Adaptive Algorithms for Multi-armed Bandit with Composite and Anonymous Feedback

Siwei Wang[1], **Haoyun Wang**[2], **Longbo Huang**[1]
[1]Tsinghua University
[2]Georgia Institute of Technology
wangsw2020@mail.tsinghua.edu.cn, hwang800@gatech.edu, longbohuang@tsinghua.edu.cn

Abstract

We study the multi-armed bandit (MAB) problem with composite and anonymous feedback. In this model, the reward of pulling an arm spreads over a period of time (we call this period as reward interval) and the player receives partial rewards of the action, convoluted with rewards from pulling other arms, successively. Existing results on this model require prior knowledge about the reward interval size as an input to their algorithms. In this paper, we propose adaptive algorithms for both the stochastic and the adversarial cases, without requiring any prior information about the reward interval. For the stochastic case, we prove that our algorithm guarantees a regret that matches the lower bounds (in order). For the adversarial case, we propose the first algorithm to jointly handle non-oblivious adversary and unknown reward interval size. We also conduct simulations based on real-world dataset. The results show that our algorithms outperform existing benchmarks.

Introduction

The multi-armed bandit (MAB) model (Berry and Fristedt 1985; Sutton and Barto 1998) has found wide applications in Internet services, e.g., (Chen et al. 2018; Chapelle, Manavoglu, and Rosales 2015; Chen, Wang, and Yuan 2013; Jain and Jamieson 2018; Wang and Huang 2018), and attracts increasing attention. The classic MAB model can be described as a time-slotted game between the environment and a player. In each time slot, the player has N actions (or arms) to choose from. After he makes the selection, the player receives a reward from the chosen arm immediately. The rewards can be independent random variables generated from certain unknown distributions, known as the stochastic MAB problem (Lai and Robbins 1985), or arbitrarily chosen by the environment, called the adversarial MAB problem (Auer et al. 2002). In both models, the player's goal is to maximize his expected cumulative reward during the game by choosing arms properly. To evaluate the player's performance, the concept of "regret", defined as the expected gap between the player's total reward and offline optimal reward, is introduced as the evaluation metric.

Most of existing MAB algorithms, e.g., UCB (Gittins, Glazebrook, and Weber 2011; Auer, Cesa-Bianchi, and Fischer 2002) and EXP3 (Auer et al. 2002), are based on the fact that feedback for the arm-pulling action can be observed precisely and immediately. However, in many real-world applications, it is common that *arm rewards spread over an interval and are convoluted with each other.* As a concrete example, consider a company conducting advertisements via Internet. The effect of an ads, i.e., how it affects the number of clicks (reward), can often spread over the next few days after it is displayed. Specifically, in the next couple of days, there will be continuous clicks affected by the ad. Moreover, during this time, the company usually launches some other ads, which may also impact the number of clicks. As a result, the company only observes aggregated information on the reward (thus the feedback is also anonymous). This situation also happens in medical problems. For example, recent research found that the variability of blood glucose level is the key in controlling diabetes (Hirsch and Brownlee 2005). Yet, diabetes medicines do not cause sudden jumps on the blood glucose level. Instead, their effects last for a period, and the blood glucose level is often jointly affected by medicines taken within a period. These two features make it hard to separate effects of different medicines, as well as to estimate their effectiveness.

To address the above difficulties, in this paper, we consider an MAB model where the reward in each time slot is a positive vector. Specifically, the reward vector from pulling arm i at time t is $\boldsymbol{r}_i(t) = (r_{i,\tau}(t), \tau \geq 1)$, where $r_{i,\tau}(t)$ denotes the reward component in time $t+\tau$ from pulling arm i at time t. In addition, at time t, the player cannot observe each individual $r_{i,\tau}(t)$ directly. Instead, he observes the aggregated reward, i.e., $\sum_{\tau \geq 1} r_{a(\tau),t-\tau}(\tau)$, where $a(\tau)$ represents the chosen arm at time step τ.

Existing solution for this problem is to group time slots into rounds (Pike-Burke et al. 2018; Cesa-Bianchi, Gentile, and Mansour 2018), and choose to pull only *one* arm in each round. With a proper selection of the round size and other parameters, the problem can be connected to the non-anonymous setting (Neu et al. 2010; Joulani, Gyorgy, and Szepesvári 2013). However, these algorithms crucially rely on the precise knowledge of the reward interval size (or the delay of rewards). As a result, underestimating the interval size leads to no theoretical guarantees, whereas overestimation worsens the performances, since their regret bounds are positively related to this estimation. This makes the algorithms sensitive and less robust.

To deal with this challenge, in this paper we remove the requirement of any prior knowledge about the reward interval size. This is motivated by the fact that, in practical, e.g., medical applications, such information can be unknown or hard to obtain exactly. To solve the problem, we propose adaptive methods with increasing round sizes, to mitigate the influence of the reward spread and convolution and improve learning. Note that since we do not possess information about the reward interval size, it is critical and challenging to properly choose the speed for round size increase. Our analysis shows that, with a proper round size increasing rate (which does not depend on the knowledge of the reward interval size), our adaptive policies always possess theoretical guarantees on regrets in both the stochastic case and the adversarial case.

Our main contributions are summarized as follows:

1. We consider the stochastic MAB model with composite and anonymous feedback, where each arm's reward spreads over a period of time. Under this model, we propose the ARS-UCB algorithm, which requires *zero* a-prior knowledge about the reward interval size. We show that ARS-UCB achieves an $O(N\log T + c(d_1, d_2, N))$ regret, where $c(d_1, d_2, N)$ is a function that does not depend on T, and d_1 and d_2 are measures of the expectation and variance of the composite rewards, respectively. Our regret upper bound matches the regret lower bound for this problem, as well as regret bounds of existing policies that require knowing the exact reward interval size.
2. We propose the ARS-EXP3 algorithm for the adversarial MAB problem with composite and anonymous feedback studied in (Cesa-Bianchi, Gentile, and Mansour 2018). ARS-EXP3 does not require any knowledge about the reward interval size, and works in the case where the delays are non-oblivious. We show that ARS-EXP3 achieves an $O((d + (N\log N)^{\frac{1}{2}})T^{\frac{2}{3}})$ regret, where d is the size of the reward interval. To the best of our knowledge, ARS-EXP3 is the first efficient algorithm in this setting (i.e., the delays can be non-oblivious).
3. We conduct extensive experiments based on real-world datasets, to validate our theoretical findings. The results are consistent with our analysis, and show that our algorithms outperform state-of-the-art benchmarks. Thus, our adaptive policies are more robust and can be used more widely in real applications.

