Nearly Optimal Adaptive Procedure with Change Detection for Piecewise-Stationary Bandit

Yang Cao
Uber Technologies Inc.

Zheng Wen
Adobe Research

Branislav Kveton
Adobe Research

Yao Xie
Georgia Tech

Abstract

Multi-armed bandit (MAB) is a class of online learning problems where a learning agent aims to maximize its expected cumulative reward while repeatedly selecting to pull arms with unknown reward distributions. We consider a scenario where the reward distributions may change in a piecewise-stationary fashion at unknown time steps. We show that by incorporating a simple change-detection component with classic UCB algorithms to detect and adapt to changes, our so-called M-UCB algorithm can achieve nearly optimal regret bound on the order of $O(\sqrt{MKT \log T})$, where T is the number of time steps, K is the number of arms, and M is the number of stationary segments. Comparison with the best available lower bound shows that our M-UCB is nearly optimal in T up to a logarithmic factor. We also compare M-UCB with the state-of-the-art algorithms in numerical experiments using a public Yahoo! dataset and a real-world digital marketing dataset to demonstrate its superior performance.

1 Introduction

Multi-armed bandit (MAB) is a class of fundamental problems in online learning and sequential decision making, where at each step a learning agent adaptively selects to pull one arm of a K-arm bandit based on its past observations, and receives one reward accordingly. The learning agent's objective is to maximize its expected cumulative reward in the first T time steps. MAB has found an extensive list of applications including communication systems [Thompson, 1933, Alaya-Feki et al., 2008], clinical trials [Vermorel and Mohri, 2005, Villar et al., 2015], online recommendation systems [Li et al., 2011, Bouneffouf et al., 2012, Kveton et al., 2014], and online advertisement campaign [Girgin et al., 2012, Schwartz et al., 2017].

Most existing literature on MAB problems focuses on two types of models: (i) the stochastic bandit model [Lai and Robbins, 1985, Auer et al., 2002a], where each of the K arms has a time-invariant reward distribution, and (ii) the adversarial bandit model [Littlestone and Warmuth, 1994, Auer et al., 2002b], where the reward distribution of each arm may change adversarially at all the time steps. However, in many real-world applications, neither of the above two models is realistic. Specifically, in such applications, the arms' reward distributions do vary with time, but much less frequently than what the adversarial bandit model assumes. For instance, in recommender systems, each item is modeled as an arm and users' clicks are modeled as rewards. In practice, a user's click probability on an item is unlikely to be time-invariant, or change significantly at all the time steps. Thus, for this case, it is too ideal to assume the stochastic bandit model and too conservative to assume the adversarial bandit model. Similar situations arise in dynamic pricing systems and investment options selection [Yu and Mannor, 2009, Cesa-Bianchi and Lugosi, 2006]. Motivated by this, we examine a scenario that lies "in between" the above two standard models, namely, the piecewise-stationary bandit that we describe below. The piecewise-stationary reward functions can also be viewed as an approximation to the slowly time-varying reward functions.

In this paper, we consider a class of non-stationary bandit problems, where the reward distribution of each arm is piecewise-constant and shifts at some unknown time steps called the *change-points*. This setting has been considered in prior works [Hartland et al., 2007, Garivier and Moulines, 2008, Yu and Mannor, 2009] as a more realistic scenario to model the users' preferences and in [Auer, 2002] to model an adversarial setting. We propose a simple but efficient algorithm called Monitored-UCB (M-UCB) by incorporating a change-point detection component into a classic Upper Confidence Bound (UCB) algorithm. M-UCB monitors the estimated mean of the reward dis-

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

tribution for the currently selected arm; once a change is detected, M-UCB algorithm will reset and learn the new optimal arm.

We show that, somewhat surprisingly, this simple M-UCB algorithm is nearly optimal for the considered scenario, in the sense that it achieves an $O(\sqrt{MKT\log T})$ regret bound under mild technical assumptions (see Section 5), where T is the number of time steps, K is the number of arms, and M is the number of stationary segments. This regret bound matches the $\Omega(\sqrt{T})$ lower bound proven in [Garivier and Moulines, 2011] up to a logarithmic factor. In practice, M-UCB is also robust, since it requires minimum parameter specification: we do not need to specify the pre- and post-change detection as the classic CUSUM procedure does [Liu et al., 2017]; the change detection is achieved by a simple two sample test for the running sample means over a sliding window. This result conveys a message that simple (rather than more sophisticated) change-point detection might suffice for piecewise stationary bandit. To the best of our knowledge, M-UCB is the first practical algorithm for piecewise-stationary multi-armed bandits that uses change-point detection and whose near optimality is proved without strong parametric assumptions.

In additional, we validate numerically the scalings of the M-UCB's regret in M and K. Experiment results in Section 6.1 show that the scalings are roughly $O(\sqrt{M})$ and $O(\sqrt{K})$, which suggests that our $O(\sqrt{MKT\log T})$ regret bound also reflects the right scalings of the M-UCB's regret in M and K. Finally, we compare M-UCB with state-of-the-art algorithms in numerical experiments based on a public Yahoo! dataset (Section 6.2) and a real-world digital marketing dataset (Section 6.3). In both experiments, M-UCB achieves at least 50% regret reduction with respect to the best performing state-of-the-art algorithm. The remainder of the paper is organized as follows: we briefly review the relevant literature in Section 2, then we describe the piecewise-stationary bandit model in Section 3. We discuss how to perform change-detection in the considered scenario in Section 3.3, and motivate and propose M-UCB algorithm in Section 4. We prove the regret bound in Section 5 and demonstrate experiment results in Section 6. We conclude the paper in Section 7.

