Policy Teaching via Environment Poisoning: Training-time Adversarial Attacks against Reinforcement Learning

Amin Rakhsha [1] **Goran Radanovic** [1] **Rati Devidze** [1] **Xiaojin Zhu** [2] **Adish Singla** [1]

Abstract

We study a security threat to reinforcement learning where an attacker poisons the learning environment to force the agent into executing a target policy chosen by the attacker. As a victim, we consider RL agents whose objective is to find a policy that maximizes average reward in undiscounted infinite-horizon problem settings. The attacker can manipulate the rewards or the transition dynamics in the learning environment at training-time and is interested in doing so in a stealthy manner. We propose an optimization framework for finding an optimal stealthy attack for different measures of attack cost. We provide sufficient technical conditions under which the attack is feasible and provide lower/upper bounds on the attack cost. We instantiate our attacks in two settings: (i) an offline setting where the agent is doing planning in the poisoned environment, and (ii) an online setting where the agent is learning a policy using a regret-minimization framework with poisoned feedback. Our results show that the attacker can easily succeed in teaching any target policy to the victim under mild conditions and highlight a significant security threat to reinforcement learning agents in practice.

1. Introduction

Understanding adversarial attacks on learning algorithms is critical to finding security threats against the deployed machine learning systems and in designing novel algorithms robust to those threats. We focus on *training-time* adversarial attacks on learning algorithms, also known as data poisoning attacks (Huang et al., 2011; Biggio & Roli, 2018; Zhu, 2018). Different from *test-time* attacks where the adversary perturbs test data to change the algorithm's decisions, poisoning attacks manipulate the training data to change the algorithm's decision-making policy itself.

Most of the existing work on data poisoning attacks has focused on supervised learning algorithms (Biggio et al., 2012; Mei & Zhu, 2015; Xiao et al., 2015; Alfeld et al., 2016; Li et al., 2016; Koh et al., 2018). In contemporary works, researchers have explored data poisoning attacks against stochastic multi-armed bandits (Jun et al., 2018; Liu & Shroff, 2019) and contextual bandits (Ma et al., 2018), which belong to family of online learning algorithms with limited feedback— such algorithms are widely used in real-world applications such as news article recommendation (Li et al., 2010) and web advertisements ranking (Chapelle et al., 2014). The feedback loop in online learning makes these applications easily susceptible to data poisoning, e.g., attacks in the form of click baits (Miller et al., 2011). In this paper, we focus on data poisoning attacks against reinforcement learning (RL) algorithms, an online learning paradigm for sequential decision-making under uncertainty (Sutton & Barto, 2018).[1] Given that RL algorithms are increasingly used in critical applications, including cyber-physical systems (Li & Qiu, 2019) and personal assistive devices (Rybski et al., 2007), it is of utmost importance to investigate the security threat to RL algorithms against different forms of poisoning attacks.

1.1. Overview of our Results and Contributions

In the following, we discuss a few of the types/dimensions of poisoning attacks in RL in order to highlight the novelty of our work in comparison to existing work.

Type of adversarial manipulation. Existing works on poisoning attacks against RL have studied the manipulation of rewards only (Zhang & Parkes, 2008; Zhang et al., 2009; Ma et al., 2019; Huang & Zhu, 2019). A key novelty of our work is that we study environment poisoning, i.e., manipulating rewards or manipulating transition dynamics.

[1]Max Planck Institute for Software Systems (MPI-SWS). [2]University of Wisconsin-Madison. Correspondence to: Adish Singla <adishs@mpi-sws.org>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

[1]Poisoning attacks is also mathematically equivalent to the formulation of machine teaching with teacher being the adversary (Zhu et al., 2018). However, the problem of designing optimized environments for teaching a target policy to an RL agent is not well-understood in machine teaching.

For certain applications, it is more natural to manipulate the transition dynamics instead of the rewards, such as (i) the inventory management problem setting where state transitions are controlled by demand and supply of products in a market and (ii) conversational agents where the state is represented by the history of conversations. Through our proposed optimization framework, we show that the problem of optimally poisoning transition dynamics is a lot more challenging than that of poisoning rewards, and the attack might not be always feasible. We provide sufficient technical conditions which ensures attacker's success and provide lower/upper bounds on the attack cost.

Objective of the learning agent. Existing works have focused on studying RL agents that maximizes cumulative reward in *discounted* problem settings. In our work, we consider RL agents that maximizes average reward in *undiscounted* infinite-horizon settings (Puterman, 1994; Mahadevan, 1996)—this is a more suitable objective for many real-world applications that have cyclic tasks or tasks without absorbing states, e.g., inventory management and scheduling problems (Tadepalli et al., 1994; Puterman, 1994), and a robot learning to avoid obstacles (Mahadevan & Connell, 1992). This subtle difference in RL objective leads to technical challenges because of the absence of the contraction property that comes from the usual discount factor (Mahadevan, 1996). The results and bounds we provide are based on several measures of the problem setting including *Hajnal measure* of the transition kernel and *diameter of the Markov chain* induced by target policy in the original unpoisoned environment. Another reason we are considering average reward objective is because there are well-studied online RL algorithms for this objective using regret-minimization framework as discussed next.

Offline planning and online learning. Existing works have focused on attacks in an *offline* setting where the adversary first poisons the reward function in the environment and then the RL agent finds a policy via *planning* (Zhang & Parkes, 2008; Zhang et al., 2009; Ma et al., 2019). In contrast, we call a setting as *online* where the adversary interacts with a *learning* agent to manipulate the feedback signals. One of the key differences in these two settings is in measuring attacker's cost: The ℓ_∞*-norm* of manipulation is commonly studied for the offline setting; for the online setting, the cumulative cost of attack over time (e.g., measured by ℓ_1*-norm* of manipulations) is more relevant and has not been studied in literature. We instantiate our attacks in both the settings with appropriate notions of attack cost. For the online learning setting, we consider an agent who is learning a policy using regret-minimization framework (Auer & Ortner, 2007; Jaksch et al., 2010).

We note that our attacks are constructive, and we provide numerical simulations to support our theoretical statements. Our results demonstrate that the attacker can easily succeed in teaching (forcing) the victim to execute the desired target policy at a minimal cost.

2. Environment and RL Agent

We consider a standard RL setting, based on Markov decision processes and RL agents that optimize their expected utility. In contrast to prior work on reward poisoning attacks that assume RL agents are optimizing their total discounted rewards, we focus on the average reward optimality criterion, as specified in more detail in the following subsections.

