Policy Poisoning in Batch Reinforcement Learning and Control

Yuzhe Ma Xuezhou Zhang Wen Sun * Xiaojin Zhu

University of Wisconsin–Madison *Microsoft Research New York

Markov Decision Process (MDP)

A Markov Decision Process (MDP) is defined as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$:
$\mathcal{S}$ is the state space
$\mathcal{A}$ is the action space
$P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition kernel
$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function
$\gamma \in [0, 1)$ is the discounting factor.

The learning goal in MDP is to find a policy π that maximizes the cumulative discounted reward:

$$Q^\pi(s,a) = \mathbb{E}[\sum_{\tau=0}^{\infty} \gamma^\tau R(s_\tau, a_\tau) \mid s_0 = s, a_0 = a, \pi]$$

The optimal value function is characterized by the Bellman optimality equation:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s', a').$$

The optimal policy is $\pi^*(s) \in \arg\max_{a\in\mathcal{A}} Q^*(s,a)$.

Model-based Batch Reinforcement Learner

Step 1. The learner estimates an MDP $\hat{M} = (\mathcal{S}, \mathcal{A}, \hat{P}, \hat{R}, \gamma)$ from a training set D.

Maximum likelihood estimate for the transition kernel: $\hat{P} \in \arg\max_P \sum_{t=0}^{T-1} \log P(s_t'|s_t, a_t)$.

Least-squares estimate for the reward function: $\hat{R} = \arg\min_R \sum_{t=0}^{T-1} (r_t - R(s_t, a_t))^2$.

Step 2. The learner finds the optimal policy $\hat{\pi}$ that maximizes the expected discounted cumulative reward on the estimated environment $\hat{M}$, i.e.,

$$\hat{\pi} \in \arg\max_{\pi:\mathcal{S}\mapsto\mathcal{A}} \mathbb{E}_{\hat{P}} \sum_{\tau=0}^{\infty} \gamma^\tau \hat{R}(s_\tau, \pi(s_\tau)),$$

Policy Poisoning: Threat Model

Knowledge of the attacker. The attacker has access to the original training set $D^0 = (s_t, a_t, r_t^0, s_t')_{t=0:T-1}$. The attacker knows the model-based RL learner's algorithm.

Available actions of the attacker. The attacker is allowed to arbitrarily modify the rewards $\mathbf{r}^0 = (r_0^0, ..., r_{T-1}^0)$ in D^0 into $\mathbf{r} = (r_0, ..., r_{T-1})$.

Attacker's goals. The attacker has a pre-specified target policy $\pi^\dagger$. The attack goals are to (1) force the learner to learn $\pi^\dagger$, (2) minimize attack cost $\|\mathbf{r} - \mathbf{r}^0\|_\alpha$ under an α-norm chosen by the attacker.

A Unified Formulation of Policy Poisoning

We give a unified framework for policy poisoning based on bi-level optimization:

$$\begin{aligned}
\min_{\mathbf{r}, \hat{R}} \quad & \|\mathbf{r} - \mathbf{r}^0\|_\alpha \\
\text{s.t.} \quad & \hat{R} = \arg\min_R \sum_{t=0}^{T-1} (r_t - R(s_t, a_t))^2 \\
& \{\pi^\dagger\} = \arg\max_{\pi:\mathcal{S}\mapsto\mathcal{A}} \mathbb{E}_{\hat{P}} \sum_{\tau=0}^{\infty} \gamma^\tau \hat{R}(s_\tau, \pi(s_\tau)).
\end{aligned}$$

The singleton set $\{\pi^\dagger\}$ on the LHS of (1) ensures that the target policy is learned uniquely.

Policy Poisoning on Tabular Certainty Equivalence (TCE)

Step 1 of TCE: $\hat{P}(s' \mid s,a) = \frac{1}{|T_{s,a}|} \sum_{t \in T_{s,a}} \mathbb{1}[s_t' = s']$, $\quad \hat{R}(s,a) = \frac{1}{|T_{s,a}|} \sum_{t\in T_{s,a}} r_t.$

Attack Goal: $Q(s, \pi^\dagger(s)) > Q(s,a), \forall s \in \mathcal{S}, \forall a \neq \pi^\dagger(s)$.