2 Literature Review

Most existing work on piecewise-stationary bandit problems are based on the idea to adapt to changes passively by adjusting the weights on the rewards. For instance, Discounted UCB (D-UCB) algorithm introduced in [Kocsis and Szepesvári, 2006] (see also [Garivier and Moulines, 2011]) averages the past rewards with a discount factor, so it weighs more on the recent rewards to compute the UCB index of each arm. In [Garivier and Moulines, 2011], D-UCB policy has been proved to achieve an $O(K\sqrt{MT}\log T)$ regret. As a slight modification of D-UCB, the Sliding-Window UCB (SW-UCB) algorithm introduced in [Garivier and Moulines, 2011] computes the UCB index based on only the most recent w rewards and the regret is proved to be $O(K\sqrt{MT\log T})$. In [Auer et al., 2002b], the authors present EXP3.S algorithm which uses a regularization method to control the action switches and achieves an $O(\sqrt{MKT\log(KT)})$ regret. Using the idea in [Herbster and Warmuth, 1998], a similar algorithm called SHIFTBAND is established in [Auer, 2002], which achieves an $O(\sqrt{MKT\log(T^3K/\delta)})$ regret with probability at least $1-\delta$. Finally, Rexp3 presented in [Besbes et al., 2014] achieves an $O((K\log KV_T)^{1/3}T^{2/3})$ regret, where V_T is the total variation budget up to time T.

There has also been work exploring the idea of monitoring the reward distributions by a change-detection (CD) algorithm and triggering the reset of the learning algorithm. In contrast to the above algorithms that passively adapt to the changes, this type of algorithms actively locate the change-points and hence usually demonstrate better performance in practice. The Adapt-EvE algorithm [Hartland et al., 2007] uses Page-Hinkley Test for change-detection and restart UCB1 algorithm once a change-point is detected. Taking a Bayesian point of view, [Mellor and Shapiro, 2013] provides an algorithm by combining a Bayesian CD algorithm and Thompson Sampling. Combining one simple CD algorithm with any other MAB algorithm with a logarithm regret, [Yu and Mannor, 2009] offers a windowed mean-shift detection (WMD) algorithm that achieves an $O(KM\log T)$ regret. However, their algorithm needs to query and observe the past rewards of some unpicked arms, which violates the bandit feedback model. Combining classic MAB algorithm used in adversarial setting such as EXP3, in [Allesiardo and Féraud, 2015], the authors present a EXP3.R algorithm which resets EXP3 algorithm if one CD algorithm detects that a sub-optimal arm becomes the optimal. The EXP3.R algorithm achieves an $O(NK\sqrt{T\log T})$ regret, where N is the number of switches of the best arm during the run. Note that $N \leq M$ in general and $N = M$ in the worst case.

A recent and related work [Liu et al., 2017] uses the CUSUM algorithm for change-point detection. Compared to this work, there are two major differences. First, we use a different change-point detection (CD) method rather than CUSUM. Our CD method is simpler, and does not require to specify any parameters. Consequently, our algorithm is applicable to general piecewise-stationary bandits with bounded rewards, while [Liu et al., 2017] is restricted to the special case with Bernoulli rewards. Second, we use different analysis techniques to derive regret bounds. Leveraging renewal processes and classic metrics of change detection, a generalizable proof structure is established. It unlocks opportunities to prove similar regret bound with different

CD methods, without taking much extra effort.

3 Problem Formulation

3.1 Piecewise-Stationary Bandit

A piecewise-stationary bandit is characterized by a triple $(\mathcal{K}, \mathcal{T}, \{f_{k,t}\}_{k\in\mathcal{K},t\in\mathcal{T}})$, where $\mathcal{K} = \{1, \ldots, K\}$ is a set of K arms, $\mathcal{T} = \{1, \ldots, T\}$ is a sequence of T time steps, and $f_{k,t}$ is the reward distribution of arm k at time t. Assume that arm k's reward at time t, $X_{k,t}$, is independently drawn from $f_{k,t}$, both across arms and across time steps. Without loss of generality, assume that the support of $f_{k,t}$ is a subset of $[0, 1]$ for all $k \in \mathcal{K}, t \in \mathcal{T}$.

We define M, the number of piecewise-stationary segments in the reward process to be

$$M = 1 + \textstyle\sum_{t=1}^{T-1} \mathbb{I}\{f_{k,t} \neq f_{k,t+1} \text{ for some } k \in \mathcal{K}\}, \quad (1)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Notice that by definition, the number of change-points is $M - 1$. We use $\nu_1, \nu_2, \ldots, \nu_{M-1}$ to denote those $M - 1$ change-points, and define $\nu_0 = 0$ and $\nu_M = T$ to simplify the exposition. To emphasize the "piecewise stationary" nature of this problem, for each stationary segment $i = 1, 2, \ldots, M$ with $t \in [\nu_{i-1} + 1, \nu_i]$, we use f_k^i and μ_k^i to respectively denote the reward distribution and the expected reward of arm k on the ith segment. Define a vector that contains all expected rewards for the ith segment $\mu^i = (\mu_1^i, \ldots, \mu_K^i)^\top$, $i = 1, \ldots, M$. Note that our model allows asynchronous changes to happen at arms, i.e., the changes do not have to happen at the same time cross multiple arms. Also note that the piecewise stationary bandit model is more general than both the stochastic and the adversarial bandit models. The stochastic bandit model can be viewed as a special case of our model with $M = 1$, and the adversarial bandit model can also be viewed as a special case of our model with $M = T$.

A learning agent will repeatedly interact with this piecewise stationary bandit for T times. The agent knows $\mathcal{T}$ and $\mathcal{K}$, but does not know $\{f_{k,t}\}_{k\in\mathcal{K},t\in\mathcal{T}}$ or any of its statistics such as M and μ^i's. At each time step $t \in \mathcal{T}$, the agent chooses an action A_t based on its past actions and observations, and will receive and observe the reward $X_{A_t,t}$.

