Stealthy Adversarial Attacks on Stochastic Multi-Armed Bandits

Zhiwei Wang[1], Huazheng Wang[2], Hongning Wang[1]
[1]Tsinghua University
[2]Oregon State University
zhiweithu@gmail.com, huazheng.wang@oregonstate.edu, wang.hongn@gmail.com

Abstract

Adversarial attacks against stochastic multi-armed bandit (MAB) algorithms have been extensively studied in the literature. In this work, we focus on reward poisoning attacks and find most existing attacks can be easily detected by our proposed detection method based on the test of homogeneity, due to their aggressive nature in reward manipulations. This motivates us to study the notion of stealthy attack against stochastic MABs and investigate the resulting attackability. Our analysis shows that against two popularly employed MAB algorithms, UCB1 and ϵ-greedy, the success of a stealthy attack depends on the environmental conditions and the realized reward of the arm pulled in the first round. We also analyze the situation for general MAB algorithms equipped with our attack detection method and find that it is possible to have a stealthy attack that almost always succeeds. This brings new insights into the security risks of MAB algorithms.

Introduction

In a stochastic multi-armed bandit (MAB) problem, a learner each time takes an arm from a presented set to interact with the environment for reward feedback, where the reward is assumed to be i.i.d. sampled from an unknown but fixed distribution (Auer, Cesa-Bianchi, and Fischer 2002; Auer 2002; Agrawal and Goyal 2017; Lattimore and Szepesvári 2020). The learner's goal is to maximize its cumulative rewards in a finite number of interactions. As such algorithms continuously learn from external feedback, their adversarial robustness has attracted increasing attention in the community (Liu and Shroff 2019; Ma et al. 2018; Jun et al. 2018). The most well-studied adversarial setting is the reward poisoning attack, where an attacker can selectively modify the reward of the learner's pulled arms to deceive the learner. Accordingly, the attacker can have two distinct goals: with a high probability, 1) force the learner to take a particular arm a linear number of times, i.e., known as a targeted attack; or 2) make the learner suffer from linear regret, i.e., known as an untargeted attack, both subject to sublinear attack cost constraint. It is known that most of the popular stochastic MABs algorithms can be easily manipulated (Jun et al. 2018; Liu and Shroff 2019), indicating a serious security vulnerability for the practical use of MAB algorithms.

The concerns have therefore spurred great research efforts in developing adversarially robust stochastic MAB algorithms. The existing efforts are mainly focused on developing new algorithms with provable robustness guarantees (Lykouris, Mirrokni, and Paes Leme 2018; Gupta, Koren, and Talwar 2019; Guan et al. 2020; Liu and Lai 2020). Various robust reward estimators or exploration strategies are introduced to tolerate the adversarial reward corruption, which, however, are at the cost of increased regret. Our study shows that most existing reward poisoning attacks can be effectively detected via the test of homogeneity (Buishand 1982). As a result, most existing MAB algorithms can be easily protected by such a detection method, with little impact on their regret. This provides a new perspective to examine the robustness or the so-called attackability (Wang, Xu, and Wang 2022) of MAB algorithms under the presence of attack detection methods.

This paper focuses on the targeted attack setting against stochastic MAB algorithms. In Section , we first introduce a method based on the test of homogeneity to detect possible reward poisoning attacks. Our key insight is that existing attack methods (Jun et al. 2018; Liu and Shroff 2019) aggressively push the realized rewards of non-target arms below that of the target arm, such that the observed reward sequence on non-target arms is no longer i.i.d. samples from the same distribution. As a result, the test of homogeneity is in a good position to actively detect such data poisoning attacks. We demonstrate that this method exhibits a low type-I error, suggesting its reliability. We then prove that with MAB algorithms like UCB1 or ϵ-greedy, if an attack method can succeed with a high probability, this attack can always be detected with a non-negligible probability. Consequently, our detection method also demonstrates a low type-II error. These results establish two key findings: 1) the proposed detection method is highly effective, and 2) existing attack methods can be easily detected using our approach.

In light of the limitations posed by existing attack methods in the presence of our proposed detection method, we introduce the concept of stealthy attack in bandit problems in Section . The results of Section also show that when the learner applies UCB1 or ϵ-greedy algorithms, no attack method can simultaneously achieve both stealthiness and efficiency, under the conditions specified by the reward gap, i.e., the bandit instance's attackability. We then propose a stealthy attack algorithm that works when the bandit instance is attackable. We should note that this limitation stems from the detection against reward poisoning attacks, rather than a flaw in the attack design. Also, it is important to note here that our work is not concerned with how the learner should react after detecting an attack, but mainly with the question of attackability of a MAB algorithm under the presence of the proposed detection mechanism, and accordingly how an attack can be carried out.

We then analyze the possibility of a stealthy attack against two most popularly used MAB algorithms, UCB1, and ϵ-greedy (Auer, Cesa-Bianchi, and Fischer 2002), equipped with our proposed attack detection method. We point out that except for the cases where we prove no stealthy attack can succeed, the stealthy attack we propose can be successful against both algorithms. Next, we analyze the feasibility of stealthy attacks against general MAB algorithms. We use construction to show that there are algorithms and corresponding attack methods where stealthy attacks can almost always succeed. This suggests that for the most general MAB algorithms, the limitations on the feasibility of the stealthy attack previously demonstrated for UCB1 and ϵ-greedy do not hold. But when we restrict the randomness of the algorithm itself, we find that we can prove a parallel result. This also opens up a new direction of research in adversarial bandit algorithms.