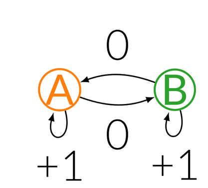
Definition. The set of ϵ-robust Q functions induced by a target policy $\pi^\dagger$ is the polytope

$$\mathcal{Q}_\epsilon(\pi^\dagger) = \{Q : Q(s, \pi^\dagger(s)) \geq Q(s,a) + \epsilon, \forall s \in \mathcal{S}, \forall a \neq \pi^\dagger(s)\}.$$

Instantiating attack on TCE:

$$\begin{aligned}
\min_{\mathbf{r}\in\mathbb{R}^T, \hat{R}, Q \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}} \quad & \|\mathbf{r} - \mathbf{r}^0\|_\alpha \\
\text{s.t.} \quad & \hat{R}(s,a) = \frac{1}{|T_{s,a}|} \sum_{t \in T_{s,a}} r_t \\
& Q(s,a) = \hat{R}(s,a) + \gamma \sum_{s'} \hat{P}(s'|s,a)\, Q\left(s', \pi^\dagger(s')\right), \forall s, \forall a \\
& Q(s, \pi^\dagger(s)) \geq Q(s,a) + \epsilon, \forall s \in \mathcal{S}, \forall a \neq \pi^\dagger(s).
\end{aligned}$$

Experimental results:



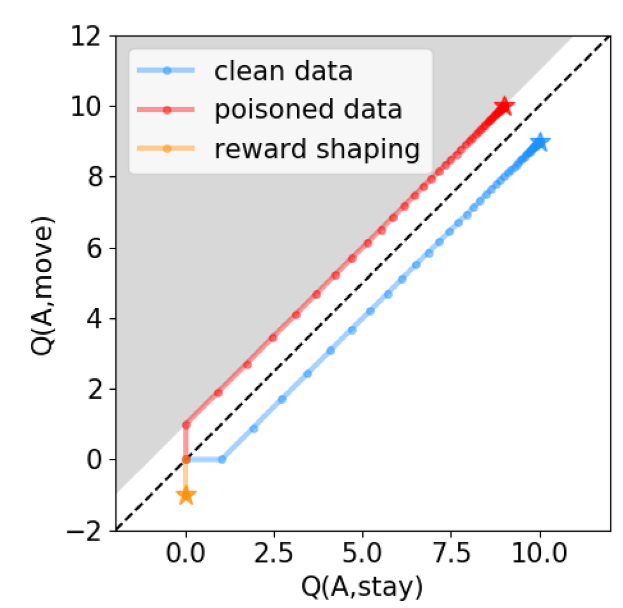
(a) A toy MDP with two states.

	stay	move
A	10	9
B	10	9

(b) Original Q values.

