdot \text{GapComplexity}}\})$ [JL20][JHL21]
  - Corruption term appear multiplicatively in the regret bound: $\tilde{\mathcal{O}}(C \min\{\sqrt{T}, \text{GapComplexity}\} + C^2)$, where C is the number of corrupted episodes [LSS20]
- **Our results**:
  - $\tilde{\mathcal{O}}\big(\min\{\sqrt{T}, \text{PolicyGapComplexity}\} + (1 + C^p)(C^p + C^r)\big)$
  - Corruptions appear on both rewards and transition functions

Our results

- Existing Results:
  - Corruptions only present on rewards: $\tilde{\mathcal{O}}(\min\{\sqrt{T}, \text{GapComplexity} + \sqrt{C^r \cdot \text{GapComplexity}}\})$ [JL20][JHL21]
  - Corruption term appear multiplicatively in the regret bound: $\tilde{\mathcal{O}}(C \min\{\sqrt{T}, \text{GapComplexity}\} + C^2)$, where C is the number of corrupted episodes [LSS20]
- **Our results**:
  - $\tilde{\mathcal{O}}\big(\min\{\sqrt{T}, \text{PolicyGapComplexity}\} + (1 + C^p)(C^p + C^r)\big)$
  - Corruptions appear on both rewards and transition functions
  - Corruption term appears additively in the regret bound

Our results

- Existing Results:
  - Corruptions only present on rewards: $\tilde{\mathcal{O}}(\min\{\sqrt{T}, \text{GapComplexity} + \sqrt{C^r \cdot \text{GapComplexity}}\})$ [JL20][JHL21]
  - Corruption term appear multiplicatively in the regret bound: $\tilde{\mathcal{O}}(C \min\{\sqrt{T}, \text{GapComplexity}\} + C^2)$, where C is the number of corrupted episodes [LSS20]
- **Our results**:
  - $\tilde{\mathcal{O}}\big(\min\{\sqrt{T}, \text{PolicyGapComplexity}\} + (1 + C^p)(C^p + C^r)\big)$
  - Corruptions appear on both rewards and transition functions
  - Corruption term appears additively in the regret bound
  - Corruption term appears in a finer definition, showing a separation between the corruptions on rewards and transitions

MAB ($|\mathcal{S}| = 1$): a warm-up [GKT19]

- Divide the time horizon into $\log(T)$ epochs in a doubling manner

MAB ($|\mathcal{S}|=1$): a warm-up [GKT19]

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Pull each arm a with probability $\frac{1/(\hat{\Delta}_a^m)^2}{\sum_{a'\in\mathcal{A}} 1/(\hat{\Delta}_{a'}^m)^2} \approx \frac{1/(\hat{\Delta}_a^m)^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_a^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_a^{m+1} - \mathcal{O}(\Delta_a)| &\lessapprox \hat{\Delta}_a^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

MAB ($|\mathcal{S}| = 1$): a warm-up [GKT19]

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Pull each arm a with probability $\frac{1/(\hat{\Delta}_a^m)^2}{\sum_{a' \in \mathcal{A}} 1/(\hat{\Delta}_{a'}^m)^2} \approx \frac{1/(\hat{\Delta}_a^m)^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_a^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_a^{m+1} - \mathcal{O}(\Delta_a)| &\lessapprox \hat{\Delta}_a^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

  - Get regret in $\mathcal{I}_m$ as $\sum_{a \in \mathcal{A}} \Delta_a * \frac{1}{(\hat{\Delta}_a^m)^2}$

MAB to RL: a naive extension

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Rollout each policy π with probability $\frac{1/(\hat{\Delta}_{\pi}{}^{m})^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}{}^{m})^2} \approx \frac{1/(\hat{\Delta}_{\pi}^{m})^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_{\pi}^{m}$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_{\pi}^{m+1} - \mathcal{O}(\Delta_{\pi})| &\lessapprox \hat{\Delta}_{\pi}^{m} + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

  - Get regret in $\mathcal{I}_m$ as $\sum_{\pi \in \Pi} \Delta_{\pi} * \frac{1}{(\hat{\Delta}_{\pi}{}^{m})^2}$

MAB to RL: a naive extension

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Rollout each policy π with probability $\frac{1/(\hat{\Delta}_\pi{}^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}{}^m)^2} \approx \frac{1/(\hat{\Delta}_\pi^m)^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_\pi^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_\pi^{m+1} - \mathcal{O}(\Delta_\pi)| &\lessapprox \hat{\Delta}_\pi^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

  - Get regret in $\mathcal{I}_m$ as $\sum_{\pi \in \Pi} \Delta_\pi * \frac{1}{(\hat{\Delta}_\pi{}^m)^2}$
- Final regret will depend on $|\Pi| = |\mathcal{A}|^{|\mathcal{S}|H}$!