In addition to the theoretical analysis, we also performed an extensive set of experiments based on simulations to validate our results. We demonstrated that existing attacks against UCB1 and ϵ-greedy can be easily detected by the test of homogeneity. And the feasibility of a stealthy attack depends on the environment (i.e., the ground-truth mean of the target arm) and the realized reward of the first pulled arm, which is out of anyone's control.

To summarize, the main contributions of this work are,

- We propose a simple but effective method to detect reward poisoning attacks against stochastic MAB algorithms. We show that the detection method is highly effective and existing attack methods can be detected using our approach. This leads us to study a stealthy attack against MAB algorithms. We show that stealthy attacks can only be successful under certain circumstances; in other words, the attackability of MAB algorithms is not universal, and there is a trade-off between stealthiness and effectiveness.

- We propose a stealthy attack method that can successfully manipulate the UCB1 or ϵ-greedy algorithm protected by the detection method, except the cases proved to be not attackable in nature. Then, we construct two examples to show that for general MAB algorithms, especially when the algorithm itself is stochastic, stealthy attacks can almost always succeed. But for algorithms whose randomness in arm selection is limited, the success of stealthy attacks still depends on the environmental condition and the reward of the first pulled arm.

Related Work

Due to the wide adoption of bandit algorithms in practice, increasing amount of research attention has been spent on adversarial attacks against such algorithms to understand their robustness implications. To date, most effort has been focused on data poisoning attacks against stochastic MAB (Jun et al. 2018; Liu and Shroff 2019) and linear contextual bandit (Garcelon et al. 2020; Wang, Xu, and Wang 2022) algorithms. And these methods follow the same principle: deliberately lower the reward of non-target arms, to deceive the learner to pull the target arm a linear number of times. Our research shows that if the bandit algorithm is protected by the test of homogeneity-based attack detection, most existing attack method will fail because the range in which the reward can be lowered is limited. Other work focusing on attacks against linear contextual bandits takes into account the situation where attackers can modify historical data (Ma et al. 2018). Some recent works studied action poisoning attacks, in which the attacker manipulates the arms chosen by the learner (Liu and Lai 2020). Recently, there are also studies on reward poisoning attacks against reinforcement learning (Behzadan and Munir 2017; Huang and Zhu 2019; Ma et al. 2019; Sun, Huo, and Huang 2020; Zhang et al. 2020; Liu and Lai 2021). However, none of these existing studies consider the possible existence of attack detection, which nullifies the attack's real-world efficacy/implication.

On the defense side, there is also an increasing amount of research to improve the robustness of bandit algorithms against adversarial attacks. Lykouris, Mirrokni, and Paes Leme (2018) introduced a multi-layer active arm elimination method to improve the bandit algorithm's robustness against reward poisoning attacks. The key idea is to use increased confidence intervals to tolerate reward corruptions. But the resulting regret also degrades linearly to the amount of corruption. Gupta, Koren, and Talwar (2019) extended the solution by performing phased arm elimination with overlapping arm sets to avoid eliminating a good arm too early. This reduces the cost of regret for being robust. Guan et al. (2020) proposed to use a median-based estimator together with calibrated pure exploration for robust bandit learning. Feng, Parkes, and Xu (2020) proved that general MAB algorithms such as Thompson Sampling, UCB, and ϵ-greedy are robust when the attacker is not allowed to decrease the realized reward of each pulled arm. Liu and Lai (2020) shifted the mean reward estimation by the difference between the estimated confidence intervals between the best and worst arms to improve their bandit algorithm's robustness. Bogunovic et al. (2021) proposed two algorithms with near-optimal regret for the stochastic linear bandit problem, under known and unknown attack budget respectively. And the key insight is to expand the confidence interval for exploration. Ding, Hsieh, and Sharpnack (2022) applied the same principle to develop contextual linear bandit algorithms robust to both reward and context poisoning attacks.

The notion of attackability was first studied in (Wang, Xu, and Wang 2022) under linear stochastic bandits, where the authors suggested that the geometry spanned by the target arm, the optimal arm, and the ground-truth bandit parameter vector decides the attackability of the learning problem.

As a result, some linear stochastic bandit instances are naturally robust. Our work shares similar a spirit from this angle, but we focus on the MAB setting, which was believed to be always attackable in prior work.

Preliminaries

We study reward poisoning attacks against stochastic multi-armed bandit algorithms (Auer, Cesa-Bianchi, and Fischer 2002). Basically, a bandit game consists of N arms with unknown but fixed σ-sub-Gaussian reward distributions $\{F_1, \cdots, F_N\}$ centered at $\{\mu_1, \ldots, \mu_N\}$. The length of the time horizon is T and predetermined. At each round t, the learner pulls an arm $I_t \in [N]$ and receives reward r_t^0 from the environment following F_{I_t}. The performance of the bandit algorithm is measured by its pseudo-regret, which is defined as $R_T = \mathbb{E}[\sum_{t=1}^{T}(\mu_{i^*} - \mu_{I_t}))]$, where i^* is the best arm at hindsight, i.e., $i^* = \arg\max_{i\in[N]} \mu_i$. The learner's goal is to minimize R_T.

The attacker sits in-between the environment and the learner. At each round t after the learner chooses to play arm I_t, the attacker manipulates the reward into $r_t = r_t^0 - \alpha_t$, which is then presented to the learner. If the attacker decides not to attack, $\alpha_t = 0$. We call $\alpha_t \in \mathbb{R}$ the attack manipulation. Without loss of generality, assume arm K is the target arm, which is not the optimal arm in hindsight: $\mu_K < \max_{i=1\ldots N} \mu_i$. Define the cumulative attack cost as $C(T) = \sum_{t=1}^{T} |\alpha_t|$. The attacker's goal is to force the learner to choose the target arm a linear number of times with a sublinear attack cost. Or formally, we consider the attack successful if after T rounds the attacker spends $o(T)$ cumulative attack cost and forces the learner to choose the target arm for $T - o(T)$ times.