3.2 Regret Minimization

The agent's objective is to maximize its expected cumulative reward in the T time steps, i.e. $\max \mathbb{E}[\sum_{t=1}^{T} X_{A_t,t}]$, which is equivalent to minimize its T-step *cumulative regret* $\mathcal{R}(T)$ defined as

$$\mathcal{R}(T) = \textstyle\sum_{t=1}^{T} \max_{k\in\mathcal{K}} \mathbb{E}\left[X_{k,t}\right] - \mathbb{E}\left[\sum_{t=1}^{T} X_{A_t,t}\right]. \quad (2)$$

Note that the regret metric defined in (2) is stricter than the regret metric considered in most adversarial bandit papers, which is defined as

$$\widetilde{\mathcal{R}}(T) = \max_{k\in\mathcal{K}} \textstyle\sum_{t=1}^{T} \mathbb{E}\left[X_{k,t}\right] - \mathbb{E}\left[\sum_{t=1}^{T} X_{A_t,t}\right]. \quad (3)$$

Clearly $\mathcal{R}(T) \geq \widetilde{\mathcal{R}}(T)$, since the regret defined in (2) is measured with respect to the optimal piecewise stationary policy, while the regret defined in (3) is measured with respect to the optimal action in hindsight.

3.3 Sequential Change-Point Detection

Sequential change-point detection, which is rooted in classical statistical sequential analysis [Siegmund, 1985, Basseville et al., 1993], aims to detect the change in underlying distributions of a sequence of observations as quickly as possible. Commonly used methods for change-point detection include CUSUM and the generalized likelihood ratio (GLR) procedure [Page, 1954, Willsky and Jones, 1976]. However, for piecewise-stationary bandits, both pre-change and post-change distributions are unknown, and thus CUSUM is not suitable since it requires specifying both pre- and post-change distribution parameters. GLR can allow for unknown parameters (e.g. [Lai and Xing, 2010]), however, it is non-recursive and thus computationally expensive and not suitable for online implementation, especially for the high-dimensional setting.

Thus, we are not going to use CUSUM or GLR, but rather a simple change-point detection component based on comparing running sample means over a sliding window, as presented in Algorithm 1. This is computationally efficient and robust, since it has minimum parameter specification. We will show this is sufficient to guide bandit decisions as it achieves a nearly optimal regret bound.

Algorithm 1 Change detection: CD($w, b, Y_1, \ldots, Y_w$)

```
Require: An even number $w$, $w$ observations $Y_1, \ldots, Y_w$
    and a prescribed threshold $b > 0$
if $|\sum_{i=w/2+1}^{w} Y_i - \sum_{i=1}^{w/2} Y_i| > b$ then
    Return True
else
    Return False
end if
```

4 M-UCB Algorithm

Now we present the Monitored UCB (M-UCB) algorithm (as described in Algorithm 2) using a simple change-point detection component for the piecewise-stationary bandits. On a high level, M-UCB combines three ideas: (1) *uniform sampling exploration* to ensure that sufficient data are gathered for all arms to perform CD, (2) *UCB-based exploration* to learn the optimal arm on each segment, and (3) a simple change-point detection component Algorithm 1 that monitors changes and triggers exploration.

Algorithm 2 Monitored UCB (M-UCB)

```
Require: $T$, $K$, even integer $w > 0$, $b > 0$ and $\gamma \in [0,1]$
 1: Initialization: $\tau \leftarrow 0$ and $n_k \leftarrow 0 \; \forall k \in \mathcal{K}$
 2: for all $t = 1, 2, \ldots, T$ do
 3:     $A \leftarrow (t - \tau)$ mod $\lfloor K/\gamma \rfloor$.
 4:     if $A \leq K$ then
 5:         $A_t \leftarrow A$.
 6:     else
 7:         for all $k = 1, \ldots, K$ do
 8:             UCB$_k \leftarrow \frac{1}{n_k} \sum_{n=1}^{n_k} Z_{k,n} + \sqrt{\frac{2\log(t-\tau)}{n_k}}$.
 9:         end for
10:         $A_t \leftarrow \arg\max_{k \in \mathcal{K}}$ UCB$_k$.
11:     end if
12:     Play arm $A_t$ and receive the reward $X_{A_t,t}$.
13:     $n_{A_t} \leftarrow n_{A_t} + 1$; $Z_{A_t,n_{A_t}} \leftarrow X_{A_t,t}$.
14:     if $n_{A_t} \geq w$ then
15:         if CD$(w, b, Z_{A_t,n_{A_t}-w+1}, \ldots, Z_{A_t,n_{A_t}})$ = True then
16:             $\tau \leftarrow t$ and $n_k \leftarrow 0 \; \forall k \in \mathcal{K}$.
17:         end if
18:     end if
19: end for
```

Additional explanations to Algorithm 2 are as follows. The inputs include the time horizon T, the number of arms K, and three tuning parameters w, b, and γ. Here, w and b are tuning parameters for the CD algorithm (line 15), which control the power of the CD algorithm; and γ controls the fraction of the uniform sampling (line 3). Let τ denote the last detection time, and let n_k denote the number of observations from the kth arm after τ. In Remark 1 of the subsequent section, we discuss how to choose these parameters based on our theoretical analysis.

At each time t, M-UCB proceeds as follows. First, M-UCB determines whether to perform a uniform sampling exploration or a UCB-based exploration at each time, according to conditions given in line 3 and 4 to ensure that the fraction of time steps performing uniform sampling is roughly γ. If UCB-based exploration is used at time t, then the standard UCB1 indices are computed using observations from the last detection time τ to the current time, and an action is chosen greedily based on the UCB1 indices (line 7-10). By playing the chosen arm, we observe the reward and update some statistics (line 12-13). Finally, when at least w observations for the chosen arm have been gathered after the last detection time τ, M-UCB will perform CD via Algorithm 1 and restarts exploration if necessary.

We would like to emphasize that the uniform sampling exploration is crucial to M-UCB. This is because that the standard UCB exploration tends to select very infrequently the arms which it believes to be suboptimal. Thus, standard UCB exploration cannot quickly detect changes in these "infrequently visited" arms.