MAB to RL: a further extension

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Rollout each policy π inside certain representative subset Π_t with probability $\frac{1/(\hat{\Delta}_{\pi}{}^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}{}^m)^2} \approx \frac{1/(\hat{\Delta}_{\pi}^m)^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_{\pi}^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_{\pi}^{m+1} - \mathcal{O}(\Delta_{\pi})| &\lessapprox \hat{\Delta}_{\pi}^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

  - Get regret in $\mathcal{I}_m$ as $\frac{1}{|\mathcal{I}_m|} \sum_{t \in \mathcal{I}_m} \sum_{\pi \in \Pi_t} \Delta_{\pi} * \frac{1}{(\hat{\Delta}_{\pi}{}^m)^2}$

MAB to RL: a further extension

- Divide the time horizon into $\log(T)$ epochs in a doubling manner
- Inside each block $\mathcal{I}_m$
  - Rollout each policy π inside certain representative subset Π_t with probability $\frac{1/(\hat{\Delta}_{\pi}{}^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}{}^m)^2} \approx \frac{1/(\hat{\Delta}_{\pi}^m)^2}{|\mathcal{I}_m|}$
  - Estimate all $\hat{\Delta}_{\pi}^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_{\pi}^{m+1} - \mathcal{O}(\Delta_{\pi})| &\lessapprox \hat{\Delta}_{\pi}^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

  - Get regret in $\mathcal{I}_m$ as $\frac{1}{|\mathcal{I}_m|} \sum_{t \in \mathcal{I}_m} \sum_{\pi \in \Pi_t} \Delta_{\pi} * \frac{1}{(\hat{\Delta}_{\pi}{}^m)^2}$
- Final regret will depend on $\max_t |\Pi_t| = poly(|\mathcal{S}||\mathcal{A}|H)$!

MAB to RL: another Problem

- Rollout each policy π inside certain representative subset Π_t with probability $\frac{1/(\hat{\Delta}_\pi^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}^m)^2} \approx \frac{1/(\hat{\Delta}_\pi^m)^2}{|\mathcal{I}_m|}$
- Estimate all $\hat{\Delta}_\pi^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_\pi^{m+1} - \mathcal{O}(\Delta_\pi)| &\lessapprox \hat{\Delta}_\pi^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

But how to find such representative sets which result accurate estimation?

MAB to RL: another Problem

- Rollout each policy π inside certain representative subset Π_t with probability $\frac{1/(\hat{\Delta}_\pi^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}^m)^2} \approx \frac{1/(\hat{\Delta}_\pi^m)^2}{|\mathcal{I}_m|}$
- Estimate all $\hat{\Delta}_\pi^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_\pi^{m+1} - \mathcal{O}(\Delta_\pi)| &\lessapprox \hat{\Delta}_\pi^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

But how to find such representative sets which result accurate estimation?

- In unknown but non-corrupted transition setting, we can adopt some existing reward-free exploration algorithms

MAB to RL: another Problem

- Rollout each policy π inside certain representative subset Π_t with probability $\frac{1/(\hat{\Delta}_\pi^m)^2}{\sum_{\pi' \in \Pi} 1/(\hat{\Delta}_{\pi'}^m)^2} \approx \frac{1/(\hat{\Delta}_\pi^m)^2}{|\mathcal{I}_m|}$
- Estimate all $\hat{\Delta}_\pi^m$ given previous history to ensure that

$$
\begin{aligned}
|\hat{\Delta}_\pi^{m+1} - \mathcal{O}(\Delta_\pi)| &\lessapprox \hat{\Delta}_\pi^m + \text{average corruptions in epoch m} \\
&\lessapprox \sqrt{1/|\mathcal{I}_m|} + \text{average corruptions until now}
\end{aligned}
$$

But how to find such representative sets which result accurate estimation?

- In unknown but non-corrupted transition setting, we can adopt some existing reward-free exploration algorithms
- When transition functions are also corrupted, the problem becomes even harder.

Our solution

- We propose a corruption robust reward-free exploration algorithm ESTALL that will
  - either return an accurate estimation on all the policies
  - or return *Fail* only when the corruptions on transition beyond certain threshold.

Our solution

- We propose a corruption robust reward-free exploration algorithm $\textsc{EstAll}$ that will
  - either return an accurate estimation on all the policies
  - or return *Fail* only when the corruptions on transition beyond certain threshold.
- $\textsc{EstAll}$ only rollouts policies at most $\mathcal{O}\left(\log(|\Pi|)\right) = \mathcal{O}\left(poly(|\mathcal{A}||\mathcal{S}|H)\right)$ time

Our solution

- We propose a corruption robust reward-free exploration algorithm EstAll that will
  - either return an accurate estimation on all the policies
  - or return *Fail* only when the corruptions on transition beyond certain threshold.
- EstAll only rollouts policies at most $\mathcal{O}(\log(|\Pi|)) = \mathcal{O}(poly(|\mathcal{A}||\mathcal{S}|H))$ time
- We propose a meta-algorithm for RL inspired by MAB setting, and use EstAll as a sub-routine.

Thanks!