# Data Poisoning Attacks on Stochastic Bandits

**Fang Liu** [1] **Ness Shroff** [1 2]

## Abstract

Stochastic multi-armed bandits form a class of online learning problems that have important applications in online recommendation systems, adaptive medical treatment, and many others. Even though potential attacks against these learning algorithms may hijack their behavior, causing catastrophic loss in real-world applications, little is known about adversarial attacks on bandit algorithms. In this paper, we propose a framework of offline attacks on bandit algorithms and study convex optimization based attacks on several popular bandit algorithms. We show that the attacker can force the bandit algorithm to pull a target arm with high probability by a slight manipulation of the rewards in the data. Then we study a form of online attacks on bandit algorithms and propose an adaptive attack strategy against any bandit algorithm *without the knowledge of the bandit algorithm*. Our adaptive attack strategy can hijack the behavior of the bandit algorithm to suffer a linear regret with only a logarithmic cost to the attacker. Our results demonstrate a significant security threat to stochastic bandits.

## 1. Introduction

Understanding adversarial attacks on machine learning systems is essential to developing effective defense mechanisms and an important step toward trustworthy artificial intelligence. A class of adversarial attacks on machine learning that have received much attention is data poisoning (Biggio et al., 2012; Mei & Zhu, 2015; Xiao et al., 2015; Alfeld et al., 2016; Li et al., 2016; Wang & Chaudhuri, 2018). Here, the attacker is able to access the leaner's training data, and has the power to manipulate a fraction of the training data in order to make the learner satisfy certain objectives. This is motivated by modern industrial scale applications of machine learning systems, where data collection and policy updates are done in a distributed way. Attacks can happen when the learner is not aware of the attacker's access to the training data.

While there has been a line of research on adversarial attacks on deep learning (Goodfellow et al., 2015; Huang et al., 2017; Lin et al., 2017) and supervised learning (Biggio et al., 2012; Mei & Zhu, 2015; Xiao et al., 2015; Alfeld et al., 2016; Li et al., 2016), little is known on adversarial attacks on stochastic multi-armed bandits, which is a form of online learning with limited feedback. This is potentially hazardous since stochastic MAB are widely used in the industry to recommend news articles (Li et al., 2010), display advertisements (Chapelle et al., 2015), allocate medical treatment (Thompson, 1933) among many others. Hence, understanding the impact of adversarial attacks on bandit algorithms is an urgent yet open problem.

Recently, there has been an important piece of offline data poisoning attacks for the contextual bandit algorithm (Ma et al., 2018). They assume that the bandit algorithm updates periodically and that the attacker can manipulate the rewards in the data before the updates in order to hijack the behavior of the bandit algorithm. Consider the news recommendation as a running example for this offline attack model. A news website has $K$ articles (i.e., arms or actions) and runs a bandit algorithm to learn a recommendation policy. Every time a user visits the website, the bandit algorithm displays an article based on historical data. Then the website receives a binary reward indicating whether the user clicks on the displayed article or not. The website keeps serving the users throughout the day and updates the bandit algorithm during the night. An attacker can perform offline data poisoning attacks before the bandit algorithm is updated. More specifically, the attacker may poison the rewards collected during the daytime and control the behavior of the bandit algorithm as it wants. The authors in (Ma et al., 2018) show that the offline attack strategy on LinUCB-type contextual bandit algorithm (Li et al., 2010) can be formulated as a convex optimization problem. However, offline attack strategies for classical bandit algorithms are still open.

In contrast to offline attacks, an online form of data poisoning attacks against bandit algorithms has also been proposed by (Jun et al., 2018). They assume that the bandit algorithm

[1]Department of Electrical and Computer Engineering, [2]Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA. Correspondence to: Fang Liu <liu.3977@osu.edu>, Ness Shroff <shroff.11@osu.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

updates step by step and the attacker sits in-between the bandit environment and the bandit algorithm. At each time step, the bandit algorithm pulls an arm and the attacker eavesdrops on the decision. Then the attacker makes an attack by manipulating the reward generated from the bandit environment. The bandit algorithm receives the poisoned reward without knowing the presence of the attacker and updates accordingly. The goal of the attacker is to control which arm appears to the bandit algorithm as the best arm at the end. Efficient attack strategies against $\epsilon$-greedy and Upper Confidence Bounds (UCB) algorithms are proposed by (Jun et al., 2018). However, online attack strategies for other bandit algorithms (e.g., Thompson Sampling (Thompson, 1933) and UCBoost (Liu et al., 2018)) are still unknown.

In this work, we have a systematic investigation of data poisoning attacks against bandit algorithms and address the aforementioned open problems. We study the data poisoning attacks in both the offline and the online cases. In the offline setting, the bandit algorithm updates periodically and the attacker can manipulate the rewards in the collected data before the update occurs. In the online setting, the attacker eavesdrops on the bandit algorithm and manipulates the feedback reward. The goal of the attacker is that the bandit algorithm considers the target arm as the optimal arm at the end. Specifically, we make the following contributions to data poisoning attacks on stochastic bandits.

1. We propose an optimization based framework for offline attacks on bandit algorithms. Then, we instantiate three offline attack strategies against $\epsilon$-greedy, UCB algorithm and Thompson Sampling, which are the solutions of the corresponding convex optimization problems. That is, there exist efficient attack strategies for the attacker against these popular bandit algorithms.
2. We study the online attacks on bandit algorithms and propose an adaptive attack strategy that can hijack any bandit algorithm *without knowing the bandit algorithm*. As far as we know, this is the first negative result showing that there is no robust and good stochastic bandit algorithm that can survive online poisoning attacks.
3. We evaluate our attack strategies by numerical results. Our attack strategies are efficient in forcing the bandit algorithms to pull a target arm at a relatively small cost. Our results expose a significant security threat as bandit algorithms are widely employed in the real world applications.