Detection Method

We first introduce our proposed detection method against adversarial reward poisoning attacks. The key idea is that since the attacker manipulates the rewards (i.e., lower the reward of non-target arms), the rewards observed by the learner on the same arm are no longer iid samples from the same distribution. Therefore, the learner can use the test of homogeneity (Buishand 1982) to detect possible manipulations of observed rewards so far. Once the attack is detected, the learner can resort to different means to handle the adversarial situation, e.g., restart the reward estimation. But the design of those different approaches is out of the scope of this paper.

Define the history of pulls before time t as $\mathcal{H}_t = \{(I_s, r_{I_s}, c_s)\}_{s=1}^{t}$, which represents all information at time t. Let $N_i(t)$ be the number of observations associated with arm i up to time t. Define $\hat{\mu}_i^0(t) := N_i(t)^{-1} \sum_{\{s:s\le t, I_s=i\}} r_s^0$ as the empirical pre-attack mean reward of arm i up to time t, and $\hat{\mu}_i(t) := N_i(t)^{-1} \sum_{\{s:s\le t, I_s=i\}} r_s$ as the corresponding empirical post-attack mean reward. Define function $\beta(n,\delta)$ and event set E_1 as follows,

$$\beta(n,\delta) := \sqrt{\frac{2\sigma^2}{n}\log\frac{\pi^2 N n^2}{3\,\delta}},$$
$$E_\delta := \left\{\forall i, \forall t \ge N : \left|\hat{\mu}_i^0(t) - \mu_i\right| < \beta\left(N_i(t), \delta\right)\right\},$$

where $\delta \in (0,1)$. Notice that $\beta(n,\delta)$ monotonically decreases with n. The following lemma shows that the distance between the pre-attack and ground-truth mean rewards of all arms is bounded by $\beta(N_i(t),\delta)$ with high probability.

Lemma 1. *For $\delta \in (0,1), \mathbb{P}(E_\delta) > 1-\delta$.*

Lemma 1 suggests a method for the learner to detect if the observed reward sequence has been manipulated, and the idea is simple: if the learner finds a set of empirical means for an arm that is too widely distributed, then E_δ does not hold, which rarely happens in the absence of attacks.

Detection Method. At the beginning of a bandit game, the learner chooses a parameter $\delta \in (0,1)$. The learner runs the following hypothesis test for $\forall t \in [T]$ until the null hypothesis is rejected. At each time t, the null hypothesis (i.e., no attack so far) and alternative hypothesis (i.e., there is an attack) are as follows:

H_0^t : the learner has not been attacked at and before t

H_1^t : the learner has been attacked at or before t

The learner rejects the null hypothesis H_0^t, if $\exists i \in [N]$ such that

$$\bigcap_{j\in[t], N_i(j)>0} \Big(\hat{\mu}_i(j) - \beta(N_i(j),\delta), \hat{\mu}_i(j) + \beta(N_i(j),\delta)\Big) = \emptyset. \tag{1}$$

The following lemma shows one aspect of the effectiveness of our detection method: the proposed detection method has a low type-I error.

Lemma 2. *The probability of the union of all the type-I errors introduced by Eq.*(1) *can be upper bounded by δ. In our problem setting, the type-I error refers to the detection method's erroneous claim of an attack when there is, in fact, no attack present.*

In the remainder of this paper, we shall assume that the learner adopts the detection method corresponding to a fixed parameter δ in conjunction to its chosen bandit algorithm, and we abbreviate $\beta(n,\delta)$ as $\beta(n)$. We study the setting where δ is publicly known and in practice, the attacker should treat δ as a hyper-parameter for fine tuning.

In the remaining part of Section , we show another aspect of the effectiveness of our detection method. Lemma 3-Corollary 2 demonstrates the proposed detection method exhibits a high probability of detecting effective attacks (i.e., a low type-II error). Specifically, If the detection method is applied on top of popularly employed MAB algorithms, such as UCB1 or ϵ-greedy, we can show effective reward poisoning attacks against MABs (Jun et al. 2018; Liu and Shroff 2019) can be successfully detected with a decent probability.

We first consider the case where the learner applies UCB1 (Auer, Cesa-Bianchi, and Fischer 2002), which selects an arm I_t to play at time t using the following rule:

$$I_t = \begin{cases} t, & \text{if } t \le N \\ \arg\max_i \left\{ \hat{\mu}_i(t-1) + 3\sigma\sqrt{\frac{\log t}{N_i(t-1)}} \right\}, & \text{otherwise} \end{cases}$$

For $1 \le i, j \le N$, define $\Delta^0_{ij} = \hat{\mu}_i(N) - \mu_j$, $\Delta_{ij} = \mu_i - \mu_j$. The following lemma gives the upper bound of target arm (i.e., arm K) pulls related to the realized reward of the first pulled arm $r_1 = \hat{\mu}_1(N)$, before the detection method declares the attack. This lemma points out the fundamental reason why effective attacks are always easily detectable.

Lemma 3. *Suppose the learner runs the UCB1 algorithm to choose arms. For any history of pulls $\mathcal{H}_T$ where the attack is not detected, if $\Delta^0_{1K} > \beta(1)$, with probability at least $1 - \frac{1}{T^3}$ the number of rounds that target arm K is pulled up to time T: $N_K(T)$ is bounded by*

$$\max\left\{ \frac{3C(T)}{\Delta^0_{1K} - \beta(1)}, \frac{81\sigma^2 \log T}{(\Delta^0_{1K} - \beta(1))^2}, \left(\frac{\pi^2 N}{3\delta}\right)^{\frac{2}{5}} \right\}$$

where $C(T)$ is the cumulative attack cost.