Related Works

Stochastic MAB with delayed feedback is first proposed in (Joulani, Gyorgy, and Szepesvári 2013; Agarwal and Duchi 2011; Desautels, Krause, and Burdick 2014). In (Joulani, Gyorgy, and Szepesvári 2013), the authors propose a BOLD framework to solve this problem. In this framework, the player only changes his decision when a feedback arrives. Then, decision making can be done the same as with non-delayed feedback. They show that the regret of BOLD can be upper bounded by $O(N(\log T + \mathbb{E}[d]))$, where d represents the random variable of delay. (Manegueu et al. 2020) then explored the case that the delay in each time slot is not i.i.d., but depends on the chosen arm. In this setting, they proposed the PatientBandits policy, which achieves near optimal regret upper bound. In addition to the stochastic case, adversarial MAB with delayed feedback also attracts people's attention. This model is first studied in (Weinberger and Ordentlich 2002), where it is assumed that the player has full feedback. The paper establishes a regret lower bound of $\Omega(\sqrt{(d+1)T\log N})$ for this model, where d is a constant feedback delay. The model with bandit feedback is investigated in (Neu et al. 2010, 2014), where the authors used the BOLD framework (Joulani, Gyorgy, and Szepesvári 2013) to obtain a regret upper bound of $O(\sqrt{(d+1)TN})$. Recently, (Zhou, Xu, and Blanchet 2019; Thune, Cesa-Bianchi, and Seldin 2019; Bistritz et al. 2019) made more optimizations on MAB with delayed feedback. Since their analytical methods are used in the non-anonymous setting, they are very different and cannot be used for our purpose.

(Pike-Burke et al. 2018) extends the model to contain anonymous feedback, and gives a learning policy called ODAAF. ODAAF uses information of the delay as inputs, including its mean and variance. This helps the algorithm to estimate the upper confidence bounds. The regret upper bound of ODAAF is $O(N(\log T + \mathbb{E}[d]))$, which is the same as BOLD with non-anonymous feedback. (Garg and Akash 2019) then explores the composite and anonymous feedback setting and makes some minor changes to generalize ODAAF policy. However, their algorithm still needs to use precise knowledge of the reward interval. As for regret lower bound, (Vernade, Cappé, and Perchet 2017) generalizes the regret lower bound of classic MAB model. They show that the stochastic MAB problem with delayed feedback still has a regret lower bound $O(N\log T)$. To the best of our knowledge, there is no known regret lower bound for the MAB model with delayed and anonymous feedback. We thus use $O(N\log T)$ as a regret lower bound in this model to compare our results with.

Inspired by the stochastic setting, (Cesa-Bianchi, Gentile, and Mansour 2018) studied the adversarial MAB model with composite and anonymous feedback, and present the CLW algorithm to solve the problem. In their paper, the losses (or the rewards) are assumed to be oblivious, so that the environment cannot change them during the game. They obtain a regret upper bound $O(\sqrt{dTN})$ for the CLW algorithm, and establish a matching $\Omega(\sqrt{dTN})$ regret lower bound.

Stochastic MAB with Composite and Anonymous Rewards

We start with the stochastic case and first introduce our model setting. Then, we present our Adaptive Round-Size UCB (ARS-UCB) algorithm and its regret upper bound with a proof sketch. The complete proofs are referred to the supplementary file (Wang, Wang, and Huang 2020).

Model Setting

We adapt the model setting in (Garg and Akash 2019), and allow the reward intervals to have infinite size. Specifically, in our setting, a player plays a game for T time slots. In each time slot, the player chooses one arm among a set of N arms $\mathcal{N} = \{1, \cdots, N\}$ to play. Each arm i, if played, generates an i.i.d. reward vector in $\mathbb{R}_+^\infty$, where $\mathbb{R}_+$ is the set of all non-negative real numbers.[1] We denote $\boldsymbol{r}_{a(t)}(t) = (r_{a(t),1}(t), r_{a(t),2}(t), \cdots)$ the reward vector generated by pulling arm $a(t) \in \mathcal{N}$ at time t, where the τ-th term $r_{a(t),\tau}(t)$ is the *partial* reward that the player obtains from arm $a(t)$ at time $t+\tau$ after pulling it at time t, and without loss of generality, we assume that $||\boldsymbol{r}_{a(t)}(t)||_1 \in [0, 1]$. We denote $D_{a(t)}$ the distribution of $\boldsymbol{r}_{a(t)}$ and $\boldsymbol{\mu}_{a(t)} \triangleq \mathbb{E}_{D_{a(t)}}[\boldsymbol{r}_{a(t)}]$ its mean. Then, at every time t, the player receives the *aggregated* reward from all previously pulled arms, i.e., $Y(t) \triangleq \sum_{\tau \le t-1} r_{a(\tau),t-\tau}(\tau)$.

Under this model, the expected total reward of pulling arm i is $s_i \triangleq ||\boldsymbol{\mu}_i||_1$. Without loss of generality, we assume $1 \ge s_1 > s_2 \ge \cdots \ge s_N \ge 0$, and denote $\Delta_i \triangleq s_1 - s_i$ for all $i \ge 2$ the reward gap of arm i. Then, the cumulative regret of the player can be expressed as $Reg(T) \triangleq Ts_1 - \mathbb{E}[\sum_{t=1}^T s_{a(t)}]$. The goal of the player is to find an algorithm to minimize his $Reg(T)$.

