Adaptive Exploration-Exploitation Tradeoff for Opportunistic Bandits

Huasen Wu [1] **Xueying Guo** [2] **Xin Liu** [2]

Abstract

In this paper, we propose and study opportunistic bandits - a new variant of bandits where the regret of pulling a suboptimal arm varies under different environmental conditions, such as network load or produce price. When the load/price is low, so is the cost/regret of pulling a suboptimal arm (e.g., trying a suboptimal network configuration). Therefore, intuitively, we could explore more when the load/price is low and exploit more when the load/price is high. Inspired by this intuition, we propose an Adaptive Upper-Confidence-Bound (AdaUCB) algorithm to adaptively balance the exploration-exploitation tradeoff for opportunistic bandits. We prove that AdaUCB achieves $O(\log T)$ regret with a *smaller coefficient* than the traditional UCB algorithm. Furthermore, AdaUCB achieves $O(1)$ regret with respect to T if the exploration cost is zero when the load level is below a certain threshold. Last, based on both synthetic data and real-world traces, experimental results show that AdaUCB significantly outperforms other bandit algorithms, such as UCB and TS (Thompson Sampling), under large load/price fluctuations.

1. Introduction

In existing studies of multi-armed bandits (MABs) (Auer et al., 2002; Bubeck & Cesa-Bianchi, 2012), pulling a suboptimal arm results in a constant regret. While this is a valid assumption in many existing applications, there exists a variety of applications where the actual regret of pulling a suboptimal arm may vary depending on external conditions. Consider the following application scenarios.

Motivating scenario 1: price variation. MAB has been widely used in studying effective procedures and treatments (Lai, 1987; Press, 2009; Villar et al., 2015), including in agriculture. In agriculture, price often varies significantly for produce and livestock. For example, the pork price varied from \$0.46/lb to \$1.28/lb, and orange \$608/ton to \$1140/ton, in 2014-2017 (Index Mundi). Commodity price forecast has achieved high accuracy and been widely used for production decisions (Brandt & Bessler, 1983). In this scenario, different treatments can be considered as arms. The effectiveness of a particular treatment is captured by the value of the arm, and is independent of the market price of the product. (The latter is true because an experiment in one farm, among tens of thousands of such farms in the US, has negligible impact on the overall production and thus the commodity's market price.) The monetary reward is proportional to price and to the effectiveness of the treatment. The goal of a producer is to minimize the overall monetary regret, compared to the oracle. Therefore, intuitively, when the product price is low, the monetary regret of pulling a suboptimal arm is low, and vice versa.

Motivating scenario 2: load variation. Network configuration is widely used in wireless networks, data-center networks, and the Internet, in order to control network topology, routing, load balancing, and thus improve the overall performance. For example, in a cellular network, a cell tower has a number of parameters to configure, including radio spectrum, transmission power, antenna angle and direction, etc. The configuration of such parameters can greatly impact the overall performance, e.g., coverage, throughput, and service quality. A network configuration can be considered as an arm, where its performance needs to be learned. Networks are typically designed and configured to handle the peak load, and thus we hope to learn the best configuration for the peak load.

Network traffic load fluctuates over time. When the network load is low, we can inject dummy traffic into the network so that the total load, the real load plus the dummy load, resembles the peak load. It allows us to learn the performance of the configuration under the peak load. At the same time, the regret of using a suboptimal configuration is low because the real load affected is low. Furthermore, in practice, we can set the priority of the dummy traffic to be lower than that of the real traffic. Because networks handle high priority

[1] Twitter Inc., San Francisco, California, USA; [2] University of California, Davis, California, USA. This work was partially completed while the first author was a postdoctoral researcher at University of California, Davis. Correspondence to: Xin Liu <xinliu@ucdavis.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

traffic first, low priority traffic results in little or no impact on the high priority traffic (Walraevens et al., 2003). In this case, the regret on the actual load is further reduced, or even negligible (when the suboptimal configuration is sufficient to handle the real load).