The above lemma shows that under the condition $\Delta^0_{1K} > \beta(1)$, when the detection method fails to detect the attack, with a high probability the attack itself will not succeed: either the target arm will not be pulled linear times or the attack cost cannot be sublinear.

We explain the dependency on the first pulled arm's realized reward $\widehat{\mu}_1(N)$. Due to the attack is not detected, the total amount of changes an attacker can make on an arm's average reward is limited. When the realized reward of the first pulled arm is very large, the attacker will not be able to reduce the average reward of this arm below the target arm, and thus fails to be effective (i.e., the target arm will be pulled $T - o(T)$ time). Otherwise, this attack will be detected. We will show in Section 5.1 that when $\widehat{\mu}_1(N)$ is small, there exists an attack method that will not be detected and has a high probability of success.

Based on Lemma 3, we have the following corollary regarding the efficiency of attack detection:

Corollary 1. *Suppose the learner runs the UCB1 algorithm. The attacker applies any attack methods such that it can fool the learner to pull the target arm $T - o(T)$ times at a cost of $o(T)$ with a high probability for any T large enough, i.e., with probability at least $1 - \epsilon$ and for T is larger than a constant T_0. Given any $\Delta_0 > 0$, there exists a constant T_1 related to Δ_0 such that when $T > T_1$, the attack will be detected with probability at least $1 - \frac{\epsilon}{(1-\frac{1}{T^3})} - \mathbb{P}(\widehat{\mu}_1(N) \le \mu_K + \beta(1) + \Delta_0)$.*

This corollary suggests that for any effective attack against the UCB1 algorithm, the detection will succeed with at least a certain probability related to the environment when T is large enough. In particular, this conclusion is also true for existing attacks against MABs (Jun et al. 2018; Liu and Shroff 2019).

We can explain the result as follows: when $\Delta^0_{1K} > \beta(1) + \Delta_0$, by Lemma 3 we know that the attack with failed detection will hardly be effective. Hence, for the attack method with a high probability of success, the detection only fails with a very low probability: no more than $\frac{\epsilon}{(1-\frac{1}{T^3})}$. And when $\Delta^0_{1K} \le \beta(1) + \Delta_0$, it is clear that the detection fails with a probability no more than $\mathbb{P}(\widehat{\mu}_1(N) \le \mu_K + \beta(1) + \Delta_0)$. Combining these two results gives an upper bound on the probability of detection failure.

We also note that the probability lower bound in the corollary depends on the environment conditions, and it is non-negligible when the ground-truth mean μ_1 is much larger than μ_K. This points out the effectiveness of our detection method. By using the properties of σ-sub-Gaussian distribution, it is easy to prove that when $\Delta_{1K} > \Delta_0 + \beta(1) + \sqrt{2\sigma^2 \log\frac{1}{p}}$, the probability lower bound in the Corollary 1 is larger than $1 - \frac{\epsilon}{1-\frac{1}{T^3}} - p \approx 1 - p$. We emphasize that the ground-truth mean in the environment is unbounded, therefore the difference between the ground-truth means(i.e., Δ_{1K}) can be great.

Next, let us consider the case that the learner employs the ϵ-greedy algorithm. The learner plays each arm once for $t = 1, \cdots, N$. For $t > N$

$$I_t = \begin{cases} \text{draw uniform}[N], & \text{w.p. } \epsilon_t \quad \text{(exploration)} \\ \arg\max_i \hat{\mu}_i(t-1), & \text{otherwise (exploitation)} \end{cases}$$

We can still prove the result that is parallel to the case when the learner applies UCB1, despite the randomness of ϵ-greedy.

Lemma 4. *Suppose the learner runs the ϵ-greedy algorithm with $\epsilon_t = \min\{1, \frac{CN}{t}\}$ and $C \ge 3$. Given any $\eta \in (0, 1)$, for any interaction sequence $\mathcal{H}_T$ where the attack is not detected, if $\Delta^0_{1K} > \beta(1)$, with probability at least $1 - \eta - \frac{1}{T^{3.5}} - \frac{1}{\log T}$ the number of rounds that target arm K is pulled up to time T can be bounded as follows,*

$$N_K(T) \le C\left(1 + \log\frac{T}{CN}\right) + \sqrt{3(C + C\log\frac{T}{CN})\log\frac{1}{\eta}}$$
$$+ \max\left\{ C_1 \log T, \frac{81\sigma^2 \log T}{(\Delta^0_{1K} - \beta(1))^2}, \frac{3C(T)}{\Delta^0_{1K} - \beta(1)}, C_2 \right\}$$

where $C_1 = e^{\frac{5a}{9}-1} C^2 N^2$, $C(T)$ is the total attack cost, $C_2 = CNe^{\frac{a}{6c}-\frac{1}{2}}$, $a = \beta^{-1}\left(\frac{\Delta^0_{1K} - \beta(1)}{3}\right)$.

Similar to the previous claim, the above lemma points to the main reason for the effectiveness of the detection method. It also states that the success of an attack under the previous detection method depends on the realized reward of the first pulled arm. As in the case of ϵ-greedy, we further explicitly point out the effectiveness of the detection method by the following corollary.