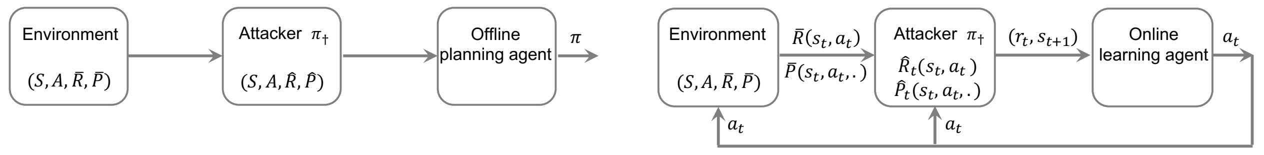
2.1. Environment, Policy, and Optimality Criteria

The environment is a Markov Decision Process (MDP) defined as $M = (S, A, R, P)$, where S is the state space, A is the action space, $R\colon S \times A \to \mathbb{R}$ is the reward function, and $P\colon S \times A \times S \to [0, 1]$ is the state transition dynamics, i.e., $P(s, a, s')$ denotes the probability of reaching state s' when taking action a in state s. We denote a deterministic policy by π, which is a map from states to actions, i.e., $\pi\colon S \to \mathcal{A}$.

In our work, we consider *average reward* optimality criteria in *undiscounted* infinite-horizon settings (Puterman, 1994; Mahadevan, 1996)—-this optimality criteria is particularly suitable for applications that have cyclic tasks or tasks without absorbing states (see discussion in Section 1.1). Given an initial state s, the expected average reward of a policy π in MDP M is given by $\rho(\pi, M, s) := \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{t=0}^{N-1} R(s_t, a_t)|s_0 = s, \pi\right]$, where the expectation is over the rewards received by the agent when starting from initial state $s_0 = s$ and following the policy π.

Under mild conditions, the above limit exists and the value $\rho(\pi, M, s)$ is independent of the initial state s. In this paper, we will assume the condition that the MDP is *ergodic*, which implies that every policy π has a stationary distribution μ^π, and $\mu^\pi(s) > 0$ for every state s (Puterman, 1994). When ergodicity holds, the average reward or *gain* of a policy can be computed as $\rho(\pi, M) = \sum_{s\in S} \mu^\pi(s)R(s, \pi(s))$; we also denote this as ρ^π when the MDP M is clear from context. For an overview of average-reward RL, we refer the reader to (Puterman, 1994; Mahadevan, 1996; Even-Dar et al., 2005).

A policy π^* is optimal if for every other deterministic policy π we have $\rho^{\pi^*} \geq \rho^\pi$, and ϵ-robust optimal if $\rho^{\pi^*} \geq \rho^\pi + \epsilon$ also holds. Finally, we define a coefficient $\alpha = \min_{s,a,s',a'} \sum_{x\in S} \min\big(P(s, a, x), P(s', a', x)\big)$. Note that $(1-\alpha)$ is equivalent to Hajnal measure of P (Puterman, 1994).



(a) Poisoning attack against an RL agent doing offline planning (b) Poisoning attack against an RL agent doing online learning

Figure 1: (a) Adversary first poisons the environment by manipulating reward function and transition dynamics, then, the RL agent finds an optimal policy via *planning* algorithms based on Dynamic Programming (Puterman, 1994). (b) Adversary interacts with an RL agent to manipulate the feedback signals; here, we consider an agent who is learning a policy using regret-minimization framework (Auer & Ortner, 2007; Jaksch et al., 2010).

2.2. RL Agent

We consider RL agents with average reward optimality criteria in the following two settings (also, see Figure 1).

Offline planning agent. In the *offline* setting, an RL agent is given an MDP M, and chooses a deterministic optimal policy $\pi^* \in \arg\max_\pi \rho(\pi, M)$. The optimal policy can be found via *planning* algorithms based on Dynamic Programming such as value iteration (Puterman, 1994).

Online learning agent. In the *online* setting, an RL agent does not know the MDP M (i.e., R and P are unknown). At each step t, the agent stochastically chooses an action a_t based on the previous observations, and then as feedback it obtains reward r_t and transitions to the next state s_{t+1}. In this paper, we consider an agent who is using a learning algorithm to take actions based on regret-minimization framework. Performance of such a learner in MDP M is measured by its *regret* which after T steps is given by $\textsc{Regret}(T, M) = \rho^* \cdot T - \sum_{t=0}^{T-1} r_t$, where $\rho^* := \rho(\pi^*, M)$ is the optimal average reward. We only consider agents with an algorithm promising a sublinear bound for total expected regret, i.e., $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$ is $o(T)$. We note that well-studied algorithms with sublinear regret exist for average reward criteria, e.g., UCRL algorithm (Auer & Ortner, 2007; Jaksch et al., 2010) and algorithms based on posterior sampling method (Agrawal & Jia, 2017).

3. Attack Models and Problem Formulation

In this section, we formulate the problem of adversarial attacks on the RL agent in both the offline and online settings. In what follows, the original MDP (before poisoning) is denoted by $\overline{M} = (S, A, \overline{R}, \overline{P})$, and an *overline* is added to the corresponding quantities before poisoning, such as $\overline{\rho}^\pi$, $\overline{\mu}^\pi$, and $\overline{\alpha}$.

The attacker has a target policy $\pi_\dagger$ and poisons the environment with the *goal* of teaching/forcing the RL agent to executing this policy.[2] The attacker is interested in doing a stealthy attack with minimal *cost* to avoid being detected.[3] We assume that the attacker knows the original MDP $\overline{M}$, i.e., the original reward function $\overline{R}$ and state transition dynamics $\overline{P}$. This assumption is standard in the existing literature on poisoning attacks against RL. The attacker requires that the RL agent behaves as specified in Section 2, however the attacker doesn't know the agent's algorithm or internal parameters.

3.1. Attack Against an Offline Planning Agent

In attacks against an offline planning agent, the attacker manipulates the original MDP $\overline{M} = (S, A, \overline{R}, \overline{P})$ to a poisoned MDP $\widehat{M} = (S, A, \widehat{R}, \widehat{P})$ which is then used by the RL agent for finding the optimal policy, see Figure 1a.

Goal of the attack. Given a margin parameter ϵ, the attacker poisons the reward function or the transition dynamics so that the target policy $\pi_\dagger$ is ϵ-robust optimal in the poisoned MDP $\widehat{M}$, i.e., the following condition holds:

$$\rho(\pi_\dagger, \widehat{M}) \geq \rho(\pi, \widehat{M}) + \epsilon, \quad \forall \pi \neq \pi^\dagger. \tag{1}$$

Cost of the attack. We consider two notions of costs defined in terms of ℓ_p-norm of differences in reward functions (i.e., $\overline{R}$ and $\widehat{R}$) and in transition dynamics (i.e., $\overline{P}$ and $\widehat{P}$). For attacks via poisoning rewards, we quantify the cost as

$$\left\|\widehat{R} - \overline{R}\right\|_p = \left(\sum_{s,a}\left(\left|\widehat{R}(s,a) - \overline{R}(s,a)\right|\right)^p\right)^{1/p},$$

where we treated the reward functions as vectors of length $|S| \cdot |A|$ with values $R(s, a)$. For attacks via poisoning transition dynamics, we quantify the cost as follows:

$$\left\|\widehat{P} - \overline{P}\right\|_p = \left(\sum_{s,a}\left(\sum_{s'}\left|\widehat{P}(s,a,s') - \overline{P}(s,a,s')\right|\right)^p\right)^{1/p}.$$

[2]Our results can be translated to attacks against the Batch RL agent studied by (Ma et al., 2019) where the attacker poisons the training data used by the agent to learn MDP parameters.