Opportunistic bandits. Motivated by these application scenarios, we study opportunistic bandits in this paper. Specifically, we define *opportunistic bandit* as a bandit problem with the following characteristics: 1) The best arm does not change over time. 2) The exploration cost (regret) of a suboptimal arm varies depending on a time-varying external condition that we refer to as **load** (which is the price in the first scenario). 3) The load is revealed before an arm is pulled, so that one can decide which arm to pull depending on the load. As its name suggests, in opportunistic bandits, one can leverage the opportunities of load variation to achieve a lower regret. In addition to the previous two examples, opportunistic bandit algorithms can be applied to other scenarios that share the above characteristics.

We note that opportunistic bandits significantly differs from non-stationary bandits (Garivier & Moulines, 2011; Besbes et al., 2014). In non-stationary bandits, the expected reward of each arm varies and the optimal arm may change over time, e.g., because of the shift of interests. In opportunistic bandits, the optimal arm does not change over time, but the regret of trying a suboptimal arm changes depending on the load. In other words, in non-stationary bandits, the dynamics of the optimal arm make finding the optimal arm more challenging. In contrast, in opportunistic bandits, the time-varying nature of the load provides opportunities to reduce the regret of finding the fixed optimal arm. Because of such fundamental differences, in non-stationary bandits, one can show polynomial regret (e.g., $\Omega(T^{2/3})$ (Besbes et al., 2014)) because one has to keep track of the optimal arm. In opportunistic bandits, we can show $O(\log T)$ (or even $O(1)$ in certain special cases) regret because we can push more exploration to slots when the regret is lower.

We also note the connection and difference between opportunistic bandits and contextual bandits (Zhou, 2015; Wu et al., 2015; Li et al., 2010; Chu et al., 2011). Broadly speaking, opportunistic bandits can be considered as a special case of contextual bandits where we can consider the load as the context. However, general contextual bandits do not take advantages of the unique properties of opportunistic bandits, in particular, the optimal bandit remains the same, and regrets differ under different contexts (i.e., load). To follow this line, the performance of contextual bandits has been compared in Appendix D.3.

Contributions. In this paper, we propose an Adaptive Upper-Confidence-Bound (AdaUCB) algorithm to dynamically balance the exploration-exploitation tradeoff in opportunistic bandits. The intuition is clear: we should explore more when the load is low and exploit more when the load is high. The design challenge is to quantify the right amount of exploration and exploitation depending on the load. The analysis challenge is due to the inherent coupling over time and thus over bandits under different conditions. In particular, due to the randomness nature of bandits, the empirical estimates of the expected rewards could deviate from the true values, which could lead to suboptimal actions when the load is high. We address these challenges by studying the lower bounds on the number of pulls of the suboptimal arms under low load. Because the exploration factor is smaller under high load than that under low load, it requires less information accuracy to make the optimal decision under high load. Thus, with an appropriate lower bound on the number of pulls of the suboptimal arms under low load, we can show that the information obtained from the exploration under the low load is sufficient for accurate decisions under the high load. As a result, the exploration under high load is reduced and thus so does the overall regret.

To the best of our knowledge, this is **the first work proposing and studying opportunistic bandits** that aims to adaptively balance the exploration-exploitation tradeoff considering load-dependent regrets. We propose AdaUCB, an algorithm that adjusts the exploration-exploitation tradeoff according to the load level. We prove that AdaUCB achieves $O(\log T)$ regret with a smaller coefficient than the traditional UCB algorithm. Furthermore, AdaUCB achieves $O(1)$ regret with respect to T in the case where the exploration cost is zero when the load level is smaller than a certain threshold. Using both synthetic and real-world traces, we show that AdaUCB significantly outperforms other bandit algorithms, such as UCB and TS (Thompson Sampling), under large load fluctuations.

2. System Model

We study an opportunistic bandit problem, where the exploration cost varies over time depending on an external condition, called **load** here. Specifically, consider a K-armed stochastic bandit system. At time t, each arm has a random *nominal reward* $X_{k,t}$, where $X_{k,t} \in [0,1]$ are independent across arms, and i.i.d. over time, with mean value $\mathbb{E}[X_{k,t}] = u_k$. Let $u^* = \max_k u_k$ be the maximum expected reward and $k^* = \arg\max u_k$ be the best arm. The arm with the best nominal reward does not depend on the load and does not change over time.