More recently, there is a line of works by (Lykouris et al., 2018; Gupta et al., 2019) studying an adversarial corruption model. While they assume that the attacker has to attack all the arms before the learner chooses an arm, we consider the case where the attacker's strategy is aware of and adaptive to the decision of the learner. The difference leads to opposite results. While they propose a robust bandit algorithm, our work shows that the existing algorithms are vulnerable to the adversarial attacks.

## 2. Problem Formulation

We consider the classical stochastic bandit setting. Suppose that there is a set $\mathcal{A} = \{1, 2, \ldots, K\}$ of $K$ arms and the bandit algorithm proceeds in discrete time $t = 1, 2, \ldots, T$. At each round $t$, the algorithm pulls an arm $a_t \in \mathcal{A}$ and the bandit environment generates a reward $r_t \in \mathcal{R}$ such that

$$r_t = \mu_{a_t} + \eta_t, \tag{1}$$

where $\eta_t$ is a zero-mean, $\sigma$-subGaussian noise and $\mu_{a_t}$ is the instantaneous reward at time $t$. In other words, the expected reward of pulling arm $a$ is $\mu_a$. Note that $\{\mu_a\}_{a\in\mathcal{A}}$ are *unknown to the bandit algorithm and the attacker*.

The performance of the bandit algorithm is measured by the regret, which is the expected difference between the total rewards obtained by an oracle that always pulls the optimal arm (i.e., the arm with the largest expected reward $\max_{a\in\mathcal{A}} \mu_a$) and the accumulative rewards collected by the bandit algorithm up to time horizon $T$. Formally, the regret $R(T)$ is given by

$$R(T) = \mathbb{E}\left[\max_{a\in\mathcal{A}} \mu_a T - \sum_{t=1}^{T} r_t\right]. \tag{2}$$

In this work, we consider the uniformly good bandit algorithm that incurs sub-linear regret, i.e., $R(T) = o(T)$.

For each reward $r_t$ returned from the bandit environment, the attacker manipulates the reward into

$$r'_t = r_t + \epsilon_t. \tag{3}$$

Then the accumulated attack cost of the attacker, $C(T)$, is measured by the norm of the vector $(\epsilon_1, \ldots, \epsilon_T)^T$. For simplicity, we consider the $l^2$-norm in the offline setting and the $l^1$-norm in the online setting. Note that the results in this work can be easily extended to any norm. For example, consider the $l^p$-norm, the total attack cost of the attacker is

$$C(T) = \left(\sum_{t=1}^{T} |\epsilon_t|^p\right)^{1/p}. \tag{4}$$

Without loss of generality, we assume that arm $a^*$ is a sub-optimal attack target, such that $\mu_{a^*} < \max_{a\in\mathcal{A}} \mu_a$. The attacker's goal is to manipulate the bandit algorithm into pulling arm $a^*$ frequently. To avoid being detected, the attacker also wants to keep the cost as small as possible.

### 2.1. Offline attack system model

The offline attack system model is illustrated in Figure 1. Besides the bandit algorithm, the bandit environment and

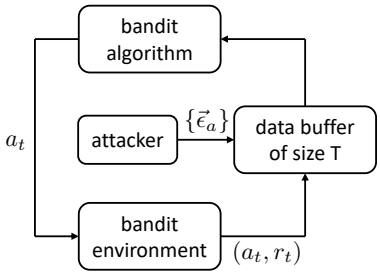



Figure 1: Offline attack system model

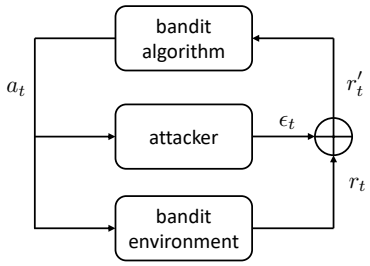



Figure 2: Online attack system model

the attacker, there is a data buffer of size $T$. This models the setting where updates of the bandit algorithm happen in mini-batches of size $T$. For round $t = 1, \ldots, T$, the bandit algorithm sequentially pulls arm $a_t$. Then the environment generates the reward $r_t$ according to Equation (1) and send the tuple $(a_t, r_t)$ to the data buffer. The data buffer stores the data stream until round $T$. At the end of round $T$, the attacker accesses the data buffer and poisons the data by Equation (3). Finally, the data buffer sends the poisoned data stream $\{(a_t, r'_t)\}_{t \le T}$ to the bandit algorithm and the bandit algorithm updates according to the received data without knowing the existence of the attacker.

The goal of the attacker in the offline setting is to force the bandit algorithm to pull the target arm $a^*$ *with high probability* at round $T+1$ (i.e., after updating with poisoned data) while incurring only a small cost. This means that the attacker wants to make the poisoning effort $\epsilon_t$ as small as possible to keep stealthy.

### 2.2. Online attack system model

The online attack system model is illustrated in Figure 2. In the online setting, the bandit algorithm updates instantly for each time step. The attacker stealthily monitors the decision of the bandit algorithm $a_t$ at each time $t$ and poisons the reward signal returned from the bandit environment by equation (3). Then the bandit algorithm receives the manipulated reward signal $r'_t$ and updates unaware of the attacker.

The goal of the attacker in the online setting is to hijack the behavior of the bandit algorithm *with high probability* by manipulating the reward signal so that the bandit algorithm pulls the target arm $a^*$ in $\Theta(T)$ time steps. In the meantime, the attacker wants to control its attack cost by poisoning as infrequently as possible in order to avoid being detected. Note that $\epsilon_t = 0$ is considered as no attack.