Corollary 2. *Suppose the learner runs ϵ-greedy algorithm with $\epsilon_t = \min\{1, \frac{CN}{t}\}$ and $C \ge 3$ to choose arm. The attacker applies any of the attack methods such that it can fool the learner to pull the target arm $T - o(T)$ times at a cost of $o(T)$ with a high probability for any T large enough, i.e., with probability at least $1 - \epsilon$ and for T is larger than a* *constant* T_0. *Given* $\Delta_0 > 0$ *and* $\eta \in (0,1)$, *there exists a constant* T_1 *related to* Δ_0 *and* η *such that when* $T > T_1$ *the attack detection will be successful with probability at least* $1 - \frac{\epsilon}{(1-\eta-T^{3.5}-\frac{1}{\log T})} - \mathbb{P}(\widehat{\mu}_1(N) \le \mu_K + \beta(1) + \Delta_0)$.

Stealthy Attacks

In this section, we propose the concept of a stealthy attack and discuss its feasibility when the learner applies different learning algorithms. Under the restriction of stealthy attacks, the magnitude of each attack is limited, and we demonstrate that the attack on the algorithm is no longer necessarily feasible, but rather depends on the environment and the realized reward.

Stealthy Attacks against UCB1 and ϵ-greedy

We first provide the definition of a stealthy attack, which basically means that an attack method will hardly be detected with our detection. We then propose a stealthy attack method against UCB1 and ϵ-greedy, and we prove its effectiveness under some conditions related to the environment and reward realization.

Stealthy Adversarial Attack. Assume that the learner's algorithm, as well as the parameter δ of the detection method, have been determined, and the learner runs the detection method as described in Section . We say that an attack algorithm is *stealthy* if for any given environment $\{F_1, \cdots, F_N\}$ we have that the detection of the attack throughout the game can succeed with probability at most δ. In other words, we require the attack to be non-detectable.

Note that we require the detection of the stealthy attack be successful with probability at most δ, which is because the probability of type-I error of our detection mechanism is at most δ. We believe that such an attack is stealthily enough.

The Limitation of Stealthy Attack. From Lemma 3, 4 we know that when $\Delta^0_{1K} > \beta(1)$ and the detection of the attack against UCB1 or ϵ-greedy failed, with high probability the attack will also failed. Combined with the definition of a stealthy attack, it is easy to demonstrate that for any stealthy attack, when $\Delta^0_{1K} > \beta(1)$ holds, then with high probability it will not succeed. This shows that under certain conditions, stealthy attacks cannot succeed in essence. The reason is that the attacker must limit the strength of attacking the non-target arms to avoid detection, which in turn leads to the attack failure under some conditions.

Stealthy attack against UCB1 and ϵ-greedy. Next we show that for the remaining cases, i.e. when $\Delta^0_{1K} < \beta(1)$, there exists a stealthy attack algorithm against both UCB1 and ϵ-greedy that can succeed with high probability.

Now we give the attack method. Suppose $\eta \in (0,1)$ is chosen by the attacker. The attacker attacks arms in the following way:

For the first N rounds(N is the number of arms) the attacker attacks in the following way: when $t < N$, for arm $i > 1$ and $i \neq K$ the attacker attacks the arm and spends minimal attack cost to make $\hat{\mu}_i(N) \le \hat{\mu}_1(N) - 2\beta(1,\eta) - 2\beta(1) - d$, $d \ge 0$ is a constant chosen by the attacker. And when the attacker attacks different learning algorithms, d can be adjusted accordingly.

After the first N rounds, the attack happens when the learner chooses to play an arm $I_t \neq K$. If $I_t \neq 1$ the attacker gives an attack $\alpha_t = \alpha_i$ where α_i is the attack cost the attacker used to attack arm i for the first time. If $I_t = 1$ the attacker computes an attack α_t with the value such that

$$\hat{\mu}_1(t) = \hat{\mu}_1(N) - \beta(1) - \beta(N_1(t-1)+1).$$

We now show that this attack method is stealthy and will succeed with high probability when $\Delta^0_{1K} < \beta(1)$.

Lemma 5. *The attack method stated above is stealthy.*

Proof. For attacks against arm 1 it's obvious that the attack detection will fail. For arm $i > 1$ and $i \neq K$, $\forall 1 \le t \le T$, $\hat{\mu}_i(t) = \hat{\mu}^0_i(t) - \alpha_i$. Then we have $\hat{\mu}_i(t) - \hat{\mu}_i(j) = \hat{\mu}^0_i(t) - \hat{\mu}^0_i(j)$, by Lemma 2 the detection to our attack will be successful with probability at most δ. Hence, the attack is stealthy. □

Theorem 1. *Suppose the learner applies the UCB1 algorithm. If* $\Delta^0_{1K} < \beta(1)$, *with probability at least* $1 - \eta - \frac{2(\Delta^0_{1K}-\beta(1))^7}{7(9\sigma^2 \log T)^{\frac{7}{2}}}$, *the attacker forces the learner to choose the target arm in at least* $T - \left(\frac{9\sigma^2}{(\Delta^0_{1K}-\beta(1))^2} + \frac{9N\sigma^2}{d^2}\right)\log T - (N-1)$ *rounds, and incurs a cumulative attack cost at most*

$$(\frac{18(\beta(1)+\beta(1,\eta))\sigma^2}{(\Delta^0_{1K}-\beta(1))^2} + \frac{9N\sigma^2(2\beta(1)+4\beta(1,\eta)+d)}{d^2})*$$

$$\log T + \left(\frac{9\sigma^2 \log T}{d^2} + 1\right) \sum_{i \neq 1,K} |\Delta_{1i}| + dN + 4\beta(1,\eta)N$$