5 Near Optimality of M-UCB

In this section, we present our main result: the M-UCB algorithm based on simple change-point detection algorithm achieves a nearly optimal regret bound.

Recall that T is the time horizon, K is the number of arms, M is the number of piecewise-stationary segments, $\nu_0, \ldots, \nu_M$ are the change-points, and for each $i = 1, \ldots, M$, $\mu^i \in [0,1]^K$ is a vector encoding the expected rewards of all arms on segment i. We also use $\mathbb{P}$ and $\mathbb{E}$ to respectively denote the probability measure and the expectation according to the piecewise-stationary bandit characterized by the tuple $(T, K, M, \{\nu_i\}_{i=0}^M, \{\mu^i\}_{i=1}^M)$. To simplify the exposition, we define the "sub-optimal gap" of arm k on the i-th piecewise-stationary segment as

$$\Delta_k^{(i)} = \max_{\tilde{k} \in \mathcal{K}} \{\mu_{\tilde{k}}^i\} - \mu_k^i \quad \forall 1 \leq i \leq M, \, k \in \mathcal{K}, \quad (4)$$

and the amplitude of the change of arm k at the ith change-point as

$$\delta_k^{(i)} = |\mu_k^{i+1} - \mu_k^i|, \quad \forall 1 \leq i \leq M-1, \, k \in \mathcal{K}. \quad (5)$$

Moreover, recall that w, b and γ are the tuning parameters for Algorithm 2. We define $L = w\lceil K/\gamma \rceil$ for shorthanded notation.

We make the following assumptions for our theoretical analysis:

Assumption 1. *The learning agent can choose w and γ s.t. (a) $M < \lfloor T/L \rfloor$ and $\nu_{i+1} - \nu_i > L, \forall 0 \leq i \leq M-1$, and (b) $\forall 1 \leq i \leq M-1$, $\exists k \in \mathcal{K}$ s.t. $\delta_k^{(i)} \geq 2\sqrt{\log(2KT^2)/w} + 2\sqrt{\log(2T)/w}$.*

We would like to clarify that Assumption 1 is only required for the analysis; Our proposed M-UCB algorithm (Algorithm 2) can be implemented regardless of this assumption. As is shown in Section 6, in the real-world experiments, our algorithm works well even if Assumption 1 does not hold. Notice that relevant literature, such as [Liu et al., 2017], makes similar assumptions. Moreover, compared with [Liu et al., 2017], we have relaxed a major assumption: they also assume the rewards are Bernoulli, which is not assumed in our algorithm and analysis.

We now briefly motivate and explain Assumption 1. Intuitively, Assumption 1(a) means that the length of the time interval between two consecutive change-points is larger than L. This guarantees that Algorithm 2 can select at least w samples from every arm, and these samples are used to feed the CD algorithm. Assumption 1(b) means that the change amplitude is over certain threshold for at least one arm at each change-point. This guarantees that the CD algorithm is able to detect the change quickly with limited information. If a lower bound $\delta > 0$ on $\min_i \max_{k \in \mathcal{K}} \delta_k^{(i)}$

can be assumed, then one can choose

$$w \approx (4/\delta^2) \cdot [(\log(2KT^2))^{1/2} + (\log(2T))^{1/2}]^2 \qquad (6)$$

to satisfy Assumption 1(b). Our main result is the following regret bound:

Theorem 1. *Running Algorithm 2 with w and γ satisfying Assumption 1 and $b = [w\log(2KT^2)/2]^{1/2}$, we have*

$$\begin{aligned}\mathcal{R}(T) \leq & \underbrace{\textstyle\sum_{i=1}^{M} \tilde{C}_i}_{(a)} + \underbrace{\gamma T}_{(b)} \\ & + \underbrace{\textstyle\sum_{i=1}^{M-1} \frac{2K\cdot\min(\frac{w}{2},\lceil\frac{b}{\delta^{(i)}}\rceil+3\sqrt{w})}{\gamma}}_{(c)} + \underbrace{3M}_{(d)},\end{aligned} \qquad (7)$$

where $\delta^{(i)} = \max_{k\in\mathcal{K}} \delta_k^{(i)}$ and $\tilde{C}_i = 8\sum_{\Delta_k^{(i)}>0} \frac{\log T}{\Delta_k^{(i)}} + \left(1 + \frac{\pi^2}{3} + K\right) \sum_{k=1}^{K} \Delta_k^{(i)}$.

Theorem 1 reveals that the regret incurred by M-UCB can be decomposed into four terms. Terms (a) and (b) in equation (7) bound on the exploration costs: term (a) bounds the cost of the UCB-based exploration, and term (b) bounds the cost of the uniform sampling. On the other hand, terms (c) and (d) bound the change-point detection costs: term (c) bounds the cost associated with the detection delay of the CD algorithm, and term (d) is incurred by the unsuccessful and incorrect detections of the change-points. The following corollary follows immediately from Theorem 1.

Corollary 1. *Assume $\delta > 0$ is a lower bound on $\min_i \max_{k\in\mathcal{K}} \delta_k^{(i)}$. If we run Algorithm 2 with a window-length w,*

$$b = [w\log(2KT^2)/2]^{1/2},$$

and

$$\gamma = \sqrt{(M-1)K \cdot \min(w/2, \lceil b/\delta \rceil + 3\sqrt{w})/(2T)},$$

then we have

$$\begin{aligned}&\mathcal{R}(T) \leq \textstyle\sum_{i=1}^{M} \tilde{C}_i \\ &+ 4\sqrt{(M-1)TK \cdot \min(w/2, \lceil b/\delta \rceil + 3\sqrt{w})} + 3M.\end{aligned} \qquad (8)$$

For any fixed w, the upper bound for the regret in equation (8) is

$$O(\sqrt{MKT\log T}) = \tilde{O}(\sqrt{MKT}),$$

where $\tilde{O}$ notation hides logarithmic factors. Compared with the lower bound in $\Omega(\sqrt{T})$ [Garivier and Moulines, 2008], our regret bound is asymptotically tight up to a logarithmic factor. In Section 6.1, we validate numerically that when scaled by $1/\sqrt{T}$, the scalings of Algorithm 2's regret in M and K are roughly $O(\sqrt{M})$ and $O(\sqrt{K})$, as is suggested in Corollary 1. We leave the derivation of the lower bound in K and M to future work.

