ACTION-MANIPULATION ATTACKS ON STOCHASTIC BANDITS

Guanlin Liu and Lifeng Lai

Department of ECE, University of California, Davis
Email:{glnliu,lflai}@ucdavis.edu

ABSTRACT

As stochastic multi-armed bandit model has many important applications, understanding the impact of adversarial attacks on this model is essential for the safe applications of this model. In this paper, we propose a new class of attack named action-manipulation attack, where an adversary can change the action signal selected by the user. We investigate the attack against a very popular and widely used bandit algorithm: Upper Confidence Bound (UCB) algorithm. Without knowledge of mean rewards of arms, our proposed attack scheme can force the user to pull a target arm very frequently by spending only logarithm cost.

***Index Terms*—** Stochastic bandits, action-manipulation attack, UCB.

1. INTRODUCTION

Multiple-armed bandits (MABs), a simple but very powerful framework of online learning that makes decisions over time under uncertainty, has many applicants in a variety of scenarios such as displaying advertisements [1], articles recommendation [2], and search engines [3], to name a few. In order to develope trustworthy machine learning systems, understanding adversarial attacks on learning systems and building robust defense mechanisms has attracted significant research interests [4, 5, 6, 7, 8, 9, 10, 11]. Of particular relevance to our work is a line of interesting recent work on online reward-manipulation attacks on MABs [12, 13, 14]. In the reward-manipulation attacks, there is an adversary that can change the reward signal from the environment. In particular, [12] proposes an interesting attack strategy that can manipulate a user, who runs either ϵ-Greedy and or Upper Confidence Bound (UCB) algorithm, into selecting a target arm while only spending effort that grows in logarithmic order. [13] proposes an optimization based framework for offline reward-manipulation attacks. Furthermore, it develops an online attack strategy that is effective in attacking any bandit algorithm that has a regret scaling in logarithm order. In the defense part, using a multi-layer approach, [14] introduces a bandit algorithm that is robust to reward-manipulation attacks under certain attack cost constraint.

This paper introduce a new class of attacks on MABs named reward-manipulation attack. In the action-manipulation attack, an attacker can change the action selected by the user to another action. The user will then receive a reward from the environment corresponding to the action chosen by the attacker. Compared with the reward-manipulation attacks discussed above, the action-manipulation attack is more difficult to carry out. In particular, as the action-manipulation attack only changes the action, it can be viewed as manipulating the rewards to be random rewards drawn from some unknown distributions of the action chosen by the attacker. This is in contrast to reward-manipulation attacks where an attacker can change the rewards to any value. Despite this challenge, we design an effective action-manipulation attack scheme to attack UCB, a popular and widely used bandit algorithm [15]. Our scheme aims to force the user to pull a target arm frequently. We assume that the attacker does not know the mean rewards of arms, as otherwise the attacker can perform the attack trivially. Although without the knowledge of the mean rewards of arms, the attacker can find a possible worst arm according to the empirical mean reward of all arms and attacks when the user pulls a non-target arm. Our analysis shows that, if the target arm selected by the attacker is not the worst arm, our action-manipulation attacks can successfully manipulate the user to select the target arm with an only logarithmic cost. In particular, our attack scheme can force the user to pull the target arm $T - O(\log(T))$ times over T rounds, with total attack cost being only $O(\log(T))$. On the other hand, we also show that, if the target arm is the worst arm, no attack algorithm with logarithmic cost can force the user to pull the worst arm more than $T - O(T^{\alpha})$ times.

This paper is organized as follows. In Section 2, we introduce the problem formulation. In Section 3, we present the proposed attack scheme and analyze the attack cost. In Section 4, we present numerical results to evaluate our attack schemes. Finally, we offer concluding remarks in Section 5.

2. PROBLEM FORMULATION

In this section, we introduce the problem formulation. We consider the standard multi-armed stochastic bandit problems

The work of G. Liu and L. Lai was supported by National Science Foundation under Grants CCF-1717943, ECCS-1711468, CNS-1824553 and CCF-1908258.

setting. The environment consists of K arms, with each arm corresponds to a fixed but unknown reward distribution. The bandit algorithm, which is also called "user" in this paper, proceeds in discrete time $t = 1, 2, \ldots, T$, in which T is the total number of rounds. At each round t, the user pulls an arm (or action) $I_t \in \{1, \ldots, K\}$ and receives a random reward r_t drawn from the reward distribution of arm I_t. The user aims to maximize the cumulative rewards over T rounds. Denote by $\tau_i(t) := \{s : s \leq t, I_s = i\}$ the set of rounds up to t when the user chooses arm i, $N_i(t) := |\tau_i(t)|$ the number of arm i that the user pulls and

$$\hat{\mu}_i(t) := N_i(t)^{-1} \sum_{s \in \tau_i(t)} r_s \tag{1}$$

as the empirical mean reward of arm i.