Let $L_t \geq 0$ be the load at time t. For simplicity, we assume $L_t \in [0,1]$. The agent observes the value of L_t before making the decision; i.e., the agent pulls an arm a_t based on both L_t and the historical observations, i.e., $a_t = \Gamma(L_t, \mathcal{H}_{t-1})$, where $\mathcal{H}_{t-1} = (L_1, a_1, X_{a_1,1}, \ldots, L_{t-1}, a_{t-1}, X_{a_{t-1},t-1})$ represents the historical observations. The agent then receives an *actual reward* $L_t X_{a_t,t}$. While the underlying nominal reward $X_{a_t,t}$ is independent of L_t conditioned on a_t, the actual reward depends on L_t. We also assume that the agent can observe the value of $X_{a_t,t}$ after pulling arm a_t at time t.

This model captures the essence of opportunistic bandits and its assumptions are reasonable. For example, in the agriculture scenario, $X_{a_t,t}$ captures the effectiveness of a treatment, e.g., the survival rate or the yield of an antibiotic treatment. The value of $X_{a_t,t}$ can always be observed by the agent after applying treatment a_t at time t. Conditioned on a_t, $X_{a_t,t}$ is also independent of L_t, the price of the commodity. Meanwhile, the actual reward, i.e., the monetary reward, is modulated by L_t (the price) as $L_t X_{a_t,t}$. In the network configuration example, $X_{a_t,t}$ captures the impact of a configuration at the peak load, e.g., success rate, throughput, or service quality score. Because the total load (the real load plus the dummy load) resembles the peak load, $X_{a_t,t}$ is independent of the real load L_t conditioned on a_t, and can always be observed. Further, because the real load is a portion of the total load and the network can identify real traffic from dummy traffic, the actual reward is thus a portion of the total reward, modulated by the real load as $L_t X_{a_t,t}$.

If system statistics are known *a priori*, then the agent will always pull the best arm and obtain the expected total reward $u^*\mathbb{E}[\sum_{t=1}^{T} L_t]$. Thus, the regret of a policy Γ is defined as

$$R_\Gamma(T) = u^*\mathbb{E}\big[\sum_{t=1}^{T} L_t\big] - \sum_{t=1}^{T} \mathbb{E}[L_t X_{a_t,t}]. \tag{1}$$

In particular, when L_t is i.i.d. over time with mean value $\mathbb{E}[L_t] = \bar{L}$, the total expected reward for the oracle solution is $u^*\bar{L}T$ and the regret is $R_\Gamma(T) = u^*\bar{L}T - \sum_{t=1}^{T} \mathbb{E}[L_t X_{a_t,t}]$. Because the action a_t can depend on L_t, it is likely that $\mathbb{E}[L_t X_{a_t,t}] \neq \bar{L}\mathbb{E}[X_{a_t,t}]$.

3. Adaptive UCB

We first recall a general version of the classic UCB1 (Auer et al., 2002) algorithm, referred to as UCB(α), which always selects the arm with the largest index defined in the following format:

$$\hat{u}_k(t) = \bar{u}_k(t) + \sqrt{\frac{\alpha \log t}{C_k(t-1)}},\ 1 \leq k \leq K,$$

where α is a constant, $C_k(t-1)$ is the number of pulls for arm-k before t, and $\bar{u}_k(t) = \frac{1}{C_k(t-1)} \sum_{\tau=1}^{t-1} \mathbb{1}(a_\tau = k) X_{k,\tau}$. It has been shown that UCB(α) achieves logarithmic regret in stochastic bandits when $\alpha > 1/2$ (Bubeck, 2010). UCB1 in (Auer et al., 2002) is a special case with $\alpha = 2$.