$$+ 4\beta(1)N$$

Denote $\beta(1) - \Delta^0_{1K}$ as Δ. The number of non-target arm pulls is $O\left((\frac{\sigma^2}{\Delta^2} + \frac{N\sigma^2}{d^2})\log T + N\right)$ and the attack cost is $O\left(\left(\frac{\sigma^2}{\Delta^2} + \frac{N\sigma^2}{d^2}\right)\log T + \frac{\sigma^2 \log T}{d^2}\sum_{i\neq 1,K}|\Delta_{1i}| + dN\right)$. We can see that a larger d decreases the non-target arm pulls, and the attacker only needs to choose $d \le \sqrt{N}\Delta$, because if $d > \sqrt{N}\Delta$ we have $\frac{\sigma^2}{\Delta^2} > \frac{N\sigma^2}{d^2}$. When choosing $d = \Theta(\Delta)$, the cost is $O(\frac{\sigma^2}{\Delta^2}(N + \sum_{i\neq 1,K}|\Delta_{1i}|)\log T + \Delta N)$. The probability is $1 - \eta - O\left(\left(\frac{\Delta}{\sigma\sqrt{\log T}}\right)^7\right)$, when $T \to \infty$ it approaches $1 - \eta$.

Theorem 2. *Suppose the learner applies the same* ϵ*-greedy algorithm as in Lemma 4. If* $\Delta^0_{1K} < \beta(1)$, *with probability at least* $1 - 3\eta$, *the attacker forces the learner to choose the target arm in at least*

$$T - t_2 - N\left(C + C\log\frac{T}{CN} + \sqrt{3(C + C\log\frac{T}{CN})\log\frac{1}{\eta}}\right)$$

rounds, and using a cumulative attack cost at most

$$C(T) \le \left(C + C\log\frac{T}{CN} + \sqrt{3(C + C\log\frac{T}{CN})\log\frac{N}{\eta}}\right)$$

$$\times (4N\beta(1,\eta) + 2N\beta(1) + \sum_{i\neq 1,K}|\Delta_{1i}| + dN) + 2\beta(1)t_2 +$$

$$\beta(1,\eta))t_2$$

where $t_2 = 4(1+\frac{\sigma^2}{\Delta^2})(\frac{3\sigma^2}{4\Delta^2}\log\frac{4}{\eta}+\frac{2\sigma^2}{\Delta^2}\log\frac{4T}{\eta})+2\log\frac{4}{\eta}+t_1$, $t_1 = \max\{CNe^{\frac{5b}{6c}-\frac{3}{2C}\log\frac{\eta}{4}-\frac{1}{2}}, 5CN\}$, $b = \frac{2\sigma^2}{\Delta^2}\log(1+\frac{\Delta^2}{\sigma^2})$

We note that $b < 2 \Rightarrow t_1 = O(N), t_2 = O(\frac{\sigma^4}{\Delta^4}\log T + N)$. Notice the cost and non-target arm pulls, the attacker only needs to choose $d = 0$ and then the cost is $O((N + \sum_{i\neq 1,K}|\Delta_{1i}| + \frac{\sigma^4}{\Delta^4})\log T)$. Compared to the cost in Theorem 1, the $\log T$ term has a higher order of 4 for $\frac{\sigma}{\Delta}$ and is thus more sensitive to its change.

In addition, the reason that the feasibility of a stealthy and effective attack does not depend on the first realized rewards of the other non-target arms is that the attacker can use the realized reward of the first pulled arm to attack the other non-target arms by making their first observed reward very low. This is exactly what our algorithm does in this section.

Stealthy Attacks on General Algorithms

Now let us consider a more general situation. We study the feasibility of the stealthy attack when the learning algorithms are not limited to either UCB1 or ϵ-greedy. The following lemma shows that there exists an effective learning algorithm that corresponds to the existence of a stealthy attack algorithm that always succeeds in the attack.

Lemma 6. *We can find a learning algorithm such that given any sub-Gaussian environment* $(F_1, \cdots, F_N)$*, the number of arm pulls of any sub-optimal arm i follows* $E(N_i(T)) = \Theta(\log T)$*, and simultaneously the attacker can find a corresponding stealthy attack method s.t. for any target arm* $K \in \{1, \cdots, N\}$ *with probability at least* $1-\eta$ *the attacker can force the learner to choose the target arm for* $T - O(\log T)$ *times with a cumulative attack cost at most* $O(\log T\sqrt{\log(\frac{1}{\eta})})$.

The construction method is shown in the appendix. The design of our learning algorithm incorporates specific "action takings", or in other words, special ways of arm selection, which are then utilized by the attack method.

Lemma 5.4 demonstrates that it is possible to find attack methods that are always stealthy and effective to some bandit algorithms; but when the bandit algorithms have limited randomness, e.g., UCB1 and ϵ-greedy, from Theorem 3 and 4 we know that it might be impossible to find such an attack method. These two different results show that there are essential differences between different bandit algorithms in terms of the feasibility of a stealthy attack.

However, when we control the randomness of the distribution of rewards and the randomness of the algorithm itself, we can get results parallel to the previous cases with UCB1 and ϵ-greedy. We call an algorithm *effective under the control of the reward randomness* (ERR) (or *effective under the control of the algorithm and reward randomness* (EARR)) if and only if it has the following property: given any $\eta \in (0,1)$ and environment $(F_1, \cdots, F_N)$. When a learner applies this algorithm, there exist two functions $g(t,\eta) : \frac{g(t,\eta)}{t} \rightarrow 0$ as $t \rightarrow \infty$ and $l(n,\eta) : l(n,\eta) \geq \beta(n)$ such that if $\forall i, \forall N \leq t \leq T$ the empirical mean of a history of pulls $\mathcal{H}_T$: $\widehat{\mu}_i(N_i(T))$ was bounded in the interval $(\mu_i - l(N_i(t),\eta), \mu_i + l(N_i(t),\eta))$, then for T is large enough with probability at least $1-\eta$, the regret R_T is bounded by $g(T,\eta)$: $R_T \leq g(T,\eta)$ (or then as long as T is large enough the regret R_T is bounded by $g(T)$: $R_T \leq g(T,\eta)$ under EARR). It is not hard to see that the common algorithms like UCB1, Thompson Sampling, and ϵ-greedy are ERR, and UCB1 is an EARR algorithm.