Corollary 1 also sheds some insights on how to choose tuning parameters w, b, and γ in Algorithm 2.

Remark 1 (Algorithm Parameter Tuning)**.** *We now discuss how to choose algorithm parameters w, b, and γ based on Corollary 1. In practice, we mainly care about large changes since small changes do not incur much regret. Assume an minimum change size $\tilde{\delta} > 0$ then following from equation* (6) *and Corollary 1, we can choose the window size $w \approx (4/\tilde{\delta}^2) \cdot [(\log(2KT^2))^{1/2} + (\log(2T))^{1/2}]^2$, $b \approx [w\log(2KT^2)/2]^{1/2}$, and $\gamma \approx (\sum_{i=1}^{M-1} K \cdot \min(w/2, \lceil b/\tilde{\delta} \rceil + 3\sqrt{w})/(2T)^{1/2}$.*

The proof outline for Theorem 1 is provided in section 5.1 and more details for technical lemmas are given in Appendix A. The main steps of the proof are as follows. First, we rely on standard bandit analysis to decompose $\mathcal{R}(T)$ over a set of "good" events and a set of "bad" events: the good events include all the sample paths that Algorithm 2 reinitializes the UCB algorithm quickly after any change-point. The set of bad events includes all the sample paths that Algorithm 2 that either fails to reinitialize the UCB algorithm quickly when there is a change-point or incorrectly reinitializes the UCB algorithm when there is not any change-point. This enables us to couple the change-point detection analysis with bandit analysis, and we identify that the parameters specified in Theorem 1 will ensure that the set of good events occurs with a high probability.

5.1 Proof Outline of Theorem 1

In this subsection, we outline the proof for Thereom 1. Detailed proofs are provided in Appendix A.

First, we bound the regret incurred by Algorithm 2 in the stationary scenario with $M = 1$, $\nu_0 = 0$, and $\nu_1 = T$.

Lemma 1 (Regret bound for the M-UCB algorithm in stationary scenarios)**.** *Consider a stationary scenario with $M = 1$, $\nu_0 = 0$, and $\nu_1 = T$. Under Algorithm 2 with parameter w, b and γ, we have that*

$$\mathcal{R}(T) \leq T \cdot \mathbb{P}(\tau_1 \leq T) + \tilde{C} + \gamma T, \qquad (9)$$

where τ_1 is the first detection time and

$$\tilde{C} = 8\textstyle\sum_{\Delta_k^{(1)}>0} \frac{\log T}{\Delta_k^{(1)}} + \left(1 + \frac{\pi^2}{3} + K\right) \sum_{k=1}^{K} \Delta_k^{(1)}.$$

Remark 2. *Lemma 1 shows that the regret for the M-UCB algorithm in the stationary scenario is incurred by three sources. The term $\mathbb{P}(\tau_1 \leq T)$ on the right-hand side of (9) is the probability of raising one false alarm. This can be controlled to be small through setting appropriate algorithm parameters. The term $\tilde{C}$ is the classic regret bound for the UCB-based exploration in stationary scenarios. The term γT is incurred by the uniform sampling exploration.*

Please refer to Appendix A.1 for the proof of Lemma 1. Next, we bound the probability of restarting Algorithm 2 when there is no change-point. This probability is equivalent to the probability of the CD algorithm raising a false alarm in the stationary scenario discussed above.

Lemma 2 (Probability of raising false alarms in the stationary scenario)**.** *Consider a stationary scenario with $M = 1$. Then under Algorithm 2 with parameter $w < T$, b and γ, we have that*

$$\mathbb{P}(\tau_1 \le T) \le wK\left(1 - \left(1 - 2\exp\left(-2b^2/w\right)\right)^{\lfloor T/w \rfloor}\right),$$

where τ_1 is the first detection time.

Remark 3. *Using the fact that $(1 - x)^a > 1 - ax$ for any $a > 1$ and $0 < x < 1$, we have that in Lemma 2 $\mathbb{P}(\tau_1 \le T) \le 2KT\exp(-2b^2/w)$. Therefore, setting $b = [w\log(2KT^2)/2]^{1/2}$ we have that $\mathbb{P}(\tau_1 \le T) \le 1/T$, which means that it is expected to raise at most only one false alarm in a stationary scenario with T time steps.*

Please refer to Appendix A.2 for the proof of Lemma 2. Then, we establish a lower bound on the probability that the CD algorithm (Algorithm 1) achieves a successful detection in scenarios with one change-point, i.e. $M = 2$.

Lemma 3 (Probability of achieving a successful detection with $M = 2$)**.** *Consider a piecewise-stationary scenario with $M = 2$, and recall that $L = w\lceil K/\gamma\rceil$. Assume that $\nu_2 - \nu_1 > L/2$. For any $\mu^1, \mu^2 \in [0,1]^K$ satisfying*

$$\delta_{\tilde{k}}^{(1)} \ge 2b/w + c$$

for some $\tilde{k} \in \mathcal{K}$ and $c > 0$, under Algorithm 2, we have that

$$\mathbb{P}(\nu_1 < \tau_1 \le \nu_1 + L/2 \mid \tau_1 > \nu_1) \ge 1 - 2\exp\left(-wc^2/4\right).$$

Remark 4. *Setting $b = \sqrt{w\log(2KT^2)/2}$ and $c = 2\sqrt{\log(2T)/w}$ in Lemma 3, we have that with probability at least $1 - 1/T$, a change can be detected in $L/2$ steps after the change occurs , provided that $\delta_{\tilde{k}}^{(1)}$ is greater than $C_1/\sqrt{w}$ for some constant C_1 that only depends on T and K. This shows that we can set a larger w to achieve a successful detection of a smaller change.*

Please refer to Appendix A.3 for the proof of Lemma 3. Lemma 3 relates the tuning parameter b and w to the smallest change that the CD algorithm can successfully detect with high probability.