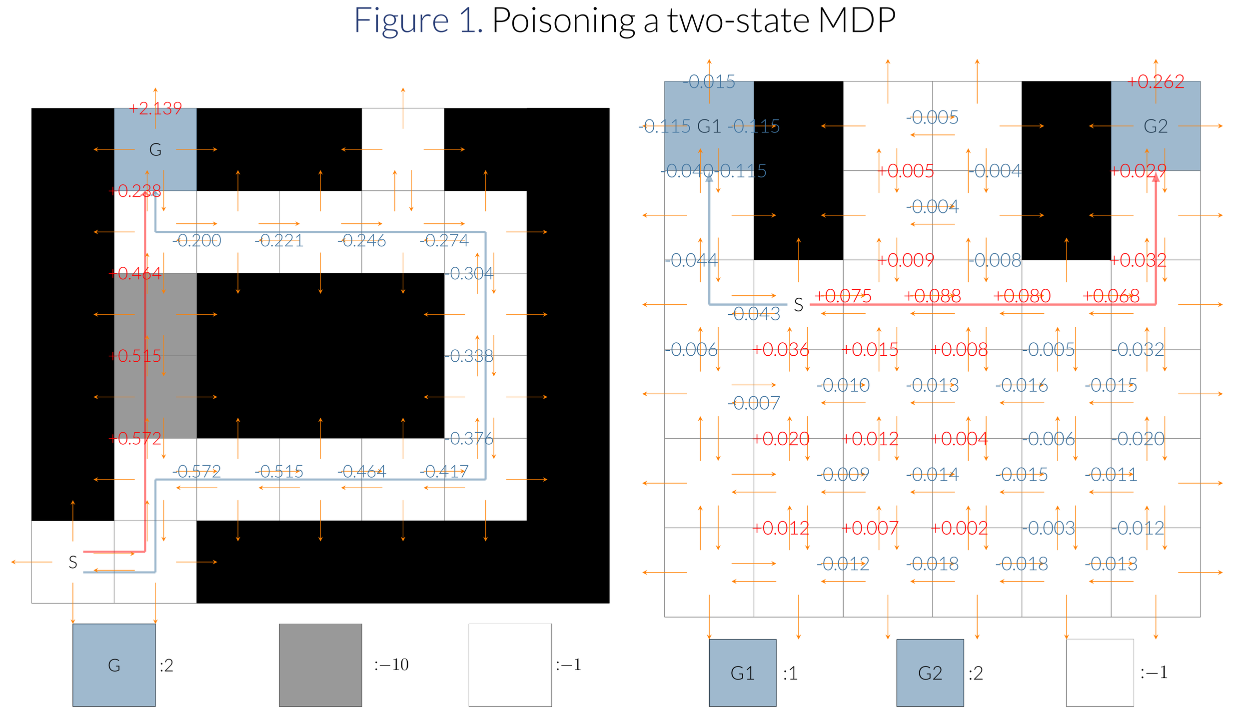
	stay	move
A	9	10
B	9	10

(c) Poisoned Q values.



(d) Trajectory for the Q values of state A during value iteration.

Figure 1. Poisoning a two-state MDP



(a) Single terminal state G. (b) Two terminal states G_1 and G_2.

Figure 2. Poisoning TCE in grid-world tasks.

Theorem. Assume $\alpha \geq 1$. Let $\mathbf{r}^*$, $\hat{R}^*$ and Q^* be an optimal solution to the attack, then

$$\frac{1}{2}(1-\gamma)\Delta(\epsilon)\left(\min_{s,a} |T_{s,a}|\right)^{\frac{1}{\alpha}} \leq \|\mathbf{r}^* - \mathbf{r}^0\|_\alpha \leq \frac{1}{2}(1+\gamma)\Delta(\epsilon) T^{\frac{1}{\alpha}}.$$

Policy Poisoning on Linear Quadratic Regulator (LQR)

The linear dynamical system is

$$s_{t+1} = As_t + Ba_t + w_t, \forall t \geq 0,$$

The cost function is $L(s,a) = \frac{1}{2} s^\top Q s + q^\top s + a^\top R a + c$.

Step 1 of LQR:

$$(\hat{A}, \hat{B}) \in \arg\min_{(A,B)} \frac{1}{2} \sum_{t=0}^{T-1} \|As_t + Ba_t - s_{t+1}\|_2^2$$

$$(\hat{Q}, \hat{R}, \hat{q}, \hat{c}) = \arg\min_{(Q\succeq 0, R\succeq \epsilon I, q, c)} \frac{1}{2} \sum_{t=0}^{T-1} \|\tfrac{1}{2} s_t^\top Q s_t + q^\top s_t + a_t^\top R a_t + c + r_t\|_2^2.$$

Optimal policy: $\hat{a}_\tau = \hat{\pi}(s_\tau) = Ks_\tau + k$, where

$$K = -\gamma\left(\hat{R} + \gamma\hat{B}^\top X \hat{B}\right)^{-1} \hat{B}^\top X \hat{A}, \quad k = -\gamma(\hat{R} + \gamma \hat{B}^\top X \hat{B})^{-1} \hat{B}^\top x.$$