ARS-UCB Algorithm

To explain the idea of ARS-UCB (Algorithm 1), we first introduce some notations. We denote $N_i(t)$ the number of times the player chooses to pull arm i up to time t, and $M_i(t)$ the cumulative observed reward (w.r.t. $Y(t)$) up to t from pulling arm i, i.e., $N_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i]$ and $M_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i] Y(\tau)$. We also denote

$$\hat{s}_i(t) \triangleq \frac{M_i(t)}{N_i(t)}, \tag{1}$$

the empirical mean of arm i, and define an unknown reward $L_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i] ||\boldsymbol{r}_i(\tau)||_1$, which is the actual cumulative gain from arm i until time t (note that $L_i(t)$ is different from $M_i(t)$). If the $L_i(t)$ value for each arm i is known, then the origin UCB policy can be directly applied with empirical mean $\frac{L_i(t)}{N_i(t)}$'s, and achieve a regret $O(\sum_{i=2}^N \frac{1}{\Delta_i} \log T)$. However, since the player can only observe $M_i(t)$, in order to achieve a good performance, we want to ensure that the difference between $M_i(t)$ and $L_i(t)$ is small. More precisely, as t goes to infinity, we want the difference between $\frac{M_i(t)}{N_i(t)}$ and $\frac{L_i(t)}{N_i(t)}$ to converge to 0.

An intuitive approach to achieve this is to choose an increasing function $f : \mathbb{N}_+ \to \mathbb{N}_+$, where $\mathbb{N}_+$ is the set of all positive integers, and to use $f(k)$ as the number of time steps in the k-th round. Then, in each round, we only pull a single arm. Figure 1 shows the difference between $M_i(t)$ and $L_i(t)$ in each round. In this figure, the rewards in the blue rectangle are the feedback in $M_i(t)$, and the rewards in the red parallelogram are those in $L_i(t)$. We see that no matter how long a round is, the difference is always bounded by the two triangle parts. Let $K_i(t)$ be the value of K_i in Algorithm 1 until time t, and denote $F(K) \triangleq \sum_{k=1}^K f(k)$. Then, at the decision time slot in Algorithm 1, $N_i(t) = F(K_i(t))$ for each arm i. On the other hand, Figure 1 shows that $|M_i(t) - L_i(t)| = O(K_i(t))$ for each arm i. If f is increasing, $K/F(K)$ will converge to 0 as K increases. Therefore, $|\frac{M_i(t)}{N_i(t)} - \frac{L_i(t)}{N_i(t)}| = O(\frac{K_i(t)}{F(K_i(t))}) \to 0$ as t goes to infinity. As a result, the algorithm behaves like UCB after some time.

Algorithm 1 Adaptive Round-Size UCB (ARS-UCB)

```
Input: $f, \alpha$.
For each arm $i$, play it for $f(1)$ times and set $K_i = 2$.
while $t < T$ do
    For all arm $i$, $u_i(t) = \min\{\hat{s}_i(t) + \sqrt{\frac{\alpha \log t}{N_i(t)}}, 1\}$, where $\hat{s}_i(t)$ is defined in Eq. (1).
    Play arm $a(t) \in \text{argmax}_i\, u_i(t)$ for $f(K_{a(t)})$ times (if there are multiple maximum $u_i(t)$, choose the arm with smallest $N_i(t)$).
    $K_{a(t)} = K_{a(t)} + 1$.
end while
```

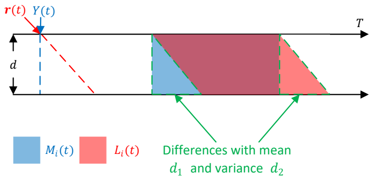


Figure 1: The difference between $M_i(t)$ and $L_i(t)$ in each round. It can be bounded by the areas of the two triangles.

From the above reasoning, we see that the input f in Algorithm 1 is introduced to control the convergence rate of $\frac{M_i(t) - L_i(t)}{N_i(t)}$. The other input α, used in the confidence radius, is to control the change of likelihood of the event $\{s_i \le u_i(t), \forall i\}$. Carefully choosing f and α is the key to ensure a good performance of the algorithm.

Theorem 1. *Suppose $\alpha > 4$ and the function f satisfies (i) f is increasing, and (ii) $\exists k_0$ such that $\forall k > k_0, F(k) \ge f(k+1)$. Then, ARS-UCB achieves that*

$$Reg(T) \le \sum_{i=2}^N \frac{8\alpha \log T}{\Delta_i} + c_f^*(d_1, d_2, N, \alpha). \tag{2}$$

Here $c_f^*(d_1, d_2, N, \alpha)$ is a constant that does not depend on T,[2] *$d_1 \triangleq \sum_{d'=1}^\infty \max_i \mathbb{E}[\sum_{\tau=d'}^\infty r_{i,\tau}]$, $d_2 \triangleq \sum_{d'=1}^\infty \max_i \mathrm{Var}[\sum_{\tau=d'}^\infty r_{i,\tau}]$, where $r_{i,\tau}$ is the τ-th term of random vector $\boldsymbol{r}_i$, and the expectation and variance are taken over the distribution D_i.*

Notice that it is not hard to find such a function f. For example, $f(k) = ck^\beta$ satisfies the two properties with integers $c \ge 1$ and $\beta \ge 1$. Another example is $f(k) = 2^{k+c}$ with $c \ge 0$ (here we need to set $f(1) = 2^{2+c}$ specifically). The value d_1 in the theorem can be regarded as an upper bound

[1] We allow the effects of pulling an arm to last forever, while prior works, e.g., (Garg and Akash 2019), all assume a finite reward interval size.