In the next lemma, for scenarios with $M = 2$, we bound the expected detection delay (EDD) by a function of the change amplitude, given that the change can be detected successfully. In other words, Lemma 3 characterizes a lower bound on change amplitude to ensure that the detection delay is no more than $L/2$ with high probability. When the change amplitude is not so small, the EDD can be smaller than $L/2$, as presented in the following lemma.

Lemma 4 (Expected detection delay)**.** *Consider a piecewise-stationary scenario with $M = 2$, and recall that $L = w\lceil K/\gamma\rceil$. Assume that $\nu_2 - \nu_1 > L/2$. For any $\mu^1, \mu^2 \in [0,1]^K$ satisfying $\delta_{\tilde{k}}^{(1)} > 2b/w + c$ for some $\tilde{k} \in \mathcal{K}$, we have that*

$$\begin{aligned}&\mathbb{E}[\tau_1 - \nu_1 \mid \nu_1 < \tau_1 \le \nu_1 + L/2]\\ &\le \frac{\min(L/2, \lceil b/\delta_{\tilde{k}}^{(1)}\rceil + 3\sqrt{w}\cdot\lceil K/\gamma\rceil)}{1 - 2\exp\left(-wc^2/4\right)}.\end{aligned}$$

Please refer to Appendix A.4 for the proof of Lemma 4.

Theorem 1 can be proved based on the above four lemmas and properties of renewal processes. Specifically, we decompose $\mathcal{R}(T)$ over a set of "good" events and a set of "bad" events: the good events include all the sample paths that Algorithm 2 reinitializes the UCB algorithm correctly and quickly after all change-points. The set of bad events includes all the sample paths that Algorithm 2 that either fails to reinitialize the UCB algorithm quickly when there is a change-point (large detection delay) or incorrectly reinitializes the UCB algorithm when there is not any change-point (false alarm). Lemma 2 and 3 can be used to upper bound the probabilities of the bad events. Together with the naive bound $\mathcal{R}(T) \le T$, we can bound the regret in the bad events. On the other hand, Lemma 1 and 4 can be used to upper bound the regret in the good events. Please refer to Appendix A.5 for the detailed proof for Theorem 1.

6 Experiments

In this section, we present some numerical experiments to validate the performance of M-UCB. We first verify the scalings of M-UCB's regret in M and K and then compare M-UCB with state-of-the-art algorithms on a publicly available benchmark Yahoo! dataset and a real-world digital marketing dataset.

6.1 Regret Scalings in M and K

To eliminate the scaling issue caused by different T's, in this subsection we scale the empirical regret by $1/\sqrt{T}$. For the illustrative purposes, we assume the rewards are Bernoulli distributed.

We first show the regret scaling in M. We fix $K = 10$, and let the locations of change-points to be evenly spaced with interval of length 20000 so for any M we have $T = 20000 \cdot M$. For the reward sequence of each arm, we set $\mu^{(i)} = \mu$ when i is odd and $\mu^{(i)} = 1 - \mu$ when i is even, where $\mu \in [0,1]^K$ is randomly chosen such that the difference between the largest and smallest entry is larger than 0.6. Consider an upper bound 20000×25 for T, an upper bound 10 for K and a lower bound 0.6 for δ. Based on Remark 1, we can set $w = 800$ and $b = \sqrt{(w/2)\cdot\log(2KT^2)}$ and

$\gamma = \sqrt{(M-1)K \cdot (2b + 3\sqrt{w})/(2T)}$. For each M, we randomly generate 100 instances, and for each instance, we run the M-UCB algorithm for 50 times. We generate the averaged regret by averaging over these 5000 simulations. The results are shown in Figure 1, which can be fitted using the simple model $y = c + ax^b$ to obtain an estimated order $b = 0.55$, which means that the regret is roughly on the order of $O(\sqrt{M})$.

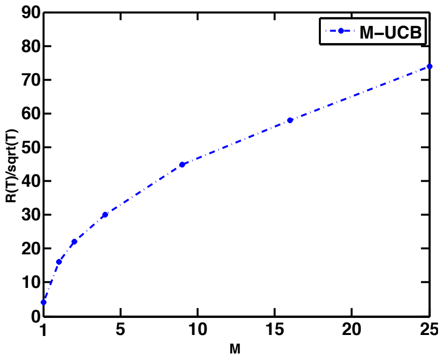

Figure 1: Cumulative regret of M-UCB up to time T scaled by $\sqrt{T}$ versus M.

We then demonstrate the regret scaling in K. We fix $M = 4$, $T = 3 \times 10^5$ and let the change-points to be evenly spaced. For each $i = 1, \ldots, 4$, we randomly generate $\mu^{(i)} \in [0,1]^K$ such that the difference between the largest and smallest entry is larger than 0.6 and one combination of K and $(\mu^{(i)})_{i=1}^4$ forms one instance. We set the same algorithmic parameters as those in the first simulation example. We then generate 100 random instances, and for each instance we repeat our algorithm for 50 times to obtain the averaged regret. The results are shown in Figure 2, which again can be fitted with the simple model $y = c + ax^b$ to obtain an estimated order $b = 0.53$, which means that the regret grows roughly at a rate of $O(\sqrt{K})$.