By the linearity of expectation and $\eta_t$ is a zero-mean noise,

$$R(T) = \sum_{a \in \mathcal{A}} \left( \max_{i \in \mathcal{A}} \mu_i - \mu_a \right) \mathbb{E}[N_a(T)], \tag{5}$$

where $N_a(t)$ is the number of pulling arm $a$ up to time $t$. Thus, the attack goal means that the attacker wants the bandit algorithm to incur a linear regret by incurring only a sub-linear attack cost.

## 3. Offline Attacks

In this section, we introduce the offline attack framework to stochastic bandits. The updates of the bandit algorithm happen in mini-batches of size $T$[1]. Between these consecutive updates, the bandit algorithm follows a fixed algorithm obtained from the last update. This allows the attacker to poison the historical data before the update and force the algorithm to pull a target arm $a^*$ with high probability.

Let $m_a$ be the number of times arm $a$ was pulled up to time $T$, i.e., $m_a := N_a(T)$. For each arm $a \in \mathcal{A}$, let $\vec{y}_a \in \mathcal{R}^{m_a}$ be the corresponding reward vector returned from the bandit environment when arm $a$ was pulled. That is, $\vec{y}_a := (r_t : a_t = a)^T$. Let $\vec{\epsilon}_a \in \mathcal{R}^{m_a}$ be the poisoning attack strategy of the attacker, i.e., $\vec{\epsilon}_a := (\epsilon_t : a_t = a)^T$. The poisoned reward vector for arm $a$ after the attack becomes $\vec{y}_a + \vec{\epsilon}_a$. To avoid being detected, the attacker hopes to make the poisoning $\vec{\epsilon}_a$ as small as possible. Without loss of generality, we consider the $l^2$-norm attack cost in the offline attacks. We have that

$$C(T)^2 = \sum_{t=1}^{T} \epsilon_t^2 = \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2. \tag{6}$$

Thus, the attacker's offline attack problem can be formulated as the following optimization problem $P$,

$$P: \min_{\vec{\epsilon}_a : a \in \mathcal{A}} \quad \sum_{a \in \mathcal{A}} ||\vec{\epsilon}_a||_2^2 \tag{7}$$

$$s.t. \quad \mathbb{P}\{a_{T+1} = a^*\} \ge 1 - \delta, \tag{8}$$

for some error tolerance $\delta > 0$. Note that we define the attack goal as forcing the bandit algorithm to pull arm $a^*$ at the next round with high probability. This is because there are some randomized bandit algorithms, such as $\epsilon$-greedy and Thompson Sampling. It is not feasible to force the randomized algorithm to pull a target arm with probability 1. But it is possible to hijack the behavior of the randomized algorithm with high probability.

[1]The batch size $T$ is a relatively large integer compared to $K$.

**Proposition 1.** *Given some error tolerance $\delta > 0$. If $\{\vec{\epsilon}_a^*\}_{a\in\mathcal{A}}$ is the optimal solution of problem $P$, then it is the optimal offline attack strategy for the attacker.*

The proof of Proposition 1 follows from equation (6) and the definition of offline attacks. Note that the constraint of problem $P$ depends on the bandit algorithm. Now, we assume that the attacker knows the bandit algorithm and we introduce algorithm-specific offline attack strategies derived from the optimization problem $P$ for three popular bandit algorithms, $\epsilon$-greedy, UCB and Thompson Sampling. For simplicity, we denote the post-attack empirical mean observed by the bandit algorithm at the end of round $t$ as

$$\tilde{\mu}_a(t) := \frac{1}{N_a(t)} \sum_{\tau=1}^{t} r'_\tau \mathbb{1}\{a_\tau = a\}. \tag{9}$$

### 3.1. Offline attack on $\epsilon$-greedy

Now, we derive the offline attack strategy for the $\epsilon$-greedy algorithm, which is a randomized algorithm with some decreasing rate function $\alpha_t$. Typically, $\alpha_t = \Theta(1/t)$. At each time $t$, the $\epsilon$-greedy algorithm pulls an arm

$$a_t = \begin{cases} \text{draw uniformly over } \mathcal{A}, & \text{w.p. } \alpha_t \\ \arg\max_{a\in\mathcal{A}} \tilde{\mu}_a(t-1), & \text{otherwise} \end{cases}. \tag{10}$$

At time step $T+1$, the $\epsilon$-greedy algorithm uniformly samples an arm from the arm set $\mathcal{A}$ with probability $\alpha_{T+1}$, which cannot be controlled by the attacker. Then, we set the error tolerance as $\delta = \frac{K-1}{K}\alpha_{T+1}$ since the target arm $a^*$ may also be pulled in the uniform sampling. Thus, the attacker poisons the reward such that the target arm $a^*$ has the largest empirical mean. After the attack, the $\epsilon$-greedy algorithm pulls an arm $a_{T+1}$ at time $T+1$ such that $\mathbb{P}\{a_{T+1} = a^*\} = 1-\delta$. In order to make the target arm the unique arm with the largest empirical mean, we introduce a margin parameter $\xi > 0$. So the attacker's optimization problem for attacking $\epsilon$-greedy algorithm is the following problem $P_1$,

$$P_1 : \min_{\vec{\epsilon}_a : a\in\mathcal{A}} \quad \sum_{a\in\mathcal{A}} ||\vec{\epsilon}_a||_2^2 \tag{11}$$

$$s.t. \quad \tilde{\mu}_{a^*}(T) \geq \tilde{\mu}_a(T) + \xi, \quad \forall a \neq a^*, \tag{12}$$

where $\tilde{\mu}_a(T) = (\vec{y}_a + \vec{\epsilon}_a)^T \mathbb{1}/m_a$ and $\mathbb{1}$ is the vector that each element is 1. The condition in the above optimization implies that the $\epsilon$-greedy algorithm will play the target arm $a^*$ at the time step $T+1$ with probability of at least $1 - \frac{K-1}{K}\alpha_{T+1}$. The following result shows that there exists at least one optimal solution of problem $P_1$, i.e., one optimal offline attack for the $\epsilon$-greedy algorithm.