[3]Note that the attacks that we will study in subsequent sections will be poisoning either rewards only or transition dynamics only. In a general attack manipulating both the rewards and dynamics simultaneously, one needs to appropriately combine the costs (also, refer to discussion in Section 7).

Here, we have computed ℓ_p-norm of a vector of length $|S| \cdot |A|$ where the value for each (s,a) is given by the ℓ_1-*norm* of difference of two distributions $\widehat{P}(s,a,.)$ and $\overline{P}(s,a,.)$.

3.2. Attack Against an Online Learning Agent

In attacks against an online learning agent, the attacker at time t manipulates the reward function $\overline{R}(s_t, a_t)$ and transition dynamics $\overline{P}(s_t, a_t, .)$ for the current state s_t and agent's action a_t, see Figure 1b. Then, at time t, the (poisoned) reward r_t is obtained from $\widehat{R}_t(s_t, a_t)$ instead of $\overline{R}(s_t, a_t)$ and the (poisoned) next state s_{t+1} is sampled from $\widehat{P}_t(s_t, a_t, .)$ instead of $\overline{P}(s_t, a_t, .)$.

Goal of the attack. Specification of the attacker's goal in this online setting is not as straightforward as that in the offline setting, primarily because the agent might never converge to any stationary policy. In our work, at time t when the current state is s_t, we measure the mismatch of agent's action a_t w.r.t. the target policy $\pi_\dagger$ as $\mathbb{1}[a_t \neq \pi_\dagger(s_t)]$ where $\mathbb{1}[.]$ denotes the indicator function. With this, we define a notion of *average mismatch* of learner's actions in time horizon T as follows:

$$\textsc{AvgMiss}(T) = \frac{1}{T} \cdot \bigg(\sum_{t=0}^{T-1} \mathbb{1}[a_t \neq \pi_\dagger(s_t)] \bigg). \quad (2)$$

The goal of the attacker is to ensure that $\textsc{AvgMiss}(T)$ is $o(1)$, or, alternatively, the total number of time steps where there is a mismatch is $o(T)$.

Cost of the attack. We consider a notion of *average cost* of attack in time horizon T denoted as $\textsc{AvgCost}(T)$. For reward poisoning attacks, $\textsc{AvgCost}(T)$ is

$$\frac{1}{T} \cdot \bigg(\sum_{t=0}^{T-1} \Big(\big| \widehat{R}_t(s_t, a_t) - \overline{R}(s_t, a_t) \big| \Big)^p \bigg)^{1/p},$$

and for dynamics poisoning attacks, $\textsc{AvgCost}(T)$ is

$$\frac{1}{T} \cdot \bigg(\sum_{t=0}^{T-1} \Big(\sum_{s'} \big| \widehat{P}_t(s_t, a_t, s') - \overline{P}(s_t, a_t, s') \big| \Big)^p \bigg)^{1/p}.$$

Note that here the ℓ_p-norm is defined over a vector of length T with values quantifying the attack cost at each time step t. One of the key differences in measuring attacker's cost for offline and online settings is the use of appropriate norm. While the ℓ_∞-*norm* of manipulation is most suitable and commonly studied for the offline setting; for the online setting, the cumulative cost of attack over time measured by ℓ_1-*norm* is most relevant.

4. Attacks in Offline Setting

In this section, we introduce and analyze attacks against an offline planing agent that derives its policy using a poisoned MDP $\widehat{M}$. The attacker tries to minimally change the original MDP $\overline{M}$, while at the same time ensuring that the target policy is optimal in the modified MDP $\widehat{M}$.

4.1. Overview of the Approach

To formulate optimization problems explicitly and provide bounds on the quality of obtainable solutions, we first define concepts that enable us do the following: i) reduce the complexity of achieving the attacker's goal explained in Section 3.1, ii) quantify how much change is required in rewards or transitions. To find MDP $\widehat{M}$ for which the target policy is ϵ-robust optimal, one could directly utilize constraints expressed by (1). However, the number of constraints in (1) equals $(|A|^{|S|} - 1)$, i.e., it is exponential in $|S|$, making optimization problems that directly utilize them intractable. We show that it is enough to satisfy these constraints for $(|S| \cdot |A| - |S|)$ policies that we call *neighbors* of the target policy and which we define as follows:

Definition 1. *For a policy π, its neighbor policy $\pi\{s;a\}$ is defined as*

$$\pi\{s;a\}(x) = \begin{cases} \pi(x) & x \neq s \\ a & x = s \end{cases}.$$

The following lemma provides a simple verification criteria for examining whether a policy of interest is ϵ-robust optimal in a given MDP. Its proof can be found in the supplementary material.

Lemma 1. *Policy π is ϵ-robust optimal* iff *we have $\rho^\pi \geq \rho^{\pi\{s;a\}} + \epsilon$ for every state s and action $a \neq \pi(s)$.*

In other words, Lemma 1 implies that (sub)optimality of the target policy can be deduced by examining its neighbor policies. By using the definition of average rewards, we conclude that the attacker's strategy, which consists of modifying MDP $\overline{M} = (S, A, \overline{R}, \overline{P})$ to MDP $\widehat{M} = (S, A, \widehat{R}, \widehat{P})$, is successful if and only if for each state s and action $a \neq \pi_\dagger(s)$ we have:

$$\sum_{s'} \widehat{\mu}^{\pi_\dagger}(s') \cdot \widehat{R}\big(s', \pi_\dagger(s')\big) \geq \quad (3)$$
$$\sum_{s'} \widehat{\mu}^{\pi_\dagger\{s;a\}}(s') \cdot \widehat{R}\big(s', \pi_\dagger\{s;a\}(s')\big) + \epsilon.$$

To simplify the exposition of our results, we introduce the following quantity w.r.t. MDP $\overline{M}$ for all $s \in S, a \in A$:

$$\overline{\chi}^\pi_\epsilon(s,a) = \begin{cases} \left[\frac{\overline{\rho}^{\pi\{s;a\}} - \overline{\rho}^\pi + \epsilon}{\overline{\mu}^{\pi\{s;a\}}(s)} \right]^+ & \text{for } a \neq \pi(s), \\ 0 & \text{for } a = \pi(s). \end{cases} \quad (4)$$

Here, $[x]^+$ is equal to $\max\{0, x\}$. As we show in our formal results, the amount of change needed in modifying the original MDP $\overline{M}$ to MDP $\widehat{M}$ so that (3) holds is captured through $\overline{\chi}^\pi_\epsilon$.

