Vulnerability-Aware Poisoning Mechanism for Online RL with Unknown Dynamics

Prakash K. Naikade
prna00001@stud.uni-saarland.de

1 Summary

To attack the policy-based deep reinforcement learning (RL) agents, this paper proposes the Vulnerability-Aware Adversarial Critic Poison (VA2C-P) technique. Policy-based deep RL agents can use this poisoning algorithm to carry out both non-targeted and targeted attacks. No prior knowledge of the surroundings is required for VA2C-P. It also works when the attacker knows the learner's model (white-box), as well as when the learner's model is hidden (black-box). The poisoning attack is modeled as a sequential two level optimization problem (Problem Q), in which the attacker either minimizes the learner's anticipated total rewards in non-targeted poisoning or pushes the learner to learn a target policy in targeted poisoning. The problem Q is break down into two decisions and VA2C-P makes these decisions to solve Problem Q. First, $when\ to\ attack$: a new metric called stability radius is proposed to determine attack timing; Second, $how\ to\ attack$: an adversarial critic mechanism is developed to solve a relaxed version of Problem Q by just taking into account the loss of the next iteration. Experiments on a range of tasks show that the recommended attack is superior.

1.1 Overview of VA2C-P Algorithm

The paper proposed Problem (Q) to poison attacking at iteration k as a *sequential bilevel optimization* problem,

$$\begin{aligned}
&\underset{\check{\mathcal{D}}_k,\cdots,\check{\mathcal{D}}_K}{\text{argmin}} && \sum\nolimits_{j=k}^{K} \lambda_j L_A(\check{\pi}_{j+1}) && \text{((a) attacker's weighted loss)} \qquad \text{(Q)}\\
&s.t. && \check{\pi}_{j+1} = \text{argmax}_{\pi} J(\pi, \tilde{\pi}_j, \check{\mathcal{O}}_j | \check{\mathcal{D}}_j), \forall k \le j \le K && \text{((b) imitate the learner)}\\
& && \sum\nolimits_{j=1}^{K} \mathbf{1}\{\check{\mathcal{D}}_j \neq \mathcal{D}_j\} \le C && \text{((c) limited-budget)}\\
& && U(\mathcal{D}_j, \check{\mathcal{D}}_j) \le \epsilon, \forall 1 \le j \le K && \text{((d) limited-power)}
\end{aligned}$$

Attacker's Weighted Loss. $L_A(\check{\pi})$ measures the attacker's loss w.r.t. a poisoned policy π. The weights of future attacker losses, $\lambda_{k:K}$ controls how much the attacker value the poisoning results in different iterations.

The attacker estimates the learner's policy known as imitating policy to be $\tilde{\pi}_j$ at the j-th iteration and proceeds with predicting the next-policy $\check{\pi}_{j+1}$ under poisoned observation, based on the learner's update rule $\text{argmax}_{\pi} J(\pi, \tilde{\pi}_j, \check{\mathcal{O}}_j | \check{\mathcal{D}}_j)$, where $\mathcal{D} \in \{\mathcal{O}^{\boldsymbol{s}}, \mathcal{O}^{\boldsymbol{a}}, \mathcal{O}^{\boldsymbol{r}}\}$ stands for the poison aim of the poisoning, $\check{\mathcal{O}}|\check{\mathcal{D}}$ denotes that $\mathcal{O}$ is poisoned into $\check{\mathcal{O}}$ given that poison aim $\mathcal{D}$ is poisoned into $\check{\mathcal{D}}$.

An attacker is restricted by two constraints, attack budget C i.e. the total number of iterations that the attacker could poison does not exceed C and attack power ϵ i.e. the total change $U(\mathcal{D}_k, \check{\mathcal{D}}_k)$ between $\mathcal{D}_k$ and $\check{\mathcal{D}}_k$ can not be larger than ϵ in one iteration.

For the update of an RL algorithm $\pi' = f(\pi, \mathcal{O})$, with any poison aim $\mathcal{D}$, the δ-stability radius of the update is defined as the minimum poison power needed to cause δ change in policy known as

Preprint. Under review.

δ-policy-discrepancy.

$$\phi_{\delta,\mathcal{D}}(f,\pi,\mathcal{O}) = \inf_{\epsilon}\{\exists\check{\mathcal{D}} \text{ s.t. } U(\mathcal{D},\check{\mathcal{D}}) \leq \epsilon \ \text{ and } \ d^{\max}[\pi'||\check{\pi}'] > \delta, \ \text{ where } \ \check{\pi}' = f(\pi,\check{\mathcal{O}}|\check{\mathcal{D}})\}, \quad (1)$$

Policy discrepancy $d^{\max}[\pi_1||\pi_2] = \max_s d\big[\pi_1(\cdot|s)||\pi_2(\cdot|s)\big]$, where $d[\cdot||\cdot]$ could be any measure of distribution distance.

This policy discrepancy metric is used to assess each update's vulnerability and target the most vulnerable.

To solve the problem of how to attack while satisfying the power constraint authors relaxed the original Problem (Q) into Problem (P),

$$\begin{aligned} &\text{argmin}_{\check{\mathcal{D}}_k} && L_A(\check{\pi}_{k+1}) \\ &s.t. && \check{\pi}_{k+1} = \text{argmax}_{\pi} J(\pi,\tilde{\pi}_k,\check{\mathcal{O}}_k|\check{\mathcal{D}}_k) \\ & && U(\mathcal{D}_k,\check{\mathcal{D}}_k) \leq \epsilon \end{aligned} \qquad \text{(P)}$$

To estimate the $L_A(\check{\pi})$ authors introduced adversarial critic mechanism. In this attacker learn a value function $\tilde{V}_\omega$ with observations of the learner. Then the attacker can use $\tilde{V}_\omega$ to design poisoned observations, directing the learner to a decreasing-value direction, which is called *Adversarial Critic*. Then, using importance sampling, the attacker's loss becomes $\mathbb{E}_{s,a\sim\pi_k}[\frac{\check{\pi}(a|s)}{\pi_k(a|s)}\big(G(s_t,a_t) - \tilde{V}_\omega(s_t)\big)]$, where G is the discounted future reward $\sum_{i=t}^{T}\gamma^{i-t}r_t$.

2 Strengths

The proposal of a poisoning strategy against policy-based RL agents is innovative and has never been investigated before. VA2C-P has been shown to be effective in both white-box and black-box attacks, in both targeted and untargeted attacks. The authors have demonstrated that the proposed VA2C-P attack outperforms previous gradient-based FGSM attacks. The proposed poisoning framework (Problem Q) is a broad definition that encompasses a wide range of scenarios.

3 Weaknesses

The attacker knowledge assumed in this paper is not necessarily less than that assumed in previous publications like Zhang 2020. Both require access to an MDP and victim environment simulator, as well as a huge quantity of training data before developing a viable attack policy. The black-box attack studied in the paper can still access the past and current observations except the policy of the target model.

4 Possible Improvements

The authors gave a way for determining whether the implemented attack was poor. The authors may be able to reflect on how ineffective the attack was and suggest ways to improve it.

5 Possible Extensions

Exploring how a black-box attacker with no knowledge of the learner's model or reward history could poison the agent could be a possible extension.