[2] The term $c_f^*(d_1, d_2, N, \alpha)$ does not depend on T as long as the function f does not depend on T, e.g., $f(k) = k^2$ or $f(k) = k^3$.

of the expected rewards in the triangle region (in Figure 1), and d_2 is an upper bound of the variance. Compared to the ODAAF policy in (Pike-Burke et al. 2018; Garg and Akash 2019), the regret upper bound of ARS-UCB also depends on the mean and variance of the feedback delay. However, our algorithm has the advantage that it does not require any prior information about d_1 and d_2, whereas the ODAAF policy takes both d_1 and d_2 as inputs. Thus, ARS-UCB can be applied to settings where such information is not available.

Another advantage of ARS-UCB is that the constant factor before the $\log T$ term in its regret upper bound is much smaller than ODAAF. This is because that ODAAF follows an elimination structure, and only eliminates a sub-optimal arm when its upper confidence bound is smaller than the *lower* confidence bound of the optimal arm. On the other hand, ARS-UCB follows the basic UCB structure, in which the player always chooses the arm with largest upper confidence bound. Since in each time step, there are only tiny changes on upper confidence bounds when t is large, the upper confidence bounds of sub-optimal arms are approximately equal to the *upper* confidence bound of the optimal arm in the end of the game. Therefore, one needs to pull each sub-optimal arm more in ODAAF to obtain a smaller upper confidence bound (to match the *lower* confidence bound of the optimal arm rather than the *upper* confidence bound). As a result, a larger regret upper bound occurs. This fact is also supported by our simulation results, i.e., ARS-UCB always outperforms ODAAF.

Lastly, although ARS-UCB chooses a same arm in each round, doing so does not cause excessive regret compared to UCB, as ARS-UCB can be viewed as grouping the plays of arms into consecutive intervals. This is also validated in our regret analysis and simulation results.

Remark 1. *The performance guarantees for ARS-UCB hold for any constant $\alpha > 4$ and increasing function f. However, to avoid a large constant in regret, choosing a function f that increases faster would be better, e.g., $f(k) = k^2$. As for α, when the delay measures d_1, d_2 are large and T is small, a larger α can reduce the regret. On the other hand, when d_1, d_2 are small but T is large, a smaller α behaves better.*

Proof Sketch of Theorem 1. Note that in classic UCB policy, $s_i \le v_i(t) \triangleq \frac{L_i(t)}{N_i(t)} + \sqrt{\frac{4\log t}{N_i(t)}}$ with high probability. Thus we want to ensure that for large enough t, we have $u_i(t) \ge v_i(t)$, i.e., $\frac{L_i(t)-M_i(t)}{N_i(t)} \le (\sqrt{\alpha}-2)\sqrt{\frac{\log t}{N_i(t)}}$ (or equivalently, $\frac{L_i(t)-M_i(t)}{\sqrt{N_i(t)}} \le (\sqrt{\alpha}-2)\sqrt{\log t}$). If this inequality holds, we know that $s_i \le u_i(t)$ with high probability.

As described in Figure 1, in a round $[t_1, t_2]$ such that $a(t) = i$ for all $t \in [t_1, t_2]$, the gap between $\sum_{t=t_1}^{t_2} Y(t)$ and $\sum_{t=t_1}^{t_2} ||\boldsymbol{r}_{a(t)}(t)||$ are the two triangle terms, i.e.,

$$\sum_{t=t_1}^{t_2} Y(t) = \sum_{t=t_1}^{t_2} ||\boldsymbol{r}_{a(t)}(t)|| + \sum_{t\le t_1-1}\sum_{\tau=t_1-t}^{\infty} r_{a(t),\tau}(t) - \sum_{t\le t_2}\sum_{\tau=t_2-t+1}^{\infty} r_{a(t),\tau}(t). \qquad (3)$$

Summing over the rounds we choose arm i, the gap between $M_i(t)$ and $L_i(t)$ is $\Theta(K_i(t))$. If $f(k)$ is increasing, we know that $\frac{K_i(t)}{\sqrt{N_i(t)}} \le \sqrt{2}$. Hence, there must be some time step $T^* = c_f^*(d_1, d_2, N, \alpha)$ such that $\frac{L_i(t)-M_i(t)}{N_i(t)} = \Theta(\frac{K_i(t)}{N_i(t)}) \le (\sqrt{\alpha}-2)\sqrt{\frac{\log t}{N_i(t)}}$ for any $t > T^*$. This means that ARS-UCB is efficient after time step T^*, which results in the regret upper bound in Eq. (2). □

Non-oblivious Adversarial MAB with Composite and Anonymous Rewards

We first introduce the adversarial model setting. Then, we present our Adaptive Round-Size EXP3 (ARS-EXP3) algorithm for the non-oblivious case and state its regret upper bound. Similarly, a proof sketch is provided, and the complete proofs are referred to the supplementary file (Wang, Wang, and Huang 2020).

Model Setting

In the adversarial MAB model with composite and anonymous feedback, there are N arms $\mathcal{N} = \{1, 2, \cdots, N\}$ and the game lasts for T time steps. In each time slot t, the adversary gives every arm i a reward vector $\boldsymbol{r}_i(t) \in \mathbb{R}_+^d$ where d is some unknown constant. To normalize the reward, we assume that $||\boldsymbol{r}_i(t)||_1 \le 1$. At any time slot t, if the player chooses to pull arm i, he receives reward $r_{i,\tau}(t)$ at time slot $t+\tau$. Similar to the stochastic scenario, in every time slot t, the player receives an *aggregated* reward $Z(t) \triangleq \sum_{\tau=t-d}^{t-1} r_{a(\tau),t-\tau}(\tau)$, where $a(t)$ represents the chosen arm at time t, and $r_{a(t),t-\tau}(\tau)$ is the $(t-\tau)$-th partial reward in $\boldsymbol{r}_{a(\tau)}(\tau)$. Denote $G_i \triangleq \sum_{t=1}^{T} ||\boldsymbol{r}_i(t)||_1$. The total regret of the player is defined as $Reg(T) \triangleq \mathbb{E}[\max_i G_i] - \mathbb{E}[\sum_{t=1}^{T} ||\boldsymbol{r}_{a(t)}(t)||_1]$. In the following, we also assume for simplicity that T is known to the player.[3]

Note that although our model is the same as the one in (Cesa-Bianchi, Gentile, and Mansour 2018), we allow the delay to be non-oblivious. Specifically, for any arm i and time step t, the actual received reward $s_i(t)$ is predetermined (i.e., the actual rewards are oblivious). However, the adversary can choose an arbitrary reward vector $\boldsymbol{r}_i(t) \in \mathbb{R}_+^d$ based on previous observations, as long as $||\boldsymbol{r}_i(t)||_1 = s_i(t)$ (i.e., how the reward spreads over time are non-oblivious). As a result, prior works cannot be applied and it requires new algorithms and analysis.