4.2. Attacks in Offline Setting via Poisoning Rewards

The first attack we consider is an attack on an offline planning agent via poisoning rewards, where the attacker's strategy consists of choosing a new reward function which makes the target policy ϵ-robust optimal. In this attack we have $\widehat{P} = \overline{P}$, and hence $\widehat{\mu}^{\pi}$ is equal to $\overline{\mu}^{\pi}$ for any policy π, so $\widehat{\mu}^{\pi}$ can be precomputed based on MDP $\overline{M}$. This allows us to formulate an optimization problem for finding an optimal reward poisoning attack using condition (3):

$$\min_{R} \quad \left\|R - \overline{R}\right\|_p \qquad \text{(P1)}$$
$$\text{s.t.} \quad \text{condition (3) holds with } \widehat{R} = R \text{ and } \widehat{\mu} = \overline{\mu}.$$

In this optimization problem, the constraints are obtained by using condition (3) for specific instantiation of $\widehat{R}$ and $\widehat{\mu}$. In particular, we set $\widehat{R}$ to be the optimization variable R, and $\widehat{\mu}$ to be the stationary distributions of the original transition kernel $\overline{P}$ (i.e., $\overline{\mu}$).

The optimization problem (P1) is a tractable convex program with linear constraints, and we show it is always feasible. Additionally, the following results provide lower and upper bounds on the attack cost, that is, on the cost of solution $\widehat{R}$.

Theorem 1. *The optimization problem* (P1) *is always feasible, and the cost of the optimum solution is bounded by*

$$\frac{\overline{\alpha}}{2} \cdot \left\|\overline{\chi}_{\epsilon}^{\pi_\dagger}\right\|_\infty \leq \left\|\widehat{R} - \overline{R}\right\|_p \leq \left\|\overline{\chi}_{\epsilon}^{\pi_\dagger}\right\|_p .$$

The proof of the theorem can be found in the supplementary material. The lower bound on the cost of the attack is obtained by extending the proof technique of (Ma et al., 2019) to the average reward criterion, and it depends on two factors. The first factor, $\frac{\overline{\alpha}}{2}$ (see the definition in Section 2.1), is related to the Hajnal measure of the original MDP $\overline{M}$. The second factor, $\left\|\overline{\chi}_{\epsilon}^{\pi_\dagger}\right\|_\infty$, captures the initial disadvantage of the target policy relative to its neighbor policies, as can be seen from the inequality $\overline{\chi}_{\epsilon}^{\pi_\dagger}(s, a) \geq (\overline{\rho}^{\pi_\dagger\{s;a\}} - \overline{\rho}^{\pi_\dagger})$.

The upper bound is obtained by analyzing the optimum solution to a modified version of the optimization problem (P1) which enforces reward function $\widehat{R}$ to be equal to $\overline{R}$ for state-action pairs that define the target policy. We further discuss the modified optimization problem in Section 5.2.

4.3. Attacks in Offline Setting via Poisoning Dynamics

Let us now consider attacks on an offline planning agent based on poisoning transition dynamics. The attacker's strategy now consists of choosing a new transition dynamics $\widehat{P}$ that makes the target policy ϵ-robust optimal. Again, we would like to utilize condition (3), however, now we cannot precompute $\widehat{\mu}$ since we are modifying the transition dynamics. We have to explicitly account for that in our optimization problem, and we can do so by noting that stationary distribution $\widehat{\mu}^{\pi}$ satisfies:

$$\widehat{\mu}^{\pi}(s) = \sum_{s'} \widehat{P}(s', \pi(s'), s) \cdot \widehat{\mu}^{\pi}(s'). \qquad (5)$$

Therefore, by using condition (3) and $\widehat{R} = \overline{R}$, we can formulate an optimization problem for finding an optimal dynamics poisoning attack:

$$\min_{P, \mu^{\pi_\dagger}, \mu^{\pi_\dagger\{s;a\}}} \quad \left\|P - \overline{P}\right\|_p \qquad \text{(P2)}$$
$$\text{s.t.} \quad \mu^{\pi_\dagger} \text{ and } P \text{ satisfy (5)},$$
$$\forall s, a \neq \pi_\dagger(s) : \mu^{\pi_\dagger\{s;a\}} \text{ and } P \text{ satisfy (5)},$$
$$\text{condition (3) holds with } \widehat{R} = \overline{R} \text{ and } \widehat{\mu} = \mu,$$
$$\forall s, a, s' : P(s, a, s') \geq \delta \cdot \overline{P}(s, a, s').$$

Here, $\delta \in (0, 1]$ in the last set of constraints is a given parameter, specifying how much one is allowed to decrease the original values of transition probabilities. $\delta > 0$ is a regularity condition which ensures that the new MDP is ergodic.[4] In the supplementary material, we provide a more detailed discussion on δ and how to choose it. Furthermore, the constraints related to condition (3) are obtained by instantiating $\widehat{R}$ and $\widehat{\mu}$. In particular, $\widehat{R}$ is set to be the original reward function $\overline{R}$ and $\widehat{\mu}$ is set to be the optimization variables μ (i.e., $\mu^{\pi_\dagger}$ and $\mu^{\pi_\dagger\{s;a\}}$).

Since the first and the second set of constraints are quadratic equality constraints, the optimization problem is in general non-convex. Furthermore, the third set of constraints defined by condition (3) might not even be feasible, for example, when reward function is constant or almost constant (i.e., $\overline{R}(s, a) = \overline{R}(s', a')$ or more generally $|\overline{R}(s, a) - \overline{R}(s', a')| \leq \epsilon$). In Theorem 2 we provide a sufficient condition that renders the optimization problem feasible and we provide bounds on the cost of the attack. Given the nature of the constraints in (P2), we describe an approach for finding an approximate solution in Section 6.