**Theorem 1.** *Given any margin parameter $\xi > 0$. For any reward instance $\{\vec{y}_a\}_{a\in\mathcal{A}}$, there exists at least one optimal solution of problem $P_1$, which is a quadratic program with linear constraints. Hence, there exists at least one optimal offline attack for the $\epsilon$-greedy algorithm.*

The proof of Theorem 1 is provided in Section A.1 in the supplementary material. Note that the error tolerance parameter, $\delta = \frac{K-1}{K}\alpha_{T+1}$, depends on the rate function $\alpha_t$ of the $\epsilon$-greedy algorithm, which is not controllable by the attacker. This counts the exploration introduced by the bandit algorithm's inner randomization, which can not be manipulated by the attacker. However, the attacker can wait for some large enough $T$ since the rate function $\alpha_t$ goes to zero finally. Moreover, the attacker's strategy (problem $P_1$) does not depend on the rate function, i.e., the attacker does not require the knowledge of the parameter of the $\epsilon$-greedy.

### 3.2. Offline attack on UCB

Now we derive the offline attack strategy for the classical UCB algorithm. For each time $t$, the UCB algorithm pulls the arm with the highest UCB index:

$$a_t = \arg\max_{a\in\mathcal{A}} u_a(t) := \tilde{\mu}_a(t-1) + 3\sigma\sqrt{\frac{\log t}{N_a(t-1)}}. \tag{13}$$

The UCB algorithm pulls the target arm $a^*$ at time $T+1$ if and only if the UCB index of arm $a^*$ is the unique largest one. Thus, the attacker can manipulate the rewards to satisfy the condition. Given any margin parameter $\xi > 0$, the attacker's optimization problem becomes

$$P_2 : \min_{\vec{\epsilon}_a : a\in\mathcal{A}} \quad \sum_{a\in\mathcal{A}} ||\vec{\epsilon}_a||_2^2 \tag{14}$$

$$s.t. \quad u_{a^*}(T+1) \geq u_a(T+1) + \xi, \quad \forall a \neq a^*$$

The condition in the above optimization implies that the UCB algorithm will pull the target arm after the poisoning attack. The following result shows that there exists at least one optimal solution of problem $P_2$, i.e., one optimal offline attack for the UCB algorithm.

**Theorem 2.** *Given any margin parameter $\xi > 0$. For any reward instance $\{\vec{y}_a\}_{a\in\mathcal{A}}$, there exists at least one optimal solution of problem $P_2$, which is a quadratic program with linear constraints. Hence, there exists at least one optimal offline attack for the UCB algorithm.*

The proof of Theorem 2 is similar to the proof of Theorem 1. Note that the above attack strategy holds for any error tolerance $\delta$ since UCB algorithm is not randomized.

### 3.3. Empirical mean based bandit algorithms

One insight from our offline attacks on the $\epsilon$-greedy algorithm and the UCB algorithm is that the empirical mean based algorithms are vulnerable to attack. This is because the empirical mean related constraints are linear and

non-empty. Then the optimization problem $P$ becomes a quadratic problem with non-empty linear constraints, which can be solved efficiently. This result holds for many other variants of UCB algorithms and Explore-Then-Commit algorithms (Garivier et al., 2016) (which uniformly samples the arm set in the first exploration phase and commit to the arm with the largest empirical mean in the second commitment phase).

### 3.4. Beyond empirical means: Thompson Sampling for Gaussian distributions

We have shown that the empirical mean based algorithms are not secure against the offline attack. It would appear that Bayesian algorithms should be more robust to the offline attack as the constraint of problem $P$ becomes non-tractable. Unfortunately, we find that Thompson Sampling, a popular Bayesian algorithm, is also vulnerable to the data poisoning.

Now, we derive the attack strategy for Thompson Sampling for Gaussian distributions[2]. In other words, the noise $\eta_t$ is i.i.d. sampled from Gaussian distribution $\mathcal{N}(0, \sigma^2)$. Thompson Sampling for Gaussian distribution with the Jeffreys prior (Korda et al., 2013) works as the following. For each time $t$, the algorithm samples $\theta_a(t)$ from the posterior distribution $\mathcal{N}(\tilde{\mu}_a(t-1)/\sigma^2, \sigma^2/N_a(t-1))$ for each arm $a$ independently. Then the algorithm pulls the arm with the highest sample value, i.e., $a_t = \arg\max_{a\in\mathcal{A}} \theta_a(t)$.

Let $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ be the probability density function (pdf) of the standard Gaussian distribution and $\Phi(x)$ be the corresponding cumulative distribution function (cdf). For simplicity, we denote $\phi_a(x)$ as the pdf of the Gaussian distribution $\mathcal{N}(\tilde{\mu}_a(T)/\sigma^2, \sigma^2/N_a(T))$ for each arm $a$ and $\Phi_a(x)$ as the corresponding cdf. Since we are interested in the samples in time $T+1$, we omit time index in the following discussion. By the law of total expectation, we have that

$$\begin{aligned}\mathbb{P}\{a_{T+1} = a^*\} &= \mathbb{P}\{\cap_{a\neq a^*}\theta_{a^*} > \theta_a\} \quad (15)\\ &= \int_{-\infty}^{+\infty} \mathbb{P}\{\cap_{a\neq a^*}\theta_a < x | \theta_{a^*} = x\}\phi_{a^*}(x)\,dx\\ &= \int_{-\infty}^{+\infty} \left(\Pi_{a\neq a^*}\mathbb{P}\{\theta_a < x|\theta_{a^*} = x\}\right)\phi_{a^*}(x)\,dx\\ &= \int_{-\infty}^{+\infty} \left(\Pi_{a\neq a^*}\Phi_a(x)\right)\phi_{a^*}(x)\,dx. \quad (16)\end{aligned}$$