ARS-EXP3 Algorithm

Our algorithm will similarly use an increasing round size. Given a round size function $g : \mathbb{N}_+ \to \mathbb{N}_+$, the round sizes of the game are set to be $g(1), g(2), \cdots$ Since T is known, we can first compute K, the number of all completed rounds during the game, and use $g(K)$ as a normalization factor.

The algorithm for this adversarial setting is called ARS-EXP3, which is shown in Algorithm 2. In the algorithm,

[3] If T is unknown, one can use the doubling-trick method, e.g., in (Lu, Pál, and Pál 2010; Slivkins 2014).

Algorithm 2 Adaptive Round-Size EXP3 (ARS-EXP3)

```
Input: $g, \gamma, T, w_1 = \cdots = w_N = 1$.
Compute the last round number $K$.
for $k = 1, 2, \cdots, K$ do
    For any $i$, $e_i = \exp(\frac{w_i}{g(K)})$, $p_i = (1-\gamma)\frac{e_i}{\sum_i e_i} + \frac{\gamma}{N}$.
    Draw $a(k) \sim p$, and then pull arm $a(k)$ in round $k$ (with size $g(k)$). Let $Z(k)$ be the collected rewards within this round $k$.
    $Z'(k) = \min\{Z(k), g(k)\}$, $w_{a(k)} = w_{a(k)} + \frac{\gamma Z'(k)}{N p_{a(k)}}$.
end for
```

the notations $N_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i]$, $M_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i] Z(t)$ and $L_i(t) \triangleq \sum_{\tau \le t} \mathbb{I}[a(\tau) = i] ||\boldsymbol{r}_i(t)||_1$ remain the same as in the stochastic case. In the classic EXP3 policy, the probability of choosing arm i depends on $L_i(t)$ but not $\frac{L_i(t)}{N_i(t)}$. Since the observed values are the $M_i(t)$'s, we need a bound on $|L_i(t) - M_i(t)|$. Yet, we cannot expect $|L_i(t) - M_i(t)|$ to converge to 0 because it can only increase during the game. This means that we cannot use the same analysis as in the stochastic case.

Our analysis is based on the following observation: the regret of using the classic EXP3 policy under our setting can be upper bounded by the regret of using classic EXP3 policy under the classic adversarial MAB model, plus the largest difference $\max_i |L_i(T) - M_i(T)|$. The reason is that if we pretend to actually receive reward of $M_i(T)$ from arm i, then the regret will be the same as that in the classic model. However, in our model, we receive $L_i(t)$. Thus, our regret upper bound should include the difference term $\max_i |L_i(T) - M_i(T)|$, which depends on the number of rounds in the game. Hence, we choose an increasing function $g(\cdot)$, to ensure that the algorithm only runs $o(T)$ rounds, so that the regret is sub-linear.

Theorem 2. *Set* $\gamma = \min\{1, \sqrt{\frac{N \log N}{(e-1)((\beta+1)T)^{\frac{1}{\beta+1}}}}\}$ *and* $g(k) = k^\beta$. *Then, Algorithm 2 achieves*

$$Reg(T) = O((N \log N)^{\frac{1}{2}} T^{\frac{2\beta+1}{2\beta+2}} + dT^{\frac{1}{\beta+1}}).$$

In particular, if $\beta = \frac{1}{2}$, *Algorithm 2 achieves a regret upper bound* $O((d + (N \log N)^{\frac{1}{2}})T^{\frac{2}{3}})$.

Remark 2. *Note that the analysis for this case is very different from that in the stochastic case. In the stochastic case, as long as the error probability is smaller than* $\frac{1}{t^3}$, *the round size does not influence the cumulative regret. In the adversarial case, however, the regret is linear in the largest round size. Thus, we need a lower increasing speed. Theorem 2 shows that* $g(k) = k^{\frac{1}{2}}$ *provides a good choice.*

Proof Sketch of Theorem 2. Let K be the last completed round until time T. Then, there are less than $g(K+1)$ slots left, which can cause at most $g(K+1)$ additional regret.

Define $G(K) \triangleq \sum_{k=1}^{K} g(k)$, and consider another game lasting for $G(K)$ time steps, where pulling arm i at time t gives reward $R_i(t) = Z(t) = \sum_{\tau=t-d}^{t-1} r_{a(\tau),t-\tau}(\tau)$. In this game, Algorithm 2 behaves the same as an EXP3 algorithm running K time steps with largest reward $g(K)$ in each step. These imply an $O(g(K)\sqrt{NK \log N})$ regret upper bound (Auer et al. 2002). Since the cumulative rewards of these two games are the same, the remaining part is the difference between the total rewards of their best arms.

Similar to the stochastic case, for a round $[t_1, t_2]$ such that $a(t) = i$ for all $t \in [t_1, t_2]$, the following equation (4) holds.

$$\sum_{t=t_1}^{t_2} Z(t) = \sum_{t=t_1}^{t_2} ||\boldsymbol{r}_{a(t)}(t)|| + \sum_{t=t_1-d}^{t_1-1} \sum_{\tau=t_1-t}^{d} r_{a(t),\tau}(t) - \sum_{t=t_2-d+1}^{t_2} \sum_{\tau=t_2-t+1}^{d} r_{a(t),\tau}(t). \tag{4}$$

This implies that during one round, the difference on the reward of any single arm between the two games can increase by at most $\sum_{t=t_1-d}^{t_1-1} \sum_{\tau=t_1-t}^{d} r_{a(t),\tau}(t)$, which is less than or equal to d. Then, since there are totally K rounds, Algorithm 2 can have an additional regret Kd.