To introduce the formal statement, let us first define relevant quantities. Let $\overline{D}^{\pi}$ denote the *diameter* of Markov chain induced by policy π in MDP $\overline{M}$, i.e., $\overline{D}^{\pi} = \max_{s,s'} \overline{T}^{\pi}(s, s')$, where $\overline{T}^{\pi}(s, s')$ is the expected time to reach s' starting from s and following policy π. Next we define *value function* of a policy and a few related quantities. Value function of policy π in MDP $\overline{M}$ is defined as

$$\overline{V}^{\pi}(s) = \lim_{N \to \infty} \mathbb{E}\left[\sum_{t=0}^{N-1} \left(\overline{R}(s_t, a_t) - \overline{\rho}_{\pi}\right) | s_0 = s, \pi\right],$$

[4] This follows because strictly positive trajectory probabilities in MDP $\overline{M}$ remain strictly positive in the new MDP $\widehat{M}$, which further implies that all states remain recurrent and aperiodic.

where the expectation is over the rewards received by agent (relative to the offset $\overline{\rho}_\pi$) when starting from initial state $s_0 = s$ and following policy π. Furthermore, let us denote by $\overline{B}^\pi(s) = \sum_{s'} \overline{P}(s, \pi(s), s') \cdot \overline{V}^\pi(s')$ the expected value of the next state given the current state s and policy π. We also define $\overline{V}^\pi_{\text{min}} = \min_s \overline{V}^\pi(s)$ to be the minimum value of $\overline{V}^\pi$, and $sp(\overline{V}^\pi) = \max_s \overline{V}^\pi(s) - \min_s \overline{V}^\pi(s)$ to be the span of $\overline{V}^\pi$. Finally, we introduce the following two quantities important for the theorem statement:

- $\overline{\beta}^\pi_\delta(s,a)$ defined as

$$\begin{aligned}\overline{\beta}^\pi_\delta(s,a) = \overline{R}(s,\pi(s)) - \overline{R}(s,a) \\ + \overline{B}^\pi(s) - \overline{V}^\pi_{\text{min}} - \delta \cdot sp(\overline{V}^\pi).\end{aligned} \tag{6}$$

- $\overline{\Lambda}(s,a)$ defined as

$$\overline{\Lambda}(s,a) = \begin{cases} \frac{\overline{\chi}^{\pi_\dagger}_0(s,a) + \epsilon\cdot(1+\overline{D}^{\pi_\dagger})}{\overline{\chi}^{\pi_\dagger}_0(s,a) + \overline{\beta}^{\pi_\dagger}_\delta(s,a)} & \text{if } \overline{\chi}^{\pi_\dagger}_\epsilon(s,a) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\overline{\chi}^{\pi_\dagger}_0$ is obtained from $\overline{\chi}^{\pi_\dagger}_\epsilon$ by setting $\epsilon = 0$.

Theorem 2. *If there exists a solution $\widehat{P}$ to the optimization problem (P2), its cost satisfies*

$$\left\|\widehat{P} - \overline{P}\right\|_p \cdot \left\|\overline{V}^{\pi_\dagger}\right\|_\infty \geq \frac{\delta \cdot \overline{\alpha}}{2} \cdot \|\overline{\chi}^{\pi_\dagger}_0\|_\infty .$$

If for every state s and action a, it holds that either $\overline{\beta}^{\pi_\dagger}_\delta(s,a) \geq \epsilon \cdot (1 + \overline{D}^{\pi_\dagger})$ OR $\overline{\chi}^{\pi_\dagger}_\epsilon(s,a) = 0$, then the optimization problem (P2) has a solution $\widehat{P}$ whose cost is upper bounded by $\left\|\widehat{P} - \overline{P}\right\|_p \leq 2 \cdot \|\overline{\Lambda}\|_p$.

The proof can be found in the supplementary material. The proof technique for the first claim is similar to the one used for proving the lower bound in Theorem 1.

The sufficient condition of Theorem 2 suggests that the initial disadvantage of the target policy $\pi_\dagger$ relative to its neighbor policy $\pi_\dagger\{s;a\}$ (captured by $\overline{\chi}^{\pi_\dagger}_\epsilon(s,a)$) can be overcome if $\overline{\beta}^{\pi_\dagger}_\delta(s,a)$ is large enough. This in turn means that either $\overline{R}(s,\pi_\dagger(s))$ is sufficiently larger than $\overline{R}(s,a)$, or the future prospect of the target policy (i.e., $\overline{B}^{\pi_\dagger}(s)$) is greater than the myopic disadvantage (i.e., $\overline{R}(s,a) - \overline{R}(s,\pi_\dagger(s))$), by a margin which depends on the value function $\overline{V}^{\pi_\dagger}$. The proof for the upper bound is constructive, and it is based on an attack that treats state $s_{\text{sink}} \in \arg\min_{s'} \overline{V}^{\pi_\dagger}(s')$ as a *sink* state. Then, we shift transitions of state-action pairs that have $\overline{\chi}^{\pi_\dagger}_\epsilon(s,a) > 0$ towards this sink state. Notice that this is a type of an attack that does not alter the transition dynamics for the the target policy. We further discuss this type of attacks in Section 5.3.

5. Attacks in Online Setting

We now turn to attacks on an agent that learns over time using the environment feedback. Unlike the planning agent from the previous section, an online learning agent derives its policy from the interaction history, i.e., tuples of the form (s_t, a_t, r_t, s_{t+1}). To attack an online learning agent, an attacker changes the environment feedback, i.e., reward r_t or state s_{t+1}.

5.1. Overview of the Approach

The underlying idea behind our approach is to utilize the fact that a learning agent has a bounded regret, and thus chooses a suboptimal action a bounded number of times. Hence, to steer a learning agent toward selecting the target policy, it suffices for the attacker to provide the feedback (i.e., reward r_t and the next state s_{t+1}) sampled from an MDP in which the target policy is ϵ-robust optimal. Such an MDP can be derived using the results from the previous sections. We first show that this approach is sound: assuming that an ergodic MDP M has $\pi_\dagger$ as its ϵ-robust optimal policy, the expected number of steps in which a learner whose experience is drawn from MDP M deviates from $\pi_\dagger$ is of order $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$.

Lemma 2. *Consider an ergodic MDP M that has $\pi_\dagger$ as its ϵ-robust optimal policy, and an online learning agent whose expected regret on an MDP M is $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$. The average mismatch of the agent is bounded by $\mathbb{E}\left[\textsc{AvgMiss}(T)\right] \leq \frac{1}{T} \cdot K(T, M)$, where*

$$K(T,M) = \frac{\mu_{\max}}{\epsilon} \cdot \left(\mathbb{E}\left[\textsc{Regret}(T,M)\right] + 2\left\|V^{\pi_\dagger}\right\|_\infty\right), \tag{8}$$

with $\mu_{\max} := \max_{s,a} \mu^{\pi_\dagger\{s;a\}}(s)$. Here, μ^π and V^π are respectively the stationary distribution and the value function of a policy π on MDP M.

The proof of the lemma can be found in the supplementary material. The lemma implies that $o(1)$ average mismatch can be achieved in expectation using the sampling based attack described above, assuming that $\widehat{M}$ is ergodic and that $\pi_\dagger$ is its ϵ-robust optimal policy. Since the solutions to the optimization problems (P1) and (P2) from Section 4 satisfy this, the lemma applies to them as well.