The Equation (16) is complicated to compute and analyze, which makes the instantiation of the offline attack problem $P$ non-trivial. To address this challenge, we derive a sufficient constraint of the original constraint and make a relaxation of the original problem $P$. By the union bound, we have that

$$\begin{aligned}\mathbb{P}\{a_{T+1}\neq a^*\} &= \mathbb{P}\{\cup_{a\neq a^*}\theta_{a^*} < \theta_a\} \leq \sum_{a\neq a^*}\mathbb{P}\{\theta_{a^*} < \theta_a\}\\ &= \sum_{a\neq a^*}\Phi\left(\frac{\tilde{\mu}_a(T) - \tilde{\mu}_{a^*}(T)}{\sigma^3\sqrt{1/m_a + 1/m_{a^*}}}\right) \quad (17)\end{aligned}$$

Thus, we are ready to introduce the attacker's problem for Thompson Sampling for Gaussian distributions,

$$P_3: \min_{\vec{\epsilon}_a: a\in\mathcal{A}} \quad \sum_{a\in\mathcal{A}} ||\vec{\epsilon}_a||_2^2 \quad (18)$$

$$s.t. \quad \sum_{a\neq a^*}\Phi\left(\frac{\tilde{\mu}_a(T) - \tilde{\mu}_{a^*}(T)}{\sigma^3\sqrt{1/m_a + 1/m_{a^*}}}\right) \leq \delta \quad (19)$$

$$\tilde{\mu}_a(T) - \tilde{\mu}_{a^*}(T) \leq 0, \quad \forall a \neq a^* \quad (20)$$

The constraint of problem $P_3$ is a sufficient condition to the constraint of the problem $P$ by Equation (17). Note that the linear constraints (20) are redundant since $\Phi(0) = 0.5$ and $\delta < \frac{K-1}{2}$ is usually satisfied. The following proposition shows that the constraint set of problem $P_3$ is convex.

**Proposition 2.** *The constraint set formed by Equations (19) and (20) is convex.*

The proof of Proposition 2 is provided in Section A.3 in the supplementary material. The following result shows that there exists at least one optimal solution of problem $P_3$ that is a feasible offline attack for the Thompson Sampling for Gaussian distributions.

**Theorem 3.** *Given any error tolerance $\delta > 0$. For any reward instance $\{\vec{y}_a\}_{a\in\mathcal{A}}$, there exists at least one optimal solution of problem $P_3$, which is a quadratic program with convex constraints. Hence, there exists at least one feasible offline attack for the Thompson Sampling algorithm for Gaussian distributions.*

By Proposition 2, the proof of Theorem 3 is similar to the proof of Theorem 1. Note that the above attack strategy is not the optimal attack strategy formulated by $P$. However, it is easy to compute since problem $P_3$ is a quadratic program with convex constraints. Another relaxation of problem $P$ is presented in Section A.4 in the supplementary material.

## 4. Online Attacks

In this section, we study online attacks against stochastic bandits. In the online attack setting, the bandit algorithm updates its policy at each round. The attacker eavesdrops on the decision (i.e., $a_t$) of the bandit algorithm and poisons the reward by adding an arbitrary $\epsilon_t \in \mathcal{R}$. Hence the reward observed by the bandit algorithm is $r'_t = r_t + \epsilon_t$. Without loss of generality, we consider the $l^1$-norm attack cost. That is, the cost of the attacker for round $t$ is $|\epsilon_t|$. Recall that $N_a(t)$ is the number of pulling arm $a$ up to time $t$.

[2]Thompson Sampling needs distribution models and Gaussian distribution is popular and well-studied in the literature. The idea of problem relaxation here can be extended to other distributions.

Without the attacks, the bandit algorithm is a uniformly good policy such that it achieves $O(\log T)$ regret[3], i.e., $\mathbb{E}[\sum_a(\max_i \mu_i - \mu_a)N_a(T)] = O(\log T)$ for any problem instance $\{\mu_a\}_{a\in\mathcal{A}}$. Moreover, the expected number of pulling the optimal arm (with the highest expected reward) is $T - o(T)$.

The goal of the attacker is to force the bandit algorithm to pull the sub-optimal target arm $a^*$ as much as possible and pays the least possible total cost. Formally, the attacker wants the bandit algorithm to receive linear expected regret, i.e., $\mathbb{E}[N_{a^*}(T)] = T - o(T)$, and pays the expected total cost $\mathbb{E}[\sum_t^T |\epsilon_t|] = O(\log T)$. In other words, the attacker wants to manipulate the rewards so that the bandit algorithm considers the target arm as the best arm in the long term.

### 4.1. Oracle constant attacks

The fact that the attacker does not know the expected rewards $\{\mu_a\}_{a\in\mathcal{A}}$ is challenging because otherwise the attacker can perform the attack trivially. Suppose the attacker knows the expected rewards, then the attacker can choose the following oracle attack strategy,

$$\epsilon_t = -\mathbb{1}\{a_t \neq a^*\}[\mu_{a_t} - \mu_{a^*} + \xi]^+, \tag{21}$$

for some margin parameter $\xi > 0$. Note that $[\cdot]^+$ indicates the function such that $[x]^+ = \max\{x, 0\}$ and $\mathbb{1}\{\cdot\}$ is the indicator function. By this attack, the bandit algorithm sees a poisoned bandit problem, where the target arm $a^*$ is the optimal arm and all the other arms are at least $\xi$ below the target arm. Thus, the bandit algorithm pulls the target arm with $\mathbb{E}[N_{a^*}(T)] = T - o(T)$ and pays the total cost $\mathbb{E}[\sum_t^T |\epsilon_t|] = O(\log T)$ since $\epsilon_t$ are bounded. This result has been shown in (Jun et al., 2018) as the following.