Combining the three components, we obtain $Reg(T) = O(g(K)\sqrt{NK \log N} + Kd + g(K+1))$.

When we set $g(k) = k^\beta$, then $K = \Theta(T^{\frac{1}{\beta+1}})$. Thus, the cumulative regret satisfies that $Reg(T) = O((N \log N)^{\frac{1}{2}} T^{\frac{2\beta+1}{2\beta+2}} + dT^{\frac{1}{\beta+1}})$. □

The adversary can choose the reward vectors properly to make sure that every switch between arms causes a constant bias between $M_i(t)$ and $L_i(t)$. This bias makes our observations inaccurate, and is then added to the final regret (the Kd term) according to our analysis. Because of this, our model setting is similar to the non-oblivious adversarial MAB model with switching cost, in which each switch leads to an additional cost. (Cesa-Bianchi, Dekel, and Shamir 2013; Dekel et al. 2014) show that the non-oblivious adversarial MAB with switching cost has a regret lower bound of $\Omega(T^{\frac{2}{3}})$. Therefore, it is reasonable that we can only obtain a similar $O(T^{\frac{2}{3}})$ regret upper bound.

Simulations

The Stochastic Setting

We start with the stochastic case. In our experiments, there are a total of 9 arms. The expected reward of the 9 arms follows the vector $\boldsymbol{s} = [.9, .8, .7, .6, .5, .4, .3, .2, .1]$. We conduct experiments on the following cases.

Random delay In this case, the reward of pulling an arm is given to the player after a random delay z. That is, $\forall \tau' \neq z, r_{a(t),\tau'}(t) = 0$ and $\mathbb{E}[r_{a(t),z}(t)] = s_{a(t)}$. We choose z to be i.i.d. uniformly in $[10, 30]$ (Figure 2(a)) or $[0, 60]$ (Figure 2(b)). For comparison, we choose the ODAAF algorithm proposed in (Pike-Burke et al. 2018) with accurate knowledge about the delay z as benchmark.

Bounded interval In this case, the reward of pulling an arm at time t takes effect in time interval $[t + d_{\min}, t + d_{\max})$ and the effects within this period remains the same. That is, $\mathbb{E}[r_{a(t),d_{\min}}(t)] = \frac{s_{a(t)}}{d_{\max} - d_{\min}}$, and $\forall \tau \in [t + d_{\min}, t +$

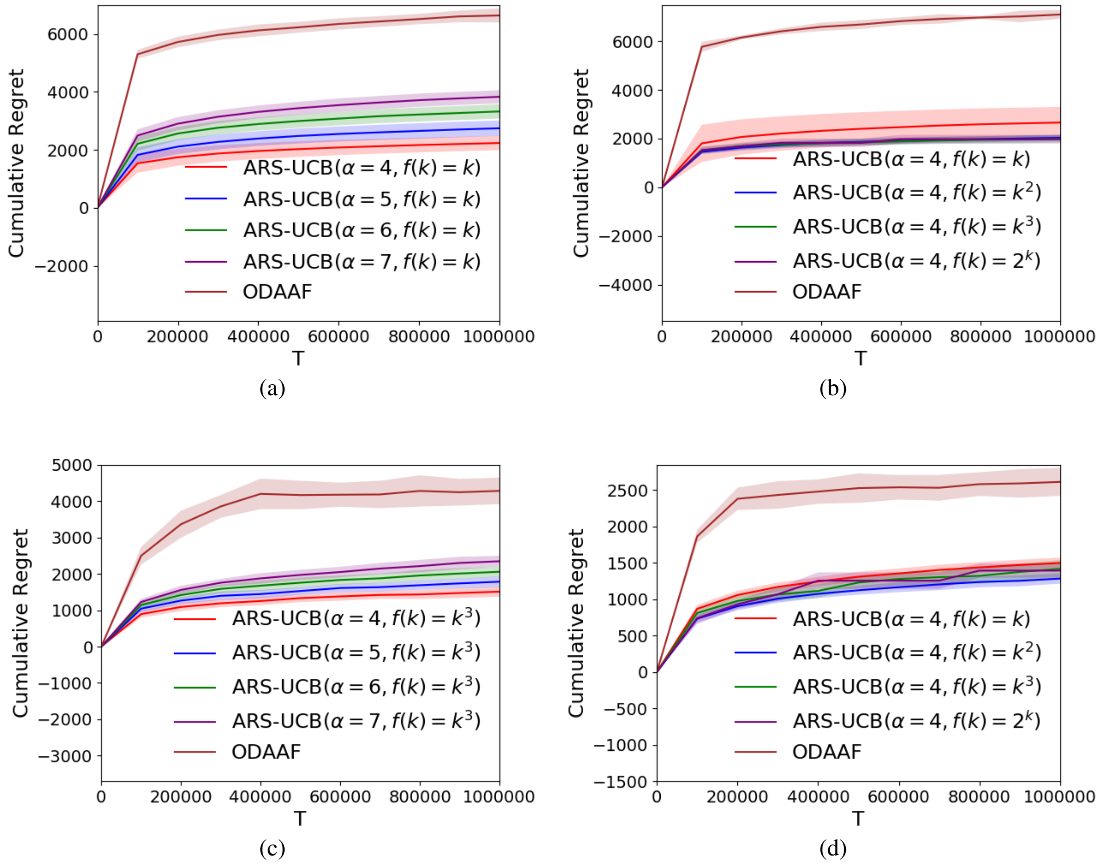


Figure 2: Experiments: Comparison between cumulative regrets of ARS-UCB and ODAAF

$d_{\max}), r_{a(t),\tau}(t) = r_{a(t),d_{\min}}(t)$ (the reward vectors are of the form $[0,\cdots,0,r,\cdots,r,0,\cdots]$). In Figure 2(c) we choose $(d_{\min}, d_{\max}) = (30, 40)$, and in Figure 2(d) we choose $(d_{\min}, d_{\max}) = (0, 20)$. For comparison, we choose the generalized ODAAF algorithm proposed in (Garg and Akash 2019) with accurate knowledge about the reward interval size $d_{\max}$ as benchmark.