$X \succeq 0$ satisfies Algebraic Riccati Equation:

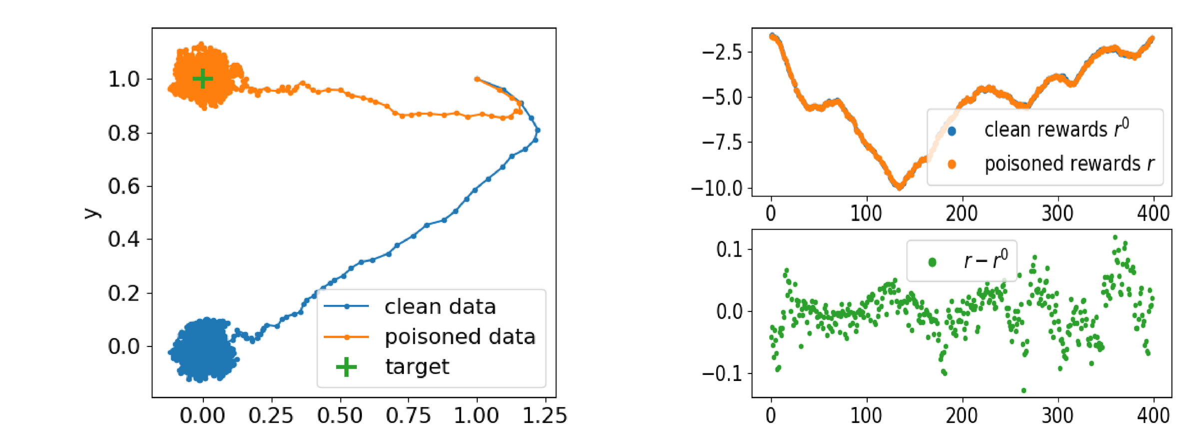
$$X = \gamma \hat{A}^\top X \hat{A} - \gamma^2 \hat{A}^\top X \hat{B}\left(\hat{R} + \gamma \hat{B}^\top X \hat{B}\right)^{-1} \hat{B}^\top X \hat{A} + \hat{Q},$$

and x satisfies $x = \hat{q} + \gamma(\hat{A} + \hat{B}K)^\top x$.

Instantiating attack on LQR:

$$\begin{aligned}
\min_{\mathbf{r}, \hat{Q}, \hat{R}, \hat{q}, \hat{c}, X\succeq 0, x} \quad & \|\mathbf{r} - \mathbf{r}^0\|_\alpha \\
\text{s.t.} \quad & -\gamma\left(\hat{R} + \gamma\hat{B}^\top X\hat{B}\right)^{-1} \hat{B}^\top X \hat{A} = K^\dagger \\
& -\gamma\left(\hat{R} + \gamma\hat{B}^\top X\hat{B}\right)^{-1} \hat{B}^\top x = k^\dagger \\
& X = \gamma \hat{A}^\top X \hat{A} - \gamma^2 \hat{A}^\top X \hat{B}\left(\hat{R} + \gamma \hat{B}^\top X \hat{B}\right)^{-1} \hat{B}^\top X \hat{A} + \hat{Q} \\
& x = \hat{q} + \gamma(\hat{A} + \hat{B}K^\dagger)^\top x \\
& (\hat{Q}, \hat{R}, \hat{q}, \hat{c}) = \arg\min_{(Q\succeq 0, R\succeq\epsilon I, q, c)} \sum_{t=0}^{T-1} \left\|\tfrac{1}{2} s_t^\top Q s_t + q^\top s_t + a_t^\top R a_t + c + r_t\right\|_2^2.
\end{aligned}$$

Experimental results:



(a) Clean and poisoned vehicle trajectory. (b) Clean and poisoned rewards.

Figure 3. Poisoning a vehicle running LQR in 4D state space.

Conclusion

We presented a policy poisoning framework against batch reinforcement learning and control.

We showed the attack problem can be formulated as convex optimization.

We provided theoretical analysis on attack feasibility and cost.

We empirically show the attack is both effective and efficient.