For the EARR algorithm, We can prove the result parallel to the result in the previous subsection.

Theorem 3. *Suppose the learner runs an EARR algorithm to choose arms. Given any* $\eta \in (0,1)$ *and environment* $(F_1, \cdots, F_N)$*, we can find a function* $g(t,\eta)$*:* $\frac{g(t,\eta)}{t} \rightarrow 0$ *as* $t \rightarrow \infty$*, such that for any history of pulls* $\mathcal{H}_t$ *where the attack is not detected, if* $\Delta^0_{1K} > 2\beta(1)$*, with probability at least* $1 - \eta - \frac{1}{T^{8(\frac{1}{2}-\nu)^2}}$ *that in* $\mathcal{H}_t$ *the number of rounds that target arm K is pulled up to time T is no more than*

$$\max\{\frac{g(T,\eta)}{\nu(\Delta^0_{1K}-2\beta(1))}, \frac{16\sigma^2\ln T}{(\Delta^0_{1K}-2\beta(1))^2}, \frac{2C(T)}{\Delta^0_{1K}-2\beta(1)}\}$$

where $C(T)$ *is the cumulative attack cost.*

This theorem shows that the stealthy attack on an EARR algorithm will fail in some cases. But for a general ERR(which may not be EARR) algorithm we do not have the same result. We can prove a similar result to Lemma 6:

Lemma 7. *We can find an ERR algorithm such that the attacker can find a corresponding stealthy attack method s.t. given any environment* $(F_1, \cdots, F_N)$ *for any target arm* $K \in \{1, \cdots, N\}$ *with probability at least* $1-\eta$ *the attacker can force the learner to choose the target arm for* $T - O(\log T)$ *times with a cumulative attack cost at most* $O\left(\log T\sqrt{\log(\frac{1}{\eta})}\right)$.

The construction is similar to that in Lemma 6. Combined with Theorem 3, this lemma suggests that there is a restriction on the possibility of a successful stealthy attack for EARR, but this does not directly hold for ERR.

Experiments

We performed extensive empirical evaluations using simulation to verify our theoretical results against different MAB algorithms, attack methods, and environment configurations. We mainly present results for UCB1 here, specific results for ϵ-greedy will be provided in the appendix.

Experiment Setup

In our simulations, we execute the reward poisoning attack method proposed in (Jun et al. 2018) as our baseline and our attack algorithms against UCB1 and ϵ-greedy algorithms in the presence of attack detection proposed in Section . We varied the number of arms N in $\{10, 30\}$ and set each arm's reward distribution to an independent Gaussian distribution. The ground-truth mean reward μ_i of each arm i is sampled from $N(0,1)$. For the ϵ-greedy algorithm, we set its exploration probability $\epsilon_t = \min\{1, \frac{CN}{t}\}$. We set $C = 500 > 3$ is chosen only for the convenience of presenting the results.

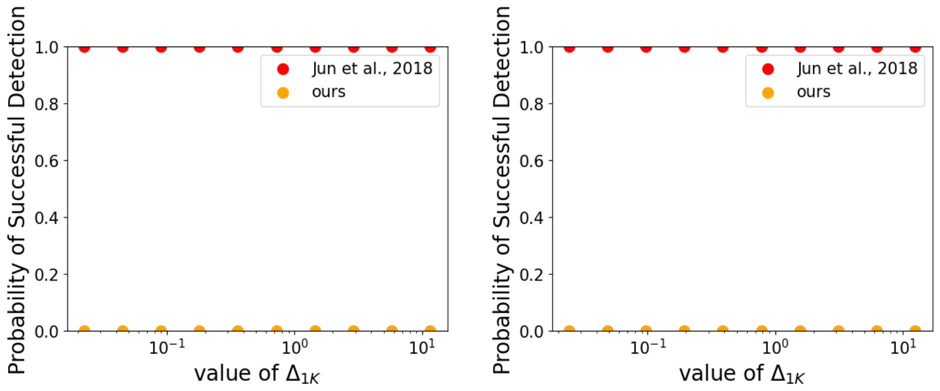

Figure 1: Probability of successful detection under (Jun et al. 2018)'s attack method when UCB1 is the victim algorithm. Left: $(N, T) = (10, 10000)$. Right: $(N, T) = (30, 20000)$

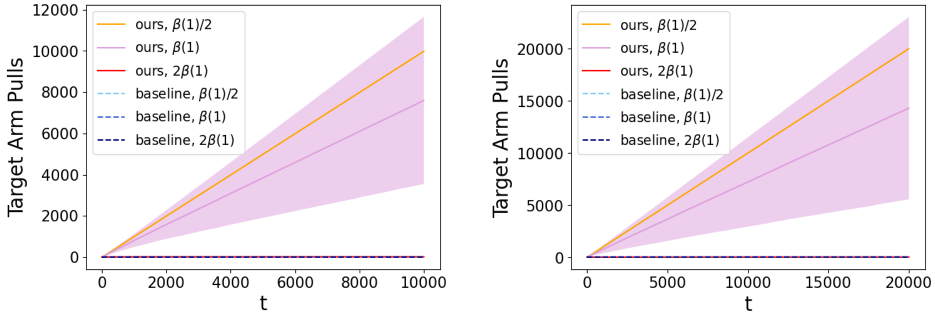

Figure 2: Target arm pulls under different attack methods when UCB1 is the victim algorithm. Left: $(N, T) = (10, 10000)$. Right: $(N, T) = (30, 20000)$

In all our experiments, we set the detection method's parameter δ to 0.05, the high probability coefficient η to 0.05, and the reward's noise scale σ in the environment to 0.1. We run each experiment for $T = 10000$ (for $N = 10$) or 20000 (for $N = 30$) iterations and repeat each experiment 20 times to report the mean and variance of performance.