**Proposition 3.** *(Proposition 1 in (Jun et al., 2018)) Assume that the bandit algorithm achieves an $O(\log T)$ regret bound. Then the oracle attack with $\xi > 0$ succeeds, i.e., $\mathbb{E}[N_{a^*}(T)] = T - o(T)$. Furthermore, the expected attack cost is $O(\sum_{i\neq a^*}[\mu_i - \mu_{a^*} + \xi]^+ \log T)$.*

The insight obtained from Proposition 3 is that the attacker does not need to attack in round $t$ if $a_t = a^*$. This helps us to design attack strategies that are similar to the oracle attack. When the ground truth $\{\mu_a\}_{a\in\mathcal{A}}$ is not known to the attacker, the attacker may guess some upper bound $\{C_a\}_{a\neq a^*}$ on $\{[\mu_a - \mu_{a^*}]^+\}_{a\neq a^*}$ and perform the following oracle constant attack,

$$\epsilon_t = -\mathbb{1}\{a_t \neq a^*\}C_{a_t}. \tag{22}$$

The following result shows the sufficient and necessary conditions for the oracle constant attack to be successful.

[3]The results in this section directly apply to the problem-independent regret bounds and high probability bounds.

**Proposition 4.** *Assume that the bandit algorithm achieves an $O(\log T)$ regret bound. Then the constant attack with $\{C_a\}_{a\neq a^*}$ succeeds if and only if $C_a > [\mu_a - \mu_{a^*}]^+$, $\forall a \neq a^*$. If the attack succeeds, then the expected attack cost is $O(\sum_{a\neq a^*} C_a \log T)$.*

The proof of Proposition 4 is provided in Section B.1 in the supplementary material. Proposition 4 shows that the attacker has to know the unknown bounds on $\{[\mu_a - \mu_{a^*}]^+\}_{a\neq a^*}$ to guarantee a successful constant attack. Moreover, the oracle constant attack is non-adaptive to the problem instance since some upper bound $C_a$ can be much larger than the quantity $[\mu_a - \mu_{a^*}]^+$ so that the attacker is paying unnecessary attack cost compared to the oracle attack. To address this challenge, we propose an adaptive constant attack that can learn the bandit environment in an online fashion and perform the attack adaptively.

### 4.2. Adaptive attack by constant estimation

Now, we are ready to propose the adaptive attack strategy. The idea is that the attacker can update upper bounds on the unknown quantity $\{[\mu_a - \mu_{a^*}]^+\}_{a\neq a^*}$ based on the empirical mean observed by the attacker. Then the attacker performs the constant attack based on the estimated upper bounds. Note that the attacker observes the tuple $(a_t, r_t)$ at each time $t$ and is able to obtain an unbiased empirical mean. Let $\hat{\mu}_a(t)$ be the pre-attack empirical mean observed by the attacker at time $t$, that is

$$\hat{\mu}_a(t) := \frac{1}{N_a(t)}\sum_{\tau=1}^{t} r_\tau \mathbb{1}\{a_\tau = a\}. \tag{23}$$

Given any $\delta \in (0, 1)$, we define a function $\beta(n)$ as

$$\beta(n) = \sqrt{\frac{2\sigma^2}{n}\log\frac{\pi^2 K n^2}{3\delta}}. \tag{24}$$

Then, the following result shows that $\{\beta(N_a(t))\}_{a\in\mathcal{A}}$ is a high-probability bound on the pre-attack empirical mean of all arms on all rounds. Define the event

$$E = \{\forall a \in \mathcal{A}, \forall t : |\hat{\mu}_a(t) - \mu_a| < \beta(N_a(t))\}. \tag{25}$$

**Lemma 1.** *(Lemma 1 in (Jun et al., 2018)) For $\delta \in (0, 1)$, $\mathbb{P}(E) > 1 - \delta$.*

The proof of Lemma 1 follows from Hoeffding inequality and union bound. Lemma 1 shows that with probability at least $1 - \delta$, we have that the expected reward $\mu_a$ is within the interval $(\hat{\mu}_a(t) - \beta(N_a(t)), \hat{\mu}_a(t) + \beta(N_a(t)))$ centered at the pre-attack empirical mean $\hat{\mu}_a(t)$. Thus, we obtain an upper bound on the quantity $\mu_a - \mu_{a^*}$ and propose the following Adaptive attack by Constant Estimation (ACE),

$$\epsilon_t = -\mathbb{1}\{a_t \neq a^*\}[\hat{\mu}_{a_t}(t) - \hat{\mu}_{a^*}(t) + \beta(N_{a_t}(t)) + \beta(N_{a^*}(t))]^+.$$

The following shows that the ACE is successful for any bandit algorithm *without the knowledge of the bandit algorithm*.