However, the expected average cost of such an attack could be $\Omega(1)$ (non-diminishing over time) for a learner with subliner expected regret, unless the original and the sampling MDP have equal rewards and transition probabilities for the state-action pairs of the target policy. Intuitively, if a learner follows the target policy and there exists s for which $\widehat{R}(s,\pi_\dagger(s)) \neq \overline{R}(s,\pi_\dagger(s))$ or $\widehat{P}(s,\pi_\dagger(s),.) \neq \overline{P}(s,\pi_\dagger(s),.)$, then the attacker would incur a non-zero cost whenever the learner visits s. To avoid this issue, we need to enforce constraints on the sampling MDP specifying that the attack does not alter rewards and transitions that correspond to the state-action pairs of the target policy. This brings us to the following template that we utilize for attacks on an online learner:

- Modify the optimization problems (P1) and (P2) by respectively adding constraints $\widehat{R}(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s))$ and $\widehat{P}(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s')$.
- Obtain the sampling MDP $\widehat{M}$ by solving the more constrained version of (P1) or (P2).
- Use the sampling MDP $\widehat{M}$ instead of the environment $\overline{M}$ during the learning process, i.e., obtain r_t from $\widehat{R}(s_t, a_t)$ in the case of poisoning rewards and $s_{t+1} \sim \widehat{P}(s_t, a_t, .)$ in the case of poisoning dynamics (see Figure 1b).

5.2. Attacks in Online Setting via Poisoning Rewards

For the case of attacks via poisoning rewards, the sampling MDP $\widehat{M}$ differs from the original MDP $\overline{M}$ only in its reward function $\widehat{R}$. Following the attack template from the above, we first find $\widehat{R}$ using a modified version of (P1) which ensures that $\widehat{R}(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s))$, i.e.:

$$\begin{aligned} \min_{R} \quad & \left\| R - \overline{R} \right\|_p \qquad \text{(P3)} \\ \text{s.t.} \quad & \forall s: \; R(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s)), \\ & \text{all constraints from problem (P1).} \end{aligned}$$

As we show in the supplementary material, the optimal solution to the optimization problem (P3) is $\widehat{R}(s,a) = \overline{R}(s,a) - \overline{\chi}_\epsilon^{\pi_\dagger}(s,a)$. Since $\overline{\chi}_\epsilon^{\pi_\dagger}(s, \pi_\dagger(s)) = 0$, using the definition of $\textsc{AvgCost}(T)$ we conclude that an upper bound on $\mathbb{E}\left[\textsc{AvgCost}(T)\right]$ could be computed using (i) the number of times that a non-target action is selected and (ii) the maximum value of $\overline{\chi}_\epsilon^{\pi_\dagger}(s,a)$. The quantity in (i) is specified in the result of Lemma 2, which brings us to the main result of this subsection.

Theorem 3. *Let $\widehat{R}$ be the optimal solution to (P3). Consider the attack defined by r_t obtained from $\widehat{R}(s_t, a_t)$ and $s_{t+1} \sim \overline{P}(s_t, a_t, .)$, and an online learning agent whose expected regret on an MDP M is $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$. The average mismatch of the learner is in expectation upper bounded by*

$$\mathbb{E}\left[\textsc{AvgMiss}(T)\right] \le \frac{K(T, \widehat{M})}{T},$$

where K is defined in (8). Furthermore, the average attack cost is in expectation upper bounded by

$$\mathbb{E}\left[\textsc{AvgCost}(T)\right] \le \left\| \overline{\chi}_\epsilon^{\pi_\dagger} \right\|_\infty \cdot \frac{\left(K(T, \widehat{M})\right)^{1/p}}{T}.$$

5.3. Attacks in Online Setting via Poisoning Dynamics

For the case of attacks via dynamics poisoning, the sampling MDP $\widehat{M}$ differs form the original MDP $\overline{M}$ in its transition dynamics $\widehat{P}$. Following the attack template, we can derive the optimization problem for finding $\widehat{P}$ by simply adding the constraints $P(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s')$ in the optimization problem (P2), i.e.:

$$\begin{aligned} \min_{P, \mu^{\pi_\dagger}, \mu^{\pi_\dagger\{s;a\}}} \quad & \left\| P - \overline{P} \right\|_p \qquad \text{(P4)} \\ \text{s.t.} \quad & \forall s, s': \; P(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s'), \\ & \text{all constraints from problem (P2).} \end{aligned}$$

In the supplementary material, we show that the additional constraint allows us to transform (P4) into a tractable convex program with linear constraints. In fact, the convex program has a specific structure so that its optimal solution can be obtained by solving $|S| \cdot (|A| - 1)$ simple convex problems—each of these simpler problems only involves $|S|$ variables and $|S| + 1$ linear constraints. Moreover, the convex program is feasible if the conditions of Theorem 2 are met, and its solutions satisfy $\left\| \widehat{P}(s,a,.) - \overline{P}(s,a,.) \right\|_1 \le 2 \cdot \overline{\Lambda}(s,a)$, where $\overline{\Lambda}$ is defined in (7). Since $\overline{\Lambda}(s, \pi_\dagger(s)) = 0$, this means that one can bound $\mathbb{E}\left[\textsc{AvgCost}(T)\right]$ based on the number of times that a non-target action is selected and the maximum value of $\overline{\Lambda}(s,a)$. Combining this insight with Lemma 2, we obtain the following theorem.

Theorem 4. *Assume that the sufficient condition of Theorem 2 holds, and let $\widehat{P}$ be the optimal solution to (P4). Consider the attack defined by r_t obtained from $\overline{R}(s_t, a_t)$ and $s_{t+1} \sim \widehat{P}(s_t, a_t, .)$, and an online learning agent whose expected regret on an MDP M is $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$. The average mismatch of the learner is in expectation upper bounded by*

$$\mathbb{E}\left[\textsc{AvgMiss}(T)\right] \le \frac{K(T, \widehat{M})}{T},$$

where K is defined in (8). Furthermore, the expected attack cost is upper bounded by

$$\mathbb{E}\left[\textsc{AvgCost}(T)\right] \le 2 \cdot \left\| \overline{\Lambda} \right\|_\infty \cdot \frac{\left(K(T, \widehat{M})\right)^{1/p}}{T},$$

where $\overline{\Lambda}$ is defined in (7).

A direct consequence of Theorem 3 and Theorem 4 is that for a learner with a sublinear expected regret, the average mismatch and the average cost are expected to decrease over time as $\mathcal{O}\left(\frac{\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right]}{T}\right)$ and $\mathcal{O}\left(\frac{\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right]^{1/p}}{T}\right)$. Note that while we considered ℓ_p

norms with $p \geq 1$ to define the attack cost, the above results can be generalized to include the case of $p = 0$. For $p = 0$, the expected value of the average cost is simply upper bounded by $\frac{K(T,\hat{M})}{T}$, and this follows from Lemma 2.