Experiment Results

We first consider the effectiveness of our attack detection method when the attacker applies the commonly used attack algorithm (Jun et al. 2018) against UCB1 and ϵ-greedy algorithms. We randomly created 10 bandit instances to repeat the experiment, where we only vary the ground-truth mean reward gap Δ_{1K}. We set $\Delta_{1K} = \beta(1)/2^{4-i}$ for the i-th bandit instance. As shown in Figure 1, with a high probability this attack algorithm's reward manipulations can be detected successfully, which is predicted by Corollary 1 and 2. We find that this result is almost unaffected under different settings of N and T. That's because when N is large enough, there may be more non-target arms whose ground-truth mean is significantly larger than the target arm's ground-truth mean.

Next, we compare the performance of the baseline algorithm and our proposed algorithm under the detection. We created different environments to run the experiment, by varying Δ_{1K} in $\{\beta(1)/2, \beta(1), 2\beta(1)\}$. We stop accumulating the number of target arm pulls once the attack is detected. Note that this does not mean that the learner will necessarily stop learning after this point. From Figure 2, we can find that because the attack will be detected quickly, our baseline attack algorithm cannot trap the victim algorithm to pull the target arm linear times. For our algorithm, because it is stealthy, the victim algorithm failed to notice the reward manipulation and was executed till the end.

Conclusion

In this paper, we studied the problem of reward poisoning attacks on stochastic multi-armed bandits. We introduced a mechanism to detect such attacks, and we find that previous attack methods against UCB1 and ϵ-greedy algorithms can be easily detected. Focusing on such a detection method, we proposed a stealthy attack method that will succeed under specific conditions concerning the stochastic bandit environments and the reward of the first pulled arm.

References

Agrawal, S.; and Goyal, N. 2017. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)*, 64(5): 30.

Auer, P. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3: 397–422.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3): 235–256.

Behzadan, V.; and Munir, A. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, 262–275. Springer.

Bogunovic, I.; Losalka, A.; Krause, A.; and Scarlett, J. 2021. Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, 991–999. PMLR.

Buishand, T. A. 1982. Some methods for testing the homogeneity of rainfall records. *Journal of hydrology*, 58(1-2): 11–27.

Ding, Q.; Hsieh, C.-J.; and Sharpnack, J. 2022. Robust stochastic linear contextual bandits under adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, 7111–7123. PMLR.

Feng, Z.; Parkes, D.; and Xu, H. 2020. The intrinsic robustness of stochastic bandits to strategic manipulation. In *International Conference on Machine Learning*, 3092–3101. PMLR.

Garcelon, E.; Roziere, B.; Meunier, L.; Tarbouriech, J.; Teytaud, O.; Lazaric, A.; and Pirotta, M. 2020. Adversarial Attacks on Linear Contextual Bandits. *Advances in Neural Information Processing Systems*, 33.

Guan, Z.; Ji, K.; Bucci Jr, D. J.; Hu, T. Y.; Palombo, J.; Liston, M.; and Liang, Y. 2020. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4036–4043.

Gupta, A.; Koren, T.; and Talwar, K. 2019. Better algorithms for stochastic bandits with adversarial corruptions. In *Conference on Learning Theory*, 1562–1578. PMLR.

Huang, Y.; and Zhu, Q. 2019. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *International Conference on Decision and Game Theory for Security*, 217–237. Springer.

Jun, K.-S.; Li, L.; Ma, Y.; and Zhu, J. 2018. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, 3640–3649.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Liu, F.; and Shroff, N. 2019. Data Poisoning Attacks on Stochastic Bandits. In *International Conference on Machine Learning*, 4042–4050.

Liu, G.; and Lai, L. 2020. Action-manipulation attacks against stochastic bandits: Attacks and defense. *IEEE Transactions on Signal Processing*, 68: 5152–5165.

Liu, G.; and Lai, L. 2021. Provably efficient black-box action poisoning attacks against reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 12400–12410.

Lykouris, T.; Mirrokni, V.; and Paes Leme, R. 2018. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 114–122.

Ma, Y.; Jun, K.-S.; Li, L.; and Zhu, X. 2018. Data poisoning attacks in contextual bandits. In *International Conference on Decision and Game Theory for Security*, 186–204. Springer.

Ma, Y.; Zhang, X.; Sun, W.; and Zhu, J. 2019. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32.

Sun, Y.; Huo, D.; and Huang, F. 2020. Vulnerability-aware poisoning mechanism for online rl with unknown dynamics. *arXiv preprint arXiv:2009.00774*.

Wang, H.; Xu, H.; and Wang, H. 2022. When Are Linear Stochastic Bandits Attackable? In *International Conference on Machine Learning*, 23254–23273. PMLR.

Zhang, X.; Ma, Y.; Singla, A.; and Zhu, X. 2020. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, 11225–11234. PMLR.