Discounted reward In this case, the reward of pulling an arm at time t takes effect from time $t+1$ and lasts forever. Moreover, its value decreases exponentially with a factor $\gamma \in (0,1)$. That is, $\mathbb{E}[r_{a(t),1}(t)] = (1-\gamma)s_{a(t)}$, and $\forall \tau > 1, r_{a(t),\tau}(t) = \gamma r_{a(t),\tau-1}(t)$ (the reward vectors are of the form $[r, \gamma r, \gamma^2 r, \cdots]$). In Figure 3(a) we choose $\gamma = 0.8$, and in Figure 3(b) we choose $\gamma = 0.9$. In this case the reward interval size is infinity, and there is no existing benchmarks. Therefore, we only compare the cumulative regrets of ARS-UCB with different parameters.

Conclusion on experimental results In the above experiments, we observe that the cumulative regrets of ARS-UCB are always logarithmic in T, which is expected from our theoretical analysis. Moreover, in all our experiments, ARS-UCB significantly outperforms ODAAF, which assumes full knowledge of the delay size. This is because that in ODAAF the player needs to pull the sub-optimal arms more to eliminate them, which leads to a worse performance than ARS-UCB. Therefore, ARS-UCB is more robust and more efficient in the case of minimizing the cumulative regret.

In Figures 2(a), 2(c) and 3(a) and we choose the same function of $f(k)$ and vary the value of α, while Figures 2(b), 2(d) and 3(b) show the performance under different $f(k)$ functions with a same value $\alpha = 4$ (other α values show similar behavior). From these results, we can see that the combination of $\alpha = 4$ and $f(k) = k^2$ behaves better in most of these problem instances, and leads to both small regrets and small variances.

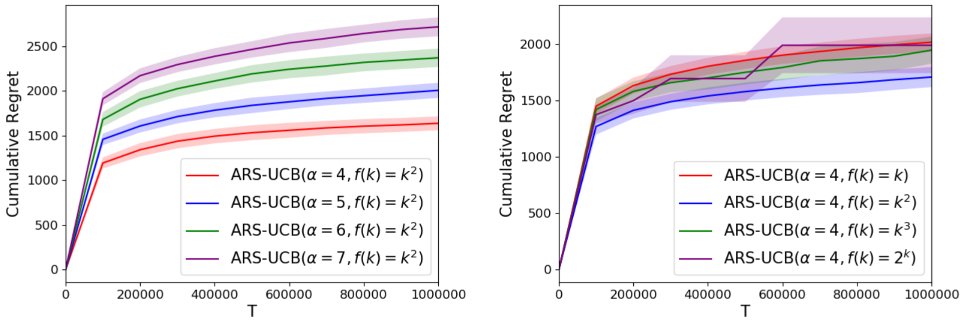
The Adversarial Setting

Here we use two datasets in Kaggle: the *Outbrain Click Prediction* (Outbrain) dataset[4], and the *Coupon Purchase Prediction* (Coupon) dataset[5].

The Outbrain dataset records whether users click a provided advertisement when they enter the system. In this experiment, the system needs to decide the category of adver-

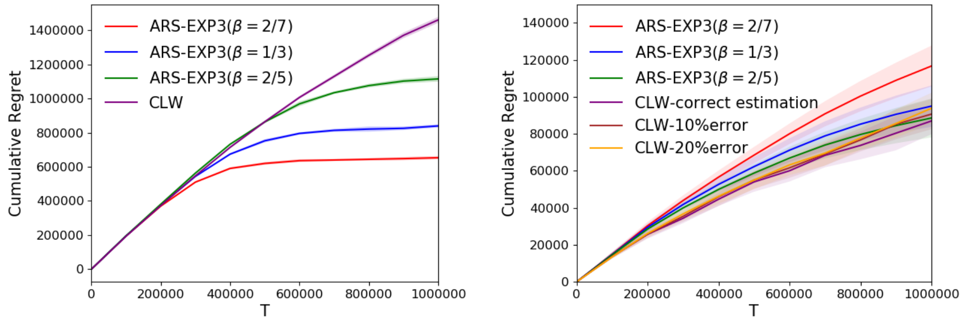
[4] https://www.kaggle.com/c/outbrain-click-prediction

[5] https://www.kaggle.com/c/coupon-purchase-prediction



(a) (b)

Figure 3: Experiments: The cumulative regrets of ARS-UCB



(a) (b)

Figure 4: Experiments: Comparison between cumulative regrets of ARS-EXP3 and CLW

tisements to show to an incoming user, and his goal is to maximize the number of users clicking the advertisement. In this dataset, a user only clicks *one* category of the advertisements, and the reward (feedback) is whether the advertisement is clicked. The Coupon dataset records whether users click offered coupons. In this case, a system is providing coupons to its users, and it needs to decide what coupon to offer on each day of the week. The goal of the system is to maximize the number of users that click the coupons. Thus, the reward (and the feedback) of one time slot is whether the coupon is clicked. In our setting, the feedback is given to the player after a delay z, which is artificially simulated.

In the experiments of Outbrain dataset, we consider non-oblivious delays with $z \leq d = 10$ in Figure 4(a). Here the adversary choose delay $z = d$ only if the chosen arm is the best one and it has been chosen for at least $3d$ times in succession, otherwise it set $z = 1$. We then use Coupon dataset to simulate the oblivious setting, in Figure 4(b), we set $d = 20$ and choose z uniformly in $[10, 20]$.