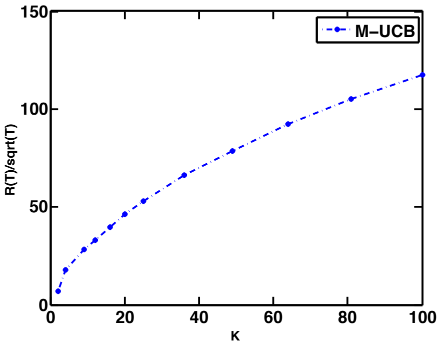

Figure 2: Cumulative regret of M-UCB up to time T scaled by $\sqrt{T}$ versus K.

The above results suggest that the regret bound in Corollary 1 is roughly tight in M and K for M-UCB algorithm.

6.2 Experiment on Yahoo! Dataset

We compare the expected cumulative regret of different algorithms using the benchmark dataset publicly published by Yahoo![1]. This dataset provides a binary value for each arrival to represent whether the user clicks the specified article [Chu et al., 2009, Li et al., 2011]. We use one arm to represent one article and assume a Bernoulli reward (one if the user clicks the article and zero otherwise). The goal is set to maximize the expected number of clicked articles using strategies that select one article for each arrival sequentially. We randomly select six different articles of which the click-through rates are greater than zero within one five-day horizon, where the click-through rates are computed by taking the mean of the number of times each article being clicked every 43200 seconds (which corresponds to a half day). If the difference between the estimated click-through rate of the current half day and that of the last half day is less than 0.01, we set the click-through rate of the current half day as that of the last half day. In this way, we obtain a piecewise-stationary scenario with $T = 43200 \times 10 = 4.32 \times 10^5$, $K = 6$ and $M = 9$, as shown in Figure 3.

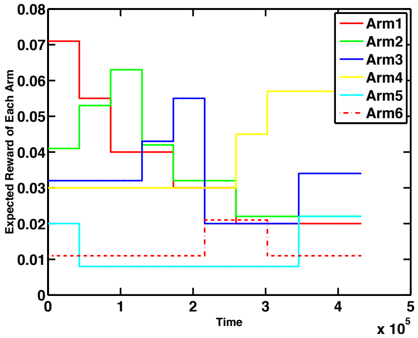

Figure 3: Click-through rates computed from Yahoo! dataset with $T = 4.32 \times 10^5, K = 6$ and $M = 9$.

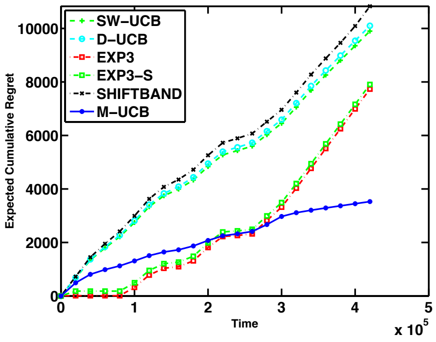

Figure 4: Expected cumulative regrets for different algorithms under the piecewise-stationary scenario shown in Figure 3.

Along with our algorithm using the same w, b and γ as

[1] Yahoo! Front Page Today Module User Click Log Dataset on https://webscope.sandbox.yahoo.com

those in subsection 6.1, we run five other algorithms, Discounted UCB (D-UCB), Sliding-Window UCB (SW-UCB), EXP3, EXP3.S and SHIFTBAND for comparison. Based on the theoretical results in [Garivier and Moulines, 2008], we choose $\gamma = 1 - 0.25\sqrt{(M-1)/T}$ and $\xi = 0.5$ for D-UCB and choose $\tau = 2\sqrt{T \log T/(M-1)}$ in SW-UCB. Based on Theorem 1 in [Auer, 2002], we choose $\delta = 0.05$, $\alpha = 2\sqrt{\log(T^3 K/\delta)}$, $\beta = 1/T$ and $\eta = \sqrt{\log(TK)M/(TK)}$ for the SHIFTBAND algorithm. For EXP3 and EXP3.S algorithms we select parameters to be the same as those in [Auer et al., 2002b]. [2] The expected cumulative regret is computed by taking the average of the regrets for 100 independent Monte Carlo trials, as shown in Figure 4.

The results show that the M-UCB algorithm achieves a better performance than other algorithms even if all the algorithms seem to have a sub-linear regret. Compared with EXP3 and EXP3.S, M-UCB achieves a 50% reduction of the cumulative regret and this number is 60% if we make comparisons with SW-UCB, D-UCB and SHIFTBAND algorithms.

It is worth clarifying that our experiment on the Yahoo dataset does not satisfy Assumption 1 in the sense that it contains many small-magnitude changes. Thus, we believe it is a fair comparison for all algorithms. Specifically, in this case $K = 6$ and $T = 432000$, and we choose $w = 800$ for M-UCB. Thus, Assumption 1 requires that all changes have magnitude no less than 0.64. However, as is shown in Figure 3, all changes have magnitude less than 0.1. This experiment shows that M-UCB works well and outperforms state-of-the-art baselines even if Assumption 1 does not hold.

6.3 Digital Marketing

In addition to the experiment on Yahoo dataset, we have also compared M-UCB (same w, b and γ as in 6.1) with the state-of-the-art algorithms on a real-world digital marketing dataset. The experiment setup is similar to the one with Yahoo dataset with $K = 10$ and $T = 2520000$. The expected arm rewards and the cumulative regrets are shown in Figure 5 and Figure 6, respectively.

Experiment results show that M-UCB can reduce the cumulative regret of SW-UCB by 60%, and reduce the cumulative regret of EXP3 by 93%. These experiment results suggest that, by adaptively detecting and quickly adapting to the changes, M-UCB algorithm is expected to achieve significant regret reductions compared with the state-of-the-art. Note that this setting also violates the Assumption 1. Specifically, Assumption 1 requires that all changes have magnitude no less than 0.68, but many changes have smaller magnitudes.