Algorithm 1 AdaUCB

```
Init: $\alpha > 0.5$, $C_k(t) = 0$, $\bar{u}_k(t) = 1$.
for $t = 1$ to $K$ do
    Pull each arm once and update $C_k(t)$ and $\bar{u}_k(t)$ accordingly;
end for
for $t = K + 1$ to $T$ do
    Observe $L_t$;
    Calculate UCB: for $k = 1, 2, \ldots, K$,
        $\hat{u}_k(t) = \bar{u}_k(t) + \sqrt{\frac{\alpha(1 - \tilde{L}_t)\log t}{C_k(t-1)}}$,    (2)
    where $\tilde{L}_t$ is the normalized load defined in Eq. (4);
    Pull the arm with the largest $\hat{u}_k(t)$:
        $a_t = \arg\max_{1 \leq k \leq K} \hat{u}_k(t)$;    (3)
    Update $\bar{u}_k(t)$ and $C_k(t)$;
end for
```

In this work, we propose an AdaUCB algorithm for opportunistic bandits. In order to capture different ranges of L_t, we first normalize L_t to be within $[0, 1]$:

$$\tilde{L}_t = \frac{[L_t]_{l^{(-)}}^{l^{(+)}} - l^{(-)}}{l^{(+)} - l^{(-)}}, \tag{4}$$

where $l^{(-)}$ and $l^{(+)}$ are the lower and upper thresholds for truncating the load level, and $[L_t]_{l^{(-)}}^{l^{(+)}} = \max\{l^{(-)}, \min(L_t, l^{(+)})\}$. Load normalization reduces the impact of different load distributions. It also restricts the coefficient of the exploration term in the UCB indices, which avoids under or over explorations. To achieve good performance, the truncation thresholds should be appropriately chosen and can be learned online in practice, as discussed in Sec. 4.3. We note that $\tilde{L}_t$ is only used in AdaUCB algorithm. The rewards and regrets are based on L_t, not $\tilde{L}_t$.

The AdaUCB algorithm adjusts the tradeoff between exploration and exploitation based on the load level L_t. Specifically, as shown in Algorithm 1, AdaUCB makes decisions based on the sum of the empirical reward (the exploitation term) $\bar{u}_k(t)$ and the confidence interval width (the exploration term). The latter term is proportional to $\sqrt{1 - \tilde{L}_t}$. In other words, AdaUCB uses an exploration factor $\alpha(1 - \tilde{L}_t)$ that is linearly decreasing in $\tilde{L}_t$. Thus, when the load level is high, the exploration term is relatively small and AdaUCB tends to emphasize exploitation, i.e., choosing the arms that perform well in the past. In contrast, when the load level is low, AdaUCB uses a larger exploration term and gives more opportunities to the arms with less explorations. Intuitively, with this load-awareness, AdaUCB explores more when the load is low and leverages the learned statistics to make better decisions when the load is high. Since the actual regret is scaled with the load level, AdaUCB can achieve an overall lower regret. Note that we have experimented a variety of load adaptation functions. The current one achieves superior empirical performance and is amenable to analyze, and thus adopted here.

4. Regret Analysis

Although the intuition behind AdaUCB is natural, the rigorous analysis of its regret is challenging. To analyze the decision in each slot, we require the statistics for the number of pulls of each arm. Unlike traditional regret analysis, we care about not only the upper bound, but also the lower bound for calculating the confidence level. However, even for fixed load levels, it is difficult to characterize the total number of pulls for suboptimal arms, i.e., obtaining tight lower and upper bounds for the regret. The gap between the lower and upper bounds makes it more difficult to evaluate the properties of UCB for general random load levels. To make the intuition more clear and analyses more readable, we start with the case of squared periodic wave load and Dirac rewards to illustrate the behavior of AdaUCB in Sec. 4.1. Then, we extend the results to the case with random binary-value load and random rewards in Sec. 4.2, and finally analyze the case with continuous load in Sec. 4.3.