6. Numerical Simulations

We perform numerical simulations on an environment represented as an MDP with four states and two actions, see Figure 2 for details. Even though simple, this environment provides a very rich and an intuitive problem setting to validate the theoretical statements and understand the effectiveness of the attacks by varying different parameters. We also vary the number of states in the MDP to check the efficiency of solving different optimization problems, and report run times. We further extend our experimental evaluation in the supplementary material and report results on an additional environment.

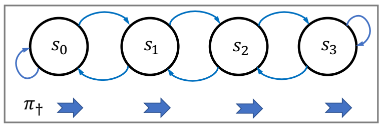


Figure 2: The environment has $|S| = 4$ states and $|A| = 2$ actions given by $\{\texttt{left}, \texttt{right}\}$. The original reward function $\overline{R}$ is action independent and has the following values: s_1 and s_2 are rewarding states with $\overline{R}(s_1, .) = \overline{R}(s_2, .) = 0.5$; state s_3 has negative reward of $\overline{R}(s_3, .) = -0.5$, and the reward of the state s_0 given by $\overline{R}(s_0, .)$ will be varied in experiments. With probability 0.9, the actions succeed in navigating the agent to left or right as shown on arrows; with probability 0.1 the agent's next state is sampled randomly from the set S. The target policy $\pi_\dagger$ is to take `right` action in all states as shown in the illustration.

6.1. Attacks in the Offline Setting: Setup and Results

For the offline setting, we considered the following attack strategies: (i) RATTACK: reward attacks using $\widehat{R}$ as solution to problem (P1) (ℓ_p*-norm* with $p = 1, 2, \infty$), (ii) NT-RATTACK: reward attacks using $\widehat{R}$ as solution to problem (P3) (solution is independent of ℓ_p*-norm*), (iii) DATTACK: dynamics attacks using $\widehat{P}$ as a solution to problem (P2) (ℓ_p*-norm* for $p = 1, 2, \infty$), and (iv) NT-DATTACK: dynamics attacks using $\widehat{P}$ as a solution to problem (P4) (solution is independent of ℓ_p*-norm*). The regularity parameter δ in the problems for solving dynamic poisoning attacks is set to 0.0001.[5]

We note that optimal solutions to the problems (P1), (P3), and (P4) can be computed efficiently using standard optimization techniques (also, refer to discussions in Section 4 and Section 5). Problem (P2) is computationally more challenging, and we provide a simple yet effective approach towards finding an approximate solution by iteratively solving the problem (P4) as follows: First, we use a simple heuristic to obtain a pool of transition kernels $\widetilde{P}$ by perturbations of $\overline{P}$ that increase the average reward of $\pi_\dagger$, and as second step, we use $\widetilde{P}$'s from this pool as as input to problem (P4) instead of $\overline{P}$. So, the runtime of solving problem (P2) depends on the number of iterations we invoke problem (P4) internally. The implementation details and code are provided in supplementary materials.

[5]Here, NT- prefix is used to highlight that *non-target only* manipulations are allowed in problems (P3) and (P4).

We vary $\overline{R}(s_0, .) \in [-5, 5]$ and vary ϵ margin $\in [0, 1]$. We use ℓ_∞*-norm* in the measure of attack cost (see Section 3.1). The results are reported as an average of 10 runs (here, DATTACK is the only stochastic algorithm because of the random initialization as discussed above).

There are two key points we want to highlight in Figure 3. First, as we increase ϵ margin, the attack problem becomes more difficult: While the reward poisoning attacks are always feasible (though with increasing attack cost), it is infeasible to do dynamics poisoning attacks for $\epsilon > 0.85$. Second, the plots also show that the solution to the generic problems (RATTACK and DATTACK for any ℓ_p*-norm* with $p = 1, 2, \infty$) can have much lower cost compared to solutions obtained by problems allowing non-target only manipulations (NT-RATTACK and NT-DATTACK).

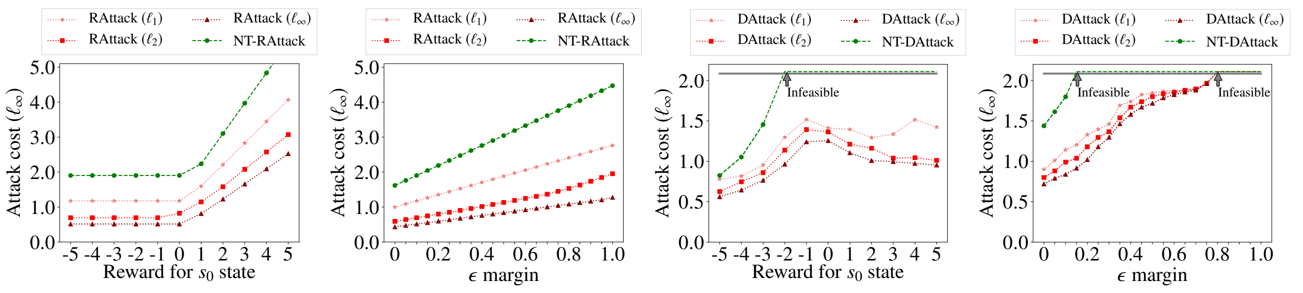
For the MDP in Figure 2 with $|S| = 4$, the run times for solving problems (P1), (P2), (P3), and (P4) are roughly 0.0110s, 3.3010s, 0.0003s, and 0.0339s, respectively. We further ran experiments on a variant of the MDP with $|S| = 100$ (keeping the same linear chain structure as in Figure 2), and this led to average run times of 0.8326s, 157.2484s, 0.6153s, and 1.1705s, respectively.

6.2. Attacks in the Online Setting: Setup and Results

For the online setting, we considered the following attack strategies: (i) reward attacks using RATTACK (with ℓ_∞, ℓ_1*-norm*) and NT-RATTACK; (ii) dynamics attacks using DATTACK (with ℓ_∞, ℓ_1*-norm*) and NT-DATTACK; and (iii) a default setting without adversary denoted as NONE where environment feedback is sampled from the original MDP $\overline{M}$.

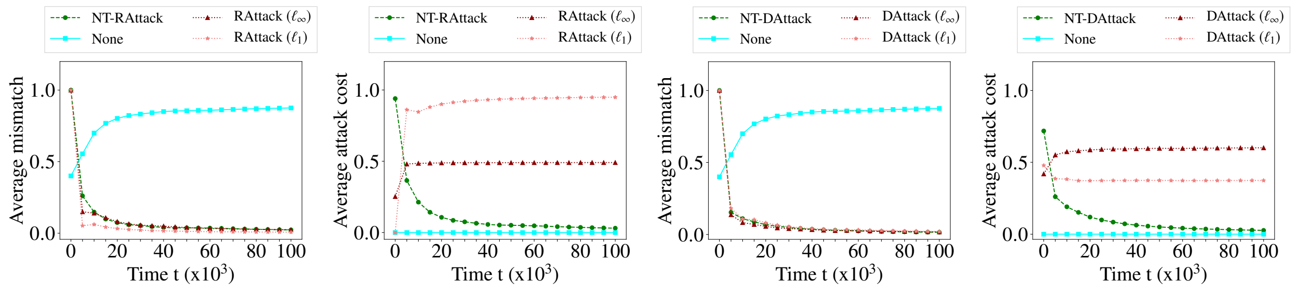
In the experiments, we fix $\overline{R}(s_0, .) = -2.5$ and $\epsilon = 0.1$; we plot the measure of the attacker's achieved goal in terms of AVGMISS and attacker's cost in terms of AVGCOST for ℓ_1*-norm* measured over time t (see Section 3.2). We consider an RL agent implementing the UCRL learning algorithm (Auer & Ortner, 2007), however the attacker does not use any knowledge of the agent's learning algorithm. The results are reported as an average of 20 runs.