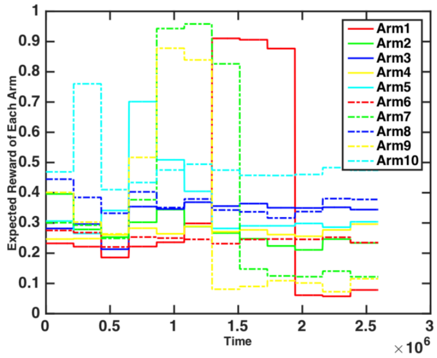

Figure 5: Expected rewards computed based on the digital marketing dataset with $T = 2.52 \times 10^6$, $K = 10$ and $M = 12$.

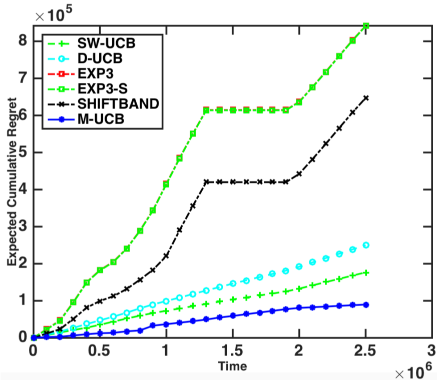

Figure 6: Expected cumulative regrets for different algorithms under the piecewise-stationary scenario shown in Figure 5.

The lower bound computed for δ is 0.68 but many changes have smaller magnitudes.

7 Conclusion

In this paper, we have developed a so-called M-UCB algorithm (Algorithm 2) for piecewise-stationary bandits with bounded rewards. M-UCB combines the UCB (with uniform exploration) with a simple change-point detection component based on running sample means over a sliding window. We prove that M-UCB algorithm achieves a nearly optimal regret bound on the order of $O(\sqrt{MKT \log T})$ under mild technical conditions. Our experiment results also show that it can achieve significant regret reduction with respect to the state-of-the-art algorithms in numerical experiments based on real-world datasets.

Our proposed M-UCB algorithm is based on the classical UCB1 algorithm. We may improve by considering other exploration schemes (e.g. KL-UCB, Thompson sampling) in the current setup. One can foresee that as long as the exploration schemes are statistically efficient, then a variant of our analysis will carry through.

[2] Specifically, The parameters for EXP3 and EXP3.S are selected based on Corollary 3.2 and 8.2 in [Auer et al., 2002b].

References

[Alaya-Feki et al., 2008] Alaya-Feki, A. B. H., Moulines, E., and LeCornec, A. (2008). Dynamic spectrum access with non-stationary multi-armed bandit. In *Signal Processing Advances in Wireless Communications, 2008. SPAWC 2008. IEEE 9th Workshop on*, pages 416–420. IEEE.

[Allesiardo and Féraud, 2015] Allesiardo, R. and Féraud, R. (2015). Exp3 with drift detection for the switching bandit problem. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–7. IEEE.

[Auer, 2002] Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

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

[Auer et al., 2002b] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

[Basseville et al., 1993] Basseville, M., Nikiforov, I. V., et al. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs.

[Besbes et al., 2014] Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pages 199–207.

[Bouneffouf et al., 2012] Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. (2012). A contextual-bandit algorithm for mobile context-aware recommender system. In *International Conference on Neural Information Processing*, pages 324–331. Springer.

[Cesa-Bianchi and Lugosi, 2006] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

[Chu et al., 2009] Chu, W., Park, S.-T., Beaupre, T., Motgi, N., Phadke, A., Chakraborty, S., and Zachariah, J. (2009). A case study of behavior-driven conjoint analysis on yahoo!: front page today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104. ACM.

[Garivier and Moulines, 2008] Garivier, A. and Moulines, E. (2008). On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*.

[Garivier and Moulines, 2011] Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 174–188. Springer.

[Girgin et al., 2012] Girgin, S., Mary, J., Preux, P., and Nicol, O. (2012). Managing advertising campaigns: an approximate planning approach. *Frontiers of Computer Science*, 6(2):209–229.

[Hartland et al., 2007] Hartland, C., Baskiotis, N., Gelly, S., Sebag, M., and Teytaud, O. (2007). Change point detection and meta-bandits for online learning in dynamic environments. *CAp*, pages 237–250.

[Herbster and Warmuth, 1998] Herbster, M. and Warmuth, M. K. (1998). Tracking the best expert. *Machine learning*, 32(2):151–178.

[Kocsis and Szepesvári, 2006] Kocsis, L. and Szepesvári, C. (2006). Discounted ucb. In *2nd PASCAL Challenges Workshop*, pages 784–791.

[Kveton et al., 2014] Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. (2014). Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*.

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

[Lai and Xing, 2010] Lai, T. L. and Xing, H. (2010). Sequential change-point detection when the pre-and post-change parameters are unknown. *Sequential analysis*, 29(2):162–175.

[Li et al., 2011] Li, L., Chu, W., Langford, J., and Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM.

[Littlestone and Warmuth, 1994] Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

[Liu et al., 2017] Liu, F., Lee, J., and Shroff, N. (2017). A change-detection based framework for piecewise-stationary multi-armed bandit problem. *arXiv preprint arXiv:1711.03539*.

[Mellor and Shapiro, 2013] Mellor, J. and Shapiro, J. (2013). Thompson sampling in switching environments with bayesian online change detection. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 442–450.

[Page, 1954] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.

[Schwartz et al., 2017] Schwartz, E. M., Bradlow, E. T., and Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*.

[Siegmund, 1985] Siegmund, D. (1985). *Sequential analysis: tests and confidence intervals*. Springer Science & Business Media.

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

[Vermorel and Mohri, 2005] Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *ECML*, volume 3720, pages 437–448. Springer.

[Villar et al., 2015] Villar, S. S., Bowden, J., and Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199.

[Willsky and Jones, 1976] Willsky, A. and Jones, H. (1976). A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic control*, 21(1):108–112.

[Yu and Mannor, 2009] Yu, J. Y. and Mannor, S. (2009). Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1177–1184. ACM.