From the experimental results, we can see that: i) in the non-oblivious setting, the CLW policy (Cesa-Bianchi, Gentile, and Mansour 2018) suffers from a linear regret, while our ARS-EXP3 policy achieves a sub-linear regret; ii) in the oblivious setting, CLW performs only slightly better than ARS-EXP3 even with the correct delay estimation d, and it would perform worse than ARS-EXP3 when the estimation on d has a tiny error, e.g., 10%, The experimental results show that in the oblivious setting, ARS-EXP3 policy is more robust, especially when there is no accurate information about the delay d. As for the non-oblivious delay case, ARS-EXP3 is the only existing efficient learning policy.

Conclusion

In this paper, we consider the MAB problem with composite and anonymous feedback, both the stochastic and adversarial settings. For the former case, we propose the ARS-UCB algorithm, and for the latter case, we design the ARS-EXP3 algorithm. These algorithms require *zero* knowledge about the feedback delay. We establish theoretical regret upper bounds for the algorithms, and then use experiments to show that our algorithms outperform existing benchmarks.

Our future research includes deriving a matching regret lower bound for the non-oblivious adversarial case. In (Cesa-Bianchi, Gentile, and Mansour 2018), the authors also provide a similar policy for bandit convex optimization (BCO) with delayed and anonymous feedback. How to adapt our framework and obtain tight regret upper bounds for BCO is another interesting future research problem.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China Grant 61672316, the Zhongguancun Haihua Institute for Frontier Information Technology and the Turing AI Institute of Nanjing.

References

Agarwal, A.; and Duchi, J. C. 2011. Distributed delayed stochastic optimization. In *Neural Information Processing Systems*, 873–881.

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

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The Non-Stochastic Multi-Armed Bandit Problem. *Siam Journal on Computing* 32(1): 48–77.

Berry, D. A.; and Fristedt, B. 1985. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer.

Bistritz, I.; Zhou, Z.; Chen, X.; Bambos, N.; and Blanchet, J. 2019. Online exp3 learning in adversarial bandits with delayed feedback. In *Neural Information Processing Systems*, 11345–11354.

Cesa-Bianchi, N.; Dekel, O.; and Shamir, O. 2013. Online learning with switching costs and other adaptive adversaries. In *Neural Information Processing Systems*, 1160–1168.

Cesa-Bianchi, N.; Gentile, C.; and Mansour, Y. 2018. Nonstochastic bandits with composite anonymous feedback. In *Conference On Learning Theory*, 750–773.

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

Chen, K.; Cai, K.; Huang, L.; and Lui, J. C. 2018. Beyond the click-through rate: web link selection with multi-level feedback. In *International Joint Conference on Artificial Intelligence*, 3308–3314.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, 151–159.

Dekel, O.; Ding, J.; Koren, T.; and Peres, Y. 2014. Bandits with switching costs: $T^{2/3}$ regret. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 459–467.

Desautels, T.; Krause, A.; and Burdick, J. W. 2014. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research* 15: 3873–3923.

Garg, S.; and Akash, A. K. 2019. Stochastic bandits with delayed composite anonymous feedback. *arXiv preprint arXiv:1910.01161* .

Gittins, J.; Glazebrook, K.; and Weber, R. 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.

Hirsch, I. B.; and Brownlee, M. 2005. Should minimal blood glucose variability become the gold standard of glycemic control? *Journal of Diabetes and Its Complications* 19(3): 178–181.

Jain, L.; and Jamieson, K. 2018. Firing bandits: Optimizing crowdfunding. In *International Conference on Machine Learning*, 2211–2219.

Joulani, P.; Gyorgy, A.; and Szepesvári, C. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*, 1453–1461.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1): 4–22.

Lu, T.; Pál, D.; and Pál, M. 2010. Contextual multi-armed bandits. In *International conference on Artificial Intelligence and Statistics*, 485–492.

Manegueu, A. G.; Vernade, C.; Carpentier, A.; and Valko, M. 2020. Stochastic bandits with arm-dependent delays. In *International Conference on Machine Learning*.

Neu, G.; Antos, A.; György, A.; and Szepesvári, C. 2010. Online Markov decision processes under bandit feedback. In *Neural Information Processing Systems*, 1804–1812.

Neu, G.; Gyorgy, A.; Szepesvari, C.; and Antos, A. 2014. Online Markov Decision Processes Under Bandit Feedback. *IEEE Transactions on Automatic Control* 59(3): 676–691.

Pike-Burke, C.; Agrawal, S.; Szepesvari, C.; and Grunewalder, S. 2018. Bandits with Delayed, Aggregated Anonymous Feedback. In *International Conference on Machine Learning*, 4105–4113.

Slivkins, A. 2014. Contextual bandits with similarity information. *The Journal of Machine Learning Research* 15(1): 2533–2568.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Thune, T. S.; Cesa-Bianchi, N.; and Seldin, Y. 2019. Nonstochastic Multi-armed Bandits with Unrestricted Delays. In *Neural Information Processing Systems*.

Vernade, C.; Cappé, O.; and Perchet, V. 2017. Stochastic Bandit Models for Delayed Conversions. In *Conference on Uncertainty in Artificial Intelligence*.

Wang, S.; and Huang, L. 2018. Multi-armed bandits with compensation. In *Neural Information Processing Systems*, 5114–5122.

Wang, S.; Wang, H.; and Huang, L. 2020. Adaptive Algorithms for Multi-armed Bandit with Composite and Anonymous Feedback. *arXiv preprint arXiv:2012.07048* .

Weinberger, M. J.; and Ordentlich, E. 2002. On delayed prediction of individual sequences. *international symposium on information theory* 48(7): 1959–1976.

Zhou, Z.; Xu, R.; and Blanchet, J. 2019. Learning in generalized linear contextual bandits with stochastic delays. In *Neural Information Processing Systems*, 5198–5209.