Specifically, we first consider the case with binary-valued load, i.e., $L_t \in \{\epsilon_0, 1-\epsilon_1\}$, where $\epsilon_0, \epsilon_1 \in [0, 0.5)$. For this case, we let $l^{(-)} = \epsilon_0$ and $l^{(+)} = 1$. Then, $\tilde{L}_t = 0$ if $L_t = \epsilon_0$, and $\tilde{L}_t = \frac{1-\epsilon_0-\epsilon_1}{1-\epsilon_0} = 1 - \frac{\epsilon_1}{1-\epsilon_0}$ if $L_t = 1-\epsilon_1$. Therefore, the indices used by AdaUCB are given as follows:

$$\hat{u}_k(t) = \begin{cases} \bar{u}_k(t) + \sqrt{\frac{\alpha \log t}{C_k(t-1)}}, & \text{if } L_t = \epsilon_0, \\ \bar{u}_k(t) + \sqrt{\frac{\alpha \epsilon_1 \log t}{(1-\epsilon_0) C_k(t-1)}}, & \text{if } L_t = 1-\epsilon_1. \end{cases} \tag{5}$$

We investigate the regret of AdaUCB under the binary-valued load described above in Sec. 4.1 and Sec. 4.2, and then study its performance under continuous load in Sec. 4.3 with the insights obtained from the binary-valued load case.

4.1. AdaUCB under Periodic Square Wave Load and Dirac Rewards

We first study a simple case with periodic square wave load and Dirac rewards. In this scenario, the evolution of the system under AdaUCB is deterministic. The analysis of this deterministic system allows us to better understand AdaUCB and quantify the benefit of load-awareness. In addition, we focus on 2-armed bandits in analysis for easy illustration in this section.

Specifically, we assume the load is $L_t = \epsilon_0$ if t is even, and $1-\epsilon_1$ if t is odd. Moreover, the rewards are fixed, i.e., $X_{k,t} = u_k$ for all k and t, but unknown *a priori*. Without loss of generality, we assume arm-1 has higher reward, i.e., $1 \ge u_1 > u_2 \ge 0$, and let $\Delta = u_1 - u_2$ be the reward difference.

Under these settings, we can obtain the bounds for the number of pulls for each arm by borrowing the idea from (Salomon et al., 2011; 2013). The proofs of these results are included in Appendix A, which are similar to (Salomon et al., 2011; 2013), except for the effort of addressing the case of $L_t = 1 - \epsilon_1$.

We first characterize the upper and lower bounds on the total number of pulls for the suboptimal arm.

Lemma 1. *In the opportunistic bandit with periodic square wave load and Dirac rewards, the number of pulls for arm-2 under AdaUCB is bounded as follows:*
1) Upper bound for any $t \ge 1$: $C_2(t) \le \frac{\alpha logt}{\Delta^2} + 1$;
2) Lower bound for any $t = 2\tau \ge 2$:
$C_2(2\tau) \ge f(\tau) = \int_2^\tau \min(h'(s), 1) \mathrm{d}s - h(2)$, *where* $h(s) = \frac{\alpha \log s}{\Delta^2}\left(1 + \sqrt{\frac{2\alpha \log s}{(2s-1)\Delta^2}}\right)^{-2}$.

Note that $C_2(2\tau)$ provides the information for making decision in slot $2\tau + 1$, when $L_t = 1 - \epsilon_1$. With the lower bound in Lemma 1, we can show that after a certain time, AdaUCB will always pull the better arm when $L_t = 1 - \epsilon_1$ with the information provided by $C_2(2\tau)$. Combining with the upper bound on $C_2(t)$, we can obtain the regret bound for AdaUCB:

Theorem 1. *In the opportunistic bandit with periodic square wave load and Dirac rewards, the regret of AdaUCB is bounded as: $R_{\mathit{AdaUCB}}(T) \le \frac{\epsilon_0 \alpha \log T}{\Delta} + O(1)$.*

Remark 1: According to (Salomon et al., 2011), the regret of UCB(α) is lower bounded by $\frac{\alpha \log T}{\Delta}$ for fixed load $L_t = 1$. Without load-awareness, we can expect that the explorations occur roughly uniformly under different load levels. Thus, the regret of UCB(α) in this opportunistic bandit is roughly $\frac{\alpha(1+\epsilon_0-\epsilon_1)\log T}{2\Delta}$, and is much larger than the regret of AdaUCB for small ϵ_0 and ϵ_1. As an extreme case, when $\epsilon_0 = 0$, the regret of AdaUCB is $O(1)$, while that of UCB(α) is $O(\log T)$.