In this paper, we consider an adversary setting, in which the attacker sits in-between the user and the environment. The attacker can monitor the actions of the user and reward signals from the environment. Furthermore, the attacker can introduce action-manipulation attacks on stochastic bandits. In particular, at each round t, after the user chooses an arm I_t, the attacker can manipulate the user's action by changing I_t to another arm $I_t^0 \in \{1, \ldots, K\}$. If the attacker decides not to attack, $I_t^0 = I_t$. The environment generates a random reward r_t from the reward distribution of post-attack arm I_t^0. Then the user and the attacker receive r_t from the environment. Note that the user does not know the attacker's manipulations and the presence of the attacker, and hence will still view r_t as the reward from action I_t and will still use (1) to compute the empirical mean reward of arm i. The attacker, on the other hand, knows that r_t is the reward from action I_t^0.

Without loss of generality and for notation convenience, we assume arm K is the "attack target" arm. The attacker's goal is to manipulate the user into pulling the target arm very frequently but by making attacks as rarely as possible. Define the set of rounds when the attacker decides to attack as $C := \{t : t \leq T, I_t^0 \neq I_t\}$. The cumulative attack cost is $|C|$, the number of rounds where the attacker decides to attack. The action-manipulation attack is different from reward-manipulation attacks introduced by interesting recent work [12, 13], where the attacker can change the reward signal from the environment.

In this paper, we assume that the reward distributions of arms follow σ^2-sub-Gaussian distributions with mean $\mu_1, \ldots, \mu_K$ respectively. Neither the user nor the attacker knows $\mu_1, \ldots, \mu_K$, but σ^2 is known to both the user and the attacker. Denote by μ_i the mean of arm i and define $\mu^* = \min_{i \in [k]} \mu_i$, $\Delta_i = \mu_i - \mu^*$ and $i^* \in \arg\min_{i \in [k]} \mu_i$.

In this paper, we focus on attacking the UCB algorithm [15]. In the UCB algorithm, the user initially pulls each of the K arms once in the first K rounds. After that, the user chooses arms according to

$$I_t = \arg\max_i \left\{ \hat{\mu}_i(t-1) + 3\sigma \sqrt{\log t / N_i(t-1)} \right\}. \tag{2}$$

Under the action-manipulation attack, as the user does not know that r_t is generated from arm I_t^0 instead of I_t, the empirical mean $\hat{\mu}_i(t)$ computed using (1) is not a proper estimate of the true mean reward of arm i anymore. On the other hand, the attack is able to obtain a good estimate of μ_i by

$$\hat{\mu}_i^0(t) := N_i^0(t)^{-1} \sum_{s \in \tau_i^0(t)} r_s, \tag{3}$$

where $\tau_i^0(t) := \{s : s \leq t, I_s^0 = i\}$ is the set of rounds up to t when the attacker changes an arm to arm i, and $N_i^0(t) = |\tau_i^0(t)|$ is the number of pulls of post-attack arm i up to round t. This information gap provides a chance for attack.

3. ATTACK STRATEGY AND COST ANALYSIS

In this section, we introduce the proposed action-manipulation attack on the UCB bandit algorithm and analyze the cost.

3.1. Attack Strategy

In this section, we assume that the target arm is not the worst arm, i.e., $\mu_K > \mu^*$. We will discuss the case where the target arm is the worst arm in Section 3.3.

Algorithm 1 Action-manipulation attack on UCB

Input:
The user's bandit algorithm, target arm K

for $t = 1, 2, \ldots$ **do**
  The user chooses arm I_t to pull according to UCB algorithm (2).
  if $t \leq K$ or $I_t = K$ **then**
    The attacker does not attack, and $I_t^0 = I_t$.
  else
    The attacker attacks and changes arm I_t to I_t^0 chosen according to (4).
  end if
  The environment generates reward r_t according to arm I_t^0.
  The attacker and the user receive r_t.
end for

The proposed attack strategy works as follows. In the first K rounds, the attacker does not attack. After that, at round t, if the user chooses a non-target arm I_t, the attacker changes it to arm I_t^0 that has the smallest lower confidence bound:

$$I_t^0 = \arg\min_i \left\{ \hat{\mu}_i^0(t-1) - \mathbf{CB}\left(N_i^0(t-1)\right) \right\}, \tag{4}$$

where

$$\mathbf{CB}(N) = \sqrt{\frac{2\sigma^2}{N} \log \frac{\pi^2 K N^2}{3\delta}}. \tag{5}$$

Here δ is a parameter that is related to the probability statements in the analytical results presented in Section 3.2.