The results in Figure 4 show that our proposed online at-



(a) Vary $\overline{R}(s_0,.)$: $\widehat{R}$ attack (b) Vary ϵ: $\widehat{R}$ attack (c) Vary $\overline{R}(s_0,.)$: $\widehat{P}$ attack (d) Vary ϵ: $\widehat{P}$ attack

Figure 3: Results for poisoning attacks in the offline setting from Section 4. (**a**, **b**) plots show results for attack on rewards and (**c**, **d**) plots show results for attack on dynamics. Details are in Section 6.1.



(a) AVGMISS: $\widehat{R}$ attack (b) AVGCOST: $\widehat{R}$ attack (c) AVGMISS: $\widehat{P}$ attack (d) AVGCOST: $\widehat{P}$ attack

Figure 4: Results for poisoning attacks in the online setting from Section 5. (**a**, **b**) plots show results for attack on rewards and (**c**, **d**) plots show results for attack on dynamics. Details are in Section 6.2.

tacks with NT-RATTACK and NT-DATTACK are highly effective: learner is forced to follow the target policy while the attacker's average cost is $o(1)$ (see Theorems 3, 4). In contrast, we can see that the online attacks with RATTACK and DATTACK lead to high cost for the attacker, i.e., the cumulative cost is linear w.r.t. time as anticipated in Section 5.1 (see discussions following Lemma 2).

7. Conclusion

We studied a security threat to reinforcement learning (RL) where an attacker poisons the environment, thereby forcing the agent into executing a target policy. Our work provides theoretical underpinnings of environment poisoning against RL along several new attack dimensions, including (i) adversarial manipulation of the transition dynamics, (ii) attack against RL agents maximizing average reward in undiscounted infinite horizon, and (iii) analyzing different attack costs for offline planning and online learning settings.

There are several promising directions for future work. These include expanding the attack models (e.g., attacking rewards and transitions simultaneously) and broadening the set of attack goals (e.g., under partial specification of target policy). At the same time, relaxing the assumptions on the attacker knowledge of the underlying MDP could lead to more robust attack strategies. Another interesting future direction would be to make the studied attack models more scalable, e.g., applicable to continuous and large environments. Another interesting topic would be to devise attack strategies against RL agents that use transfer learning approaches, especially in multi-agent RL systems, see (Da Silva & Costa, 2019).

While the paper provides a separate treatment for reward and transitions attack models, simultaneously attacking rewards and transitions might lead to more cost-effective solutions. One possible way to formulate the optimization problem for the simultaneous attack model is to define the cost function as a weighted sum of the cost functions used in (P1) and (P2), and combine the corresponding constraints.

While the experimental results demonstrate the effectiveness of the studied attack models, they do not reveal which types of learning algorithms are most vulnerable to the attack strategies studied in the paper. Further experimentation using a diverse set of the state of the art learning algorithms could reveal this, and provide some guidance in designing defensive strategies and novel RL algorithms robust to manipulations.

Acknowledgements

Xiaojin Zhu is supported in part by NSF 1545481, 1623605, 1704117, 1836978 and the MADLab AF Center of Excellence FA9550-18-1-0166.

References

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, 2017.

Alfeld, S., Zhu, X., and Barford, P. Data poisoning attacks against autoregressive models. In *AAAI*, 2016.

Auer, P. and Ortner, R. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 49–56, 2007.

Biggio, B. and Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *ICML*, 2012.

Chapelle, O., Manavoglu, E., and Rosales, R. Simple and scalable response prediction for display advertising. *ACM TIST*, 5(4):61:1–61:34, 2014.

Da Silva, F. L. and Costa, A. H. R. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64:645–703, 2019.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Experts in a markov decision process. In *NIPS*, pp. 401–408, 2005.

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. D. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58, 2011.

Huang, Y. and Zhu, Q. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *GameSec*, pp. 217–237, 2019.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Jun, K., Li, L., Ma, Y., and Zhu, X. J. Adversarial attacks on stochastic bandits. In *NeurIPS*, pp. 3644–3653, 2018.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *CoRR*, abs/1811.00741, 2018.

Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, pp. 1885–1893, 2016.

Li, C. and Qiu, M. *Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies*. Chapman and Hall/CRC, 2019.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pp. 661–670, 2010.

Liu, F. and Shroff, N. B. Data poisoning attacks on stochastic bandits. In *ICML*, pp. 4042–4050, 2019.

Ma, Y., Jun, K., Li, L., and Zhu, X. Data poisoning attacks in contextual bandits. In *GameSec*, pp. 186–204, 2018.

Ma, Y., Zhang, X., Sun, W., and Zhu, J. Policy poisoning in batch reinforcement learning and control. In *NeurIPS*, pp. 14543–14553, 2019.

Mahadevan, S. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22(1-3):159–195, 1996.

Mahadevan, S. and Connell, J. Automatic programming of behavior-based robots using reinforcement learning. *Artif. Intell.*, 55(2):311–365, 1992.

Mei, S. and Zhu, X. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, pp. 2871–2877, 2015.

Miller, B., Pearce, P., Grier, C., Kreibich, C., and Paxson, V. What's clicking what? techniques and innovations of today's clickbots. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 164–183. Springer, 2011.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1st edition, 1994. ISBN 0471619779.

Rybski, P. E., Yoon, K., Stolarz, J., and Veloso, M. M. Interactive robot task training through dialog and demonstration. In *Proceedings of the International Conference on Human-robot interaction*, pp. 49–56, 2007.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tadepalli, P., Ok, D., et al. H-learning: A reinforcement learning method to optimize undiscounted average reward. 1994.

Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., and Roli, F. Is feature selection secure against training data poisoning? In *ICML*, pp. 1689–1698, 2015.

Zhang, H. and Parkes, D. C. Value-based policy teaching with active indirect elicitation. In *AAAI*, 2008.

Zhang, H., Parkes, D. C., and Chen, Y. Policy teaching through reward function learning. In *EC*, 2009.

Zhu, X. An optimal control view of adversarial machine learning. *CoRR*, abs/1811.04422, 2018.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.