Remark 2: The above analysis provides us insights about the benefit of load-awareness in opportunistic bandits. With load-awareness, AdaUCB forces exploration to the slots with lower load and the information obtained there is sufficient to make good decisions in higher-load slots. Thus, the overall regret of AdaUCB is much smaller than traditional load-agnostic algorithms.

4.2. AdaUCB under Random Binary-Valued Load and Random Rewards

We now consider the more general case with random binary-valued load and random rewards. We assume that load $L_t \in \{\epsilon_0, 1-\epsilon_1\}$ and $\mathbb{P}\{L_t = \epsilon_0\} = \rho \in (0,1)$. We consider i.i.d random reward $X_{k,t} \in [0,1]$ and $\mathbb{E}[X_{k,t}] = u_k$, where $1 \geq u_1 > u_2 \geq u_3 \geq ... \geq u_K \geq 0$. Let $\Delta_k = u_1 - u_k$, and $\Delta^* = \min_{k>1} \Delta_k = \Delta_2$ be the minimum gap between the suboptimal arms and the optimal arm.

Compared with the deterministic case in Sec. 4.1, the analysis under random load and rewards is much more challenging. In particular, due to the reward randomness, the empirical value $\bar{u}_k(t)$ will deviate from its true value u_k. Unlike Dirac reward, this deviation could result in suboptimal decisions even when ϵ_0 and ϵ_1 are small. Thus, we need to carefully lower bound the number of pulls for each arm so that the deviation is bounded with high probability. We only provide sketches for the proofs here due to the space limit and refer readers to Appendix B for more detailed analyses.

We consider a larger α ($\alpha > 2$ in general, or larger when explicitly stated) for theoretical analysis purpose, similarly to earlier UCB papers such as (Auer et al., 2002). As we will see in the simulations, AdaUCB with $\alpha > 1/2$ works well under general random load.

We first propose a loose but useful bound for the number of pulls for the optimal arm. Let $C_k^{(0)}(t)$ be the number of slots where arm-k is pulled when $L_t = \epsilon_0$, i.e., $C_k^{(0)}(t) = \sum_{\tau=1}^{t} \mathbb{1}(L_\tau = \epsilon_0, a_\tau = k)$.

Lemma 2. *In the opportunistic bandit with random binary-valued load and random rewards, for a constant $\eta \in (0, \rho)$, there exists a constant T_2, such that under AdaUCB, for all $t \geq T_2$*

$$\begin{aligned}&\mathbb{P}\{C_1^{(0)}(t) < \frac{(\rho-\eta)t}{2}\} \\ \leq \quad & e^{-2\eta^2 t} + \frac{[2(K-1)]^{2\alpha-1}}{2\alpha-2}\big[(\rho-\eta)t\big]^{-2\alpha+2}.\end{aligned}$$

Sketch of Proof: The key intuition of proof is that when $C_1^{(0)}(t)$ is too small, the optimal arm will be pulled with high probability. Specifically, let $k' > 1$ be the index of arm that has been pulled for the most time among the suboptimal arms before t, and $t' < t$ be the last slot when k' is pulled under load $L_t = \epsilon_0$ for the last time. If $C_1^{(0)}(t) < \frac{(\rho-\eta)t}{2}$, then $C_{k'}(t'-1) \geq C_{k'}^{(0)}(t'-1) = \Theta(t)$ with high probability. Using the fact that $\frac{\log t}{t} \to 0$ as $t \to \infty$, we know there exists a constant T_2 such that for $t \geq T_2$, the confidence width $\sqrt{\frac{\alpha \log t'}{C_{k'}(t'-1)}}$ will be sufficiently small compared with the minimum gap $\Delta^* \leq \Delta_k$. Moreover, the algorithm will pull the best arm when the UCB deviation is sufficiently small. Then, we can bound the probability of the event $C_1^{(0)}(t) < \frac{(\rho-\eta)t}{2}$ by bounding the deviation of UCBs.