**Theorem 4.** *Given any $\delta \in (0, 0.5)$, assume that the bandit algorithm achieves an $O(\log T)$ regret bound with probability at least $1-\delta$. With probability at least $1-2\delta$, the ACE attacker forces the bandit algorithm to play the target arm $a^*$ in $N_{a^*}(T)$ times, such that $N_{a^*}(T) = T - o(T)$, using an instantaneous attack cost,*

$$|\epsilon_t| \leq \mathbb{1}\{a_t \neq a^*\}\big([\mu_{a_t} - \mu_{a^*}]^+ + 2\beta(N_{a_t}(t)) + 2\beta(N_{a^*}(t))\big)\,.$$

*In particular, with probability at least $1-2\delta$, the ACE attack is successful and the accumulated attack cost is*

$$\sum_{t=1}^{T} |\epsilon_t| \leq O\left(\sum_{a \neq a^*} \big([\mu_a - \mu_{a^*}]^+ + 4\beta(1)\big) \log T\right).$$

The proof of Theorem 4 is provided in Section B.2 in the supplementary material. Theorem 4 shows that the ACE is universally successful for any bandit algorithm, without knowing any prior information on $\{\mu_a\}_{a\in\mathcal{A}}$. Besides, the ACE incurs an high-probability accumulated attack cost as small as that of the oracle attack[4] with only an additional bounded additive term, $O(4\beta(1)K\log T)$. That is, the ACE is close to the hindsight-oracle attack strategy. Moreover, the ACE even requires no knowledge of the bandit algorithm. This is an advantage over the algorithm-dependent online attack strategies designed in (Jun et al., 2018) since the attacker may not know which bandit algorithm the learner is in practice. As far as we know, this is the first negative result showing that there is no robust bandit algorithm that can be immune to the adaptive online attack. This exposes a significant security threat to the bandit learning systems.

## 5. Numerical Results

In this section, we run simulations on attacking $\epsilon$-greedy, UCB and Thompson Sampling algorithms to illustrate our theoretical results. All the simulations are run in MATLAB and the codes can be found in the supplemental materials. Note that we implement $\epsilon$-greedy with $\alpha_t = \frac{1}{t}$.

### 5.1. Offline attacks

To study the effectiveness of the offline attacks, we consider the following experiment. The bandit has $K = 5$ arms and the reward noise is a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.1$. The attacker's target arm is arm $K$ such that $\mu_K = 0$, while the expected rewards of other arms are uniformly distributed in the unit interval $[0, 1]$. We set $T = 1000$ and the error tolerance to $\delta = 0.05$.

In each attack trial, we first generate the instance of the bandit by drawing the expected rewards from the uniform distribution on $[0, 1]$. Then we run the bandit algorithm for $T$ rounds and collect all the historical data. Without any attack, the bandit algorithm would have converged to some optimal arm in the trial, which is not the target arm as the target arm is the one with the least payoff. Then we set the margin parameter as $\xi = 0.001$ and run the corresponding offline attacks. The attack strategy is the solution of the optimization problem $P_i$. Since all the problems are quadratic program with linear (convex) constraints, they can be solved by standard optimization tools.

We run 1000 attack trials. Note that the attack cost depends on the instance of the bandit. To evaluate the attack cost, we use the poisoning effort ratio (Ma et al., 2018):

$$\frac{||\vec{\epsilon}||_2}{||\vec{y}||_2} = \sqrt{\frac{\sum_{a\in\mathcal{A}} ||\vec{\epsilon}_a||_2^2}{\sum_{a\in\mathcal{A}} ||\vec{y}_a||_2^2}}. \tag{26}$$

The poisoning effort ratio measures the ratio of the total cost over the norm of the original rewards. To evaluate the attack effectiveness, we check whether the poisoned data satisfy the constraint of the optimization problem $P$. If so, the bandit algorithm will play the target arm with probability at least $1-\delta$.

Figure 3 shows the results of the attack against the $\epsilon$-greedy, UCB and Thompson Sampling. First, the attack strategies are effective as all the attacks are successful. Second, the attack costs are small as shown in the histograms. The ratios of attacking $\epsilon$-greedy, UCB and Thompson Sampling are less than $10\%$, $2\%$ and $5\%$.

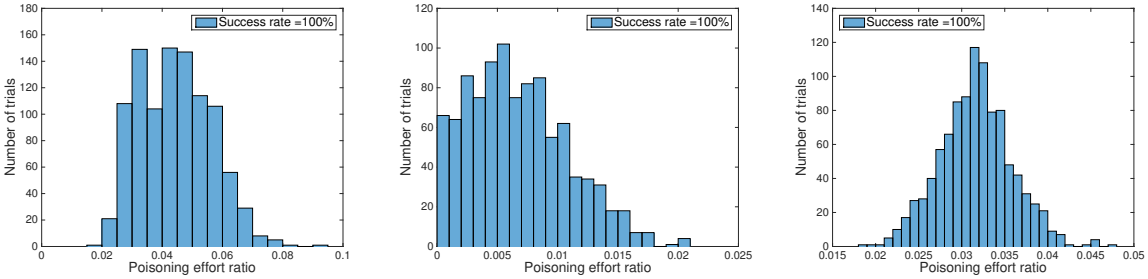
### 5.2. Online attacks

We compare our adaptive attack strategy with two attack strategies proposed by (Jun et al., 2018). Note that these two attacks are against $\epsilon$-greedy and UCB algorithm and require the knowledge of the algorithm. In contrast, we highlight that our attack strategy ACE is an universal attack strategy against any bandit algorithm.

We consider the following experiment. The bandit has two arms. The expected rewards of arms 1 and 2 are $\mu_1 = \Delta$ and $\mu_2 = 0$ with $\Delta > 0$. The attacker's target is arm 2. The noise of the rewards is i.i.d. sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.1$. We set the error tolerance to $\delta = 0.05$ and time horizon to $T = 10^5$ rounds. For the implementations of the attack strategies proposed by (Jun et al., 2018), we choose the tuning parameter $\Delta_0 = \sigma$, which is suggested by (Jun et al., 2018) when $T$ is not known to the attacker. We run 100 attack trials with different $\Delta$ values.

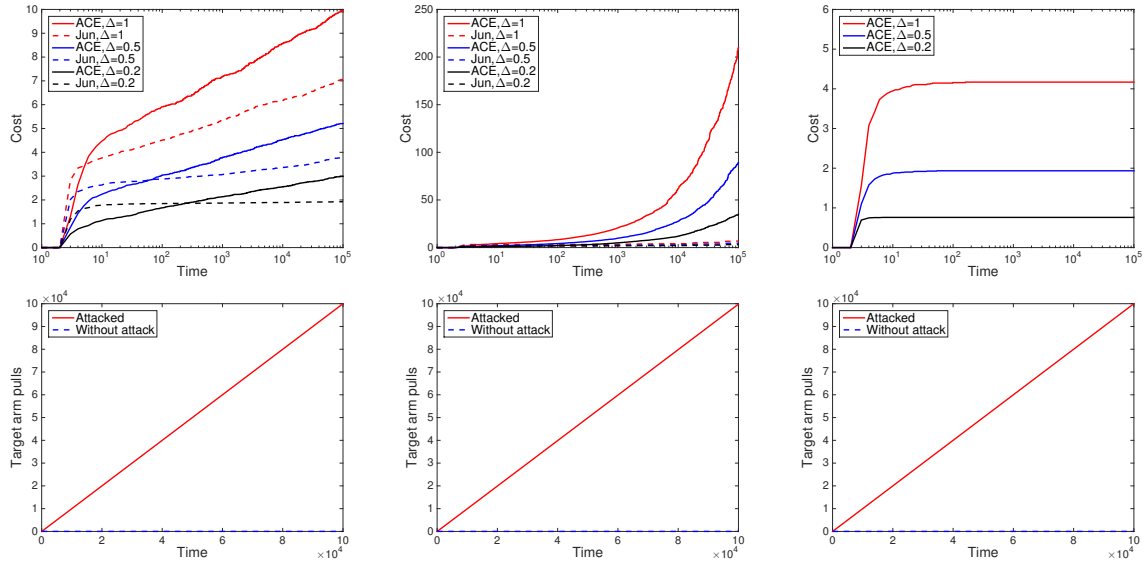
Figure 4 shows the average attack cost and number of target arm pulls in the online attacks. Note that the target arm pulls are the cases when $\Delta = 1$. First, we compare the number of target arm pulls with ACE attack and without. ACE attack dramatically forces the bandit algorithm to pull the target arm linearly in time. Second, the attack costs are relatively

[4] High-probability bounds can be adapted from Proposition 3.




(a) Attack on $\epsilon$-greedy algorithm (b) Attack on UCB algorithm (c) Attack on Thompson Sampling

Figure 3: Histograms of poisoning effort ratio in the offline attacks.




(a) Attack on $\epsilon$-greedy algorithm (b) Attack on UCB algorithm (c) Attack on Thompson Sampling

Figure 4: Attack cost and target arm pulls in the online attacks.

small compared to the loss of the bandit algorithm, which is linear in time. Generally, the attack costs of ACE attack are bounded by $O(\log T)$ and increase as the reward gap $\Delta$ becomes larger. These verify the result of Theorem 4. On the other hand, the attack costs on Thompson Sampling and $\epsilon$-greedy are relatively smaller than that of UCB. This is because Thompson Sampling and $\epsilon$-greedy converges to the "optimal" arm very fast while the exploration for "non-optimal" arm may still increase over time. Finally, compared to the algorithm-specific attacks proposed by (Jun et al., 2018), the attack cost of ACE on $\epsilon$-greedy is slightly worse while the attack cost of ACE on UCB is much larger than Jun's attack. In the case of attacking UCB algorithm, our universal attack strategy takes more cost than the algorithm-specific attack, but without the need to know the algorithm.

## 6. Conclusion

In this work, we study the open problem of data poisoning attacks on bandit algorithms. We propose an offline attack framework for the stochastic bandits and propose three algorithm-specific offline attack strategies against $\epsilon$-greedy, UCB and Thompson Sampling. Then, we study an online attack against the bandit algorithms and propose the adaptive attack strategy that can hijack the behavior of any bandit algorithm without requiring the knowledge of the bandit algorithm. Our theoretical results and simulations show that the bandit algorithms are vulnerable to the poisoning attacks in both online and offline setting.

## Acknowledgements

This work has been supported in part by a grant from DTRA (HDTRA1-14-1-0058), a grant from Office of Naval Research (N00014-17-1-2417), grants from National Science foundation (CNS-1446582, CNS-1518829, CNS-1409336) and a grant from Army Research Office (WN11NF-15-1-0277). This work has also been supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT), (2017-0-00692, Transport-aware Streaming Technique Enabling Ultra Low-Latency AR/VR Services).

## References

Alfeld, S., Zhu, X., and Barford, P. Data poisoning attacks against autoregressive models. In *AAAI*, pp. 1452–1458, 2016.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1467–1474, 2012.

Chapelle, O., Manavoglu, E., and Rosales, R. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):61, 2015.

Garivier, A., Lattimore, T., and Kaufmann, E. On explore-then-commit strategies. In *Advances in Neural Information Processing Systems*, pp. 784–792, 2016.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Gupta, A., Koren, T., and Talwar, K. Better algorithms for stochastic bandits with adversarial corruptions. *arXiv preprint arXiv:1902.08647*, 2019.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Jun, K.-S., Li, L., Ma, Y., and Zhu, X. Adversarial attacks on stochastic bandits. *arXiv preprint arXiv:1810.12188*, 2018.

Korda, N., Kaufmann, E., and Munos, R. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems*, pp. 1448–1456, 2013.

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, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

Liu, F., Wang, S., Buccapatnam, S., and Shroff, N. Ucboost: a boosting approach to tame complexity and optimality for stochastic bandits. *arXiv preprint arXiv:1804.05929*, 2018.

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

Ma, Y., Jun, K.-S., Li, L., and Zhu, X. Data poisoning attacks in contextual bandits. *arXiv preprint arXiv:1808.05760*, 2018.

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

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Wang, Y. and Chaudhuri, K. Data poisoning attacks against online learning. *arXiv preprint arXiv:1808.08994*, 2018.

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