If at round t the user chooses the target arm, the attacker does not attack. Thus the cumulative attack cost of our attack scheme is equal to the total of times when the non-target arms are selected by the user. The algorithm is summarized in Algorithm 1.

Here, we highlight the main idea why our attack strategy works. For $i \neq K$, we will show that this attack will ensure that $\hat{\mu}_i$ computed using (1) by the user converges to μ^*. On the other hand, as the attacker does not attack when the user selects K, $\hat{\mu}_K$ computed by the user will still converge to the true mean μ_K. Because the assumption that the target arm is not the worst, which implies that $\mu_K > \mu^*$, $\hat{\mu}_i$ could be smaller than $\hat{\mu}_K$. Then the non-target arms would pull rarely as $\hat{\mu}_i$ is smaller than $\hat{\mu}_K$. Hence, the attack cost would also be small. The rigorous analysis of the cost will be provided in Section 3.2.

3.2. Cost Analysis

.

To analyze the cost of the proposed scheme, we need to track $\hat{\mu}_i^0(t)$, the estimate obtained by the attacker using (3), and $\hat{\mu}_i(t)$, the estimate obtained by the user using (1).

The analysis of $\hat{\mu}_i^0(t)$ is relatively simple, as the attacker knows which arm is truly pulled and hence $\hat{\mu}_i^0(t)$ is the true estimate of the mean of arm i. Define event

$$E_1 := \{\forall i, \forall t > K : |\hat{\mu}_i^0(t) - \mu_i| < \mathbf{CB}(N_i^0(t))\}. \quad (6)$$

Roughly speaking, event E_1 is the event that the empirical mean computed by the attacker using (3) is close to the true mean. We have the following lemma showing that the attacker can accurately estimate the average reward to each arm.

Lemma 1. *(Lemma 1 in [12]) For $\delta \in (0,1)$, $\mathbb{P}(E_1) > 1-\delta$.*

The analysis of $\hat{\mu}_i(t)$ computed by the user is more complex. When the user pulls arm i, because of the action-manipulation attacks, the random reward may be drawn from different reward distributions. Define $\tau_{i,j}(t) := \{s : s \leq t, I_s = i \text{ and } I_s^0 = j\}$ as the set of rounds up to t when the user chooses arm i and the attacker changes it to arm j. Lemma 2 shows a high-probability confidence bounds of $\hat{\mu}_{i,j}(t) := N_{i,j}(t)^{-1} \sum_{s \in \tau_{i,j}(t)} r_s$, the empirical mean rewards of a part of arm i whose post-attack arm is j, where $N_{i,j}(t) := |\tau_{i,j}(t)|$. Define event

$$E_2 := \{\forall i \neq K, \forall j, \forall t > K : |\hat{\mu}_{i,j}(t) - \mu_j| < \sqrt{\frac{2\sigma^2}{N_{i,j}(t)} \log \frac{\pi^2 K^2 (N_{i,j}(t))^2}{3\delta}}\}. \quad (7)$$

Lemma 2. *For $\delta \in (0,1)$, $\mathbb{P}(E_2) > 1 - \frac{K-1}{K}\delta$.*

Events E_1, E_2 are important, as under these events, we can build a connection between $\hat{u}_i(t)$ and μ^*.

Lemma 3. *Under events E_1 and E_2 and Algorithm 1, we have*

$$\hat{\mu}_i(t) \leq u^* + \frac{1}{N_i(t)} \sum_{j \neq i^*} \frac{8\sigma^2}{\Delta_j} \log \frac{\pi^2 K t^2}{3\delta} + \sqrt{\frac{2\sigma^2 K}{N_i(t)} \log \frac{\pi^2 (N_i(t))^2}{3\delta}}, \forall i, t. \quad (8)$$

Lemma 3 shows an upper bound of the empirical mean reward of pre-attack arm i, for all arm $i \neq K$. Our main results is the following upper bound on the attack cost $|C|$.

Theorem 1. *With probability at least $1 - 2\delta$, when $T \geq \left(\frac{\pi^2 K}{3\delta}\right)^{\frac{2}{5}}$, the attacker can manipulate the user into pulling the target arm in at least $T - |C|$ rounds, using an attack cost*

$$|C| \leq \frac{K-1}{4\Delta_K^2} \left(3\sigma\sqrt{\log T} + \sqrt{2\sigma^2 K \log \frac{\pi^2 T^2}{3\delta}} + \left(\left(3\sigma\sqrt{\log T} + \sqrt{2\sigma^2 K \log \frac{\pi^2 T^2}{3\delta}}\right)^2 + 4\Delta_K \sum_{j \neq i^*} \frac{8\sigma^2}{\Delta_j} \log \frac{\pi^2 K T^2}{3\delta}\right)^{\frac{1}{2}}\right)^2. \quad (9)$$

The cost bound in Theorem 1 is complicated. The following corollary provides a simpler bound that is more explicit and interpretable.

Corollary 1. *Under the same assumptions in Theorem 1, the total attack cost $|C|$ is bounded*

$$O\left(K \frac{\sigma^2}{\Delta_K^2} \left(K + \sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j} + \sqrt{K \sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j}}\right) \log T\right), \quad (10)$$

and the total number of target arm pulls is $T - |C|$.

From Corollary 1, we can see that the attack cost scales as $\log T$. Two important constants $\frac{\sigma}{\Delta_K}$ and $\sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j}$ have impact on the prelog factor. In Section 4, we provide numerical examples to illustrate the effects of these two constants.

3.3. Attacks fail when the target arm is the worst arm

In our action-manipulation attack, the attacker can not force users to pull the worst arm very frequently by spending an only logarithmic cost. The main reason is that, when the target arm is the worst, the average reward of each arm is larger or equal to that of the target arm. As the result, our attack scheme is not able to ensure that the target arm has a higher expected reward than the user's estimate of the rewards

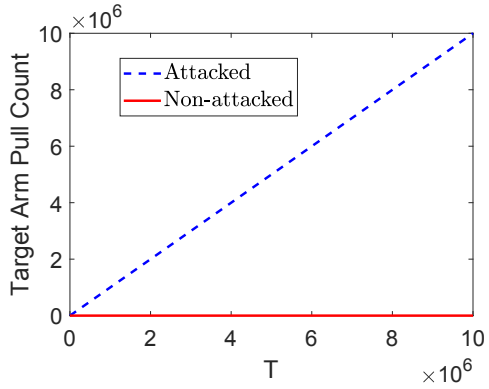

Fig. 1. Number of target arm pulls

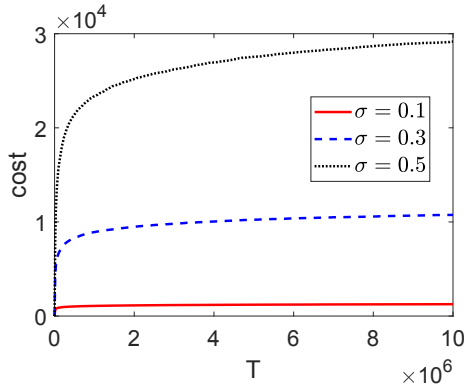

Fig. 2. Attack cost as $\frac{\sigma}{\Delta_K}$ varies

of other arms. In fact, the following theorem shows that all action-manipulation attack can not manipulate the user into pulling the worst arm $T - O(\log(T))$ by spending an only logarithmic cost.

Theorem 2. *With probability at least* $1-\delta$ *with* $\delta < \frac{1}{2}$*, limiting the attack cost by* $O(\log(T))$*, there is no attack that can force the user to pick the worst arm more than* $T - O(T^{\alpha})$ *times, in which* $\alpha < \frac{9}{64K}$.

This theorem shows a contrast between the case where arm K, the target arm, is not the worst arm and the case where arm K is the worst arm. If arm K is not the worst arm, our scheme is able to force the user to pick the target arm $T - O(\log(T))$ times. On the other hand, if arm K is the worst, Theorem 2 shows that there is no attack strategy that can force the user to pick the worst arm more than $T - O(T^{\alpha})$ times.

4. NUMERICAL EXAMPLE

We now provide numerical examples to illustrate the analytical results obtained. In our simulation, the bandit has 10 arms. The rewards distribution of each arm i is $\mathcal{N}(\mu_i, \sigma)$. The mean rewards of all arms are $\mu_1, \ldots, \mu_K$ respectively.

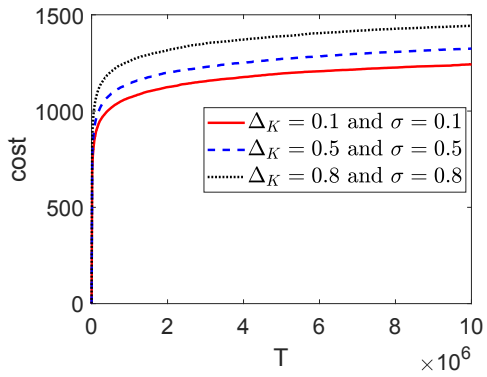

Fig. 3. Attack cost as $\sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j}$ varies

The attacker's target arm is K. We let $\delta = 0.05$. We then run the experiment for multiple trials and in each trial we run $T = 10^7$ rounds.

In Figure 1, we fix $\sigma = 0.1$ and $\Delta_K = 0.1$ and compare the number of rounds when target arm pulled with and without attack. In this experiment, the mean rewards of all arms are 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.1, and 0.2 respectively. Arm K is not the worst arm, but its average reward is lower than most arms. The results are averaged over 20 trials. The attacker successfully manipulates the user into pulling the target arm very frequently.

In Figure 2, in order to study how $\frac{\sigma}{\Delta_K}$ affects the attack cost, we fix $\Delta_K = 0.1$ and set σ as 0.1, 0.3 and 0.5 respectively. The mean rewards of all arms are same as above. From the figure, we can see that as $\frac{\sigma}{\Delta_K}$ increases, the attack cost increases. In addition, as predicted in our analysis, the attack cost increases over rounds in a logarithmic order.

Figure 3 illustrates how $\sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j}$ affects the attack cost. In this experiment, we fix $\frac{\sigma}{\Delta_K} = 1$ and set Δ_K as 0.2, 0.6 and 0.9 respectively. The mean rewards of all arms are the same as above. The figure illustrates that, as $\sum_{j \neq i^*} \frac{\Delta_K}{\Delta_j}$ increases, the attack cost also increases. This is consistent with our analysis in Corollary 1.

5. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a new class of attacks on stochastic bandits: action-manipulation attacks. We have analyzed the attack against on the UCB algorithm and proved that the proposed attack scheme can force the user to almost always pull a non-worst arm with only logarithm effort. Our theoretical results and numerical examples show a significant vulnerability of UCB algorithm under action-manipulation attacks. In the future, we will investigate action-manipulation attacks on other bandit algorithms such as ϵ-Greedy and contextual bandits etc. It is also of interest to investigate the defense strategy to mitigate the effects of this attack.

6. REFERENCES

[1] Olivier Chapelle, Eren Manavoglu, and Romer Rosales, "Simple and scalable response prediction for display advertising," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2014.

[2] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. of International Conference on World Wide Web*, New York, NY, Apr. 2010, pp. 661–670.

[3] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan, "Cascading bandits: Learning to rank in the cascade model," in *Proc. of International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, July 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 767–776.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[6] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *arXiv preprint arXiv:1703.06748*, 2017.

[7] Shike Mei and Xiaojin Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. of AAAI Conference on Artificial Intelligence*, Austin, TX, Jan. 2015, pp. 2871–2877.

[8] Battista Biggio, Blaine Nelson, and Pavel Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[9] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli, "Is feature selection secure against training data poisoning?," in *Proc. of International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, July 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 1689–1698.

[10] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in *Advances in Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 1885–1893.

[11] Scott Alfeld, Xiaojin Zhu, and Paul Barford, "Data poisoning attacks against autoregressive models," in *Proc. of AAAI Conference on Artificial Intelligence*, Phoenix, AZ, Feb. 2016, pp. 1452–1458.

[12] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu, "Adversarial attacks on stochastic bandits," in *Proc. of International Conference on Neural Information Processing Systems*, Montréal, Canada, Dec. 2018, pp. 3644–3653.

[13] Fang Liu and Ness Shroff, "Data poisoning attacks on stochastic bandits," in *Proc. of International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, CA, June 2019, vol. 97, pp. 4042–4050.

[14] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme, "Stochastic bandits robust to adversarial corruptions," in *Proc. of Annual ACM SIGACT Symposium on Theory of Computing*, Los Angeles, CA, June 2018, pp. 114–122.

[15] Sébastien Bubeck and Nicolo Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.