Next we bound the total number of pulls of the suboptimal arm as follows.

Lemma 3. *In the opportunistic bandit with random binary-valued load and random rewards, under AdaUCB, we have*

$$\mathbb{E}[C_k(T)] \leq \frac{4\alpha \log T}{\Delta_k^2} + O(1), 1 < k \leq K. \qquad (6)$$

Sketch of Proof: To prove this lemma, we discuss the slots when the suboptimal arm is pulled under low and high load levels, respectively. When the load is low, i.e., $L_t = \epsilon_0$, AdaUCB becomes UCB(α) and thus we can bound the probability of pulling the suboptimal arm similarly to (Auer et al., 2002). When the load is high, i.e., $L_t = 1-\epsilon_1$, the index becomes $\hat{u}_k(t) = \bar{u}_k(t) + \sqrt{\frac{\alpha\epsilon_1 \log t}{(1-\epsilon_0)C_k(t-1)}}$. In this case, with high probability, the index of the optimal arm is lower bounded by $u_1 - \left(1 - \sqrt{\frac{\epsilon_1}{(1-\epsilon_0)}}\right)\sqrt{\frac{\alpha \log t}{C_1(t-1)}}$ according to Lemma 2. With similar adjustment on the UCB index for the suboptimal arm, we can bound the probability of pulling the suboptimal arm under high load. The conclusion of the lemma then follows by combining the above two cases.

Now we further lower bound the pulls of the suboptimal arm with high probability.

Lemma 4. *In the opportunistic bandit with random binary-valued load and random rewards, for a positive number $\delta \in (0,1)$, we have for any $k > 1$,*

$$\begin{aligned}&\mathbb{P}\Big\{C_k(t) < \frac{\alpha \log t}{4(\Delta_k+\delta)^2}\Big\} \\ = \quad & O\big(t^{-(2\alpha-3)} + t^{-(2\alpha(\frac{1-\delta}{2-\delta})^2-2)}\big).\end{aligned}$$

Sketch of Proof: Although the analysis is more difficult, the intuition of proving this lemma is similar to that of Lemma 2: if $C_k(t)$ is too small at a certain slot, then we will pull the suboptimal arm instead of the optimal arm with high probability. To be more specific, we focus on the slot t' when the optimal arm is pulled for the last time before t under load $L_t = \epsilon_0$. According to Lemma 2, $C_1(t) \geq C_1^{(0)}(t) \geq \frac{(\rho-\eta)t}{2}$ with high probability, indicating $t' \geq (\rho-\eta)t/2$ with high probability. Moreover, the index for the optimal arm $\hat{u}_1(t') \leq u_1 + \delta$ with high probability for a sufficiently large t', because $\sqrt{\frac{\log t}{t}} \to 0$ as $t \to \infty$. On the other hand, we can show that for the suboptimal arm, $\hat{u}_k(t') > u_1 + \delta = u_k + (\Delta_k + \delta)$ with high probability when $C_k(t'-1) < \frac{\alpha \log t}{4(\Delta_k+\delta)^2}$. Thus, the probability of pulling the optimal arm at t' is bounded by a small value, implying the conclusion of the lemma.

Using the above lemmas, now we can further refine the upper bound on the regret of AdaUCB and show that AdaUCB achieves smaller regret than traditional UCB.

Theorem 2. *Using AdaUCB in the opportunistic bandit with random binary-valued load and random rewards, if* $\alpha > 16$ *and* $\sqrt{\frac{\epsilon_1}{1-\epsilon_0}} < \frac{1}{8}$*, we have*

$$R_{\mathit{AdaUCB}}(T) \leq 4\epsilon_0 \alpha \log T \sum_{k>1} \frac{1}{\Delta_k} + O(1). \qquad (7)$$

Sketch of Proof: The key idea of the proof is to find an appropriate $\delta \in (0, \Delta^*)$, such that $\alpha > 16(1+\frac{\delta}{\Delta^*})^2$ and $\sqrt{\frac{\epsilon_1}{1-\epsilon_0}} < \frac{\Delta^*}{8(\Delta^*+\delta)}$. In fact, the existence of this δ is guaranteed under the assumptions $\alpha > 16$ and $\sqrt{\frac{\epsilon_1}{1-\epsilon_0}} < \frac{1}{8}$. Using this δ, we can then use Lemma 4 to bound the probability of pulling the suboptimal arm when the load is high. This indicates most explorations occur when the load is low, i.e., $L_t = \epsilon_0$. The conclusion of this theorem then follows according to Lemma 3.

Remark 3: Although there is no tight lower bound for the regret of UCB(α), we know that for traditional (load-oblivious) bandit algorithms, $\mathbb{E}[C_k(T)]$ is lower bounded by $\frac{\log T}{KL(u_k,u_1)}$ (Lai & Robbins, 1985) for large T, where $KL(u_k, u_1)$ is the Kullback-Leibler divergence. Without load-awareness, the regret will be roughly lower bounded by $\frac{(1-\epsilon_0-\epsilon_1)\log T}{2} \sum_{k>1} \frac{\Delta_k}{KL(u_k,u_1)}$. In contrast, with load-awareness, AdaUCB can achieve much lower regret than load-oblivious algorithms, when the load fluctuation is large, i.e., ϵ_0 and ϵ_1 are small.

Theorem 2 directly implies the following result.

Corollary 1. *Using AdaUCB in the opportunistic bandit with random reward under i.i.d. random binary load where* $\epsilon_0 = 0$*, if* $\alpha > 16$ *and* $\epsilon_1 < \frac{\sqrt{2}}{4}$*, we have* $R_{\mathit{AdaUCB}}(T) = O(1)$.

Remark 4: We note that this $O(1)$ bound is in the sense of expected regret, which is different from the high probability $O(1)$ regret bound (Abbasi-Yadkori et al., 2011). Specifically, while the opportunistic bandits can model the whole spectrum of load-dependent regret, Corollary 1 highlights one end of the spectrum where there are "free" learning opportunities. In this case, we push most explorations to the "free" exploration slots and result in an $O(1)$ expected regret. Note that even under "free" exploration, we assume here that the value of the arms can be observed as discussed in Sec. 2.

It is worth noting that there are realistic scenarios where the exploration cost of a suboptimal arm is zero or close to zero. Consider the network configuration case where we use throughput as the reward. In this case, $X_{a_t,t}$ is the percentage of the peak load that configuration a_t can handle. Because of the dummy low-priority traffic injected into the network, we can learn the true value of $X_{a_t,t}$ under the peak load. At the same time, configuration a_t, although suboptimal, may completely satisfy the real load L_t because it is high priority and thus served first. Therefore, although a suboptimal arm, a_t sacrifices no throughput on the real load L_t, and thus generates a real regret of zero. In other words, even if the system load is always positive, the chance of zero regret under a suboptimal arm is greater than zero, and in practice, can be non-negligible. To capture this effect, we can modify the regret defined in Eq. (1) by replacing L_t with 0 when L_t is smaller than a threshold.

Last, we note that, under the condition of Corollary 1, it is easy to design other heuristic algorithms that can perform well. For example, one can do round-robin exploration when the load is zero and chooses the best arm when the load is non-zero. However, such naive strategies are difficult to extend to more general cases. In contrast, AdaUCB applies to a wide range of situations, with both theoretical performance guarantees and desirable empirical performance.

Dependence on ρ: In the regret analysis, we focus on the asymptotic behavior of the regret as T goes to infinity. In the bound, the constant term contains the impact of other factors, in particular the ratio of low load ρ, as shown in Appendix B.5. From the analysis, one can see that the constant term increases as $\rho \to 0$. It suggests that one should use the traditional UCB when ρ is small because there exists little load fluctuation. In practice, AdaUCB achieves much smaller regret than traditional UCB and TS algorithms, even for small values of ρ such as $\rho = 0.05$ under binary load and $\rho = 0.001$ under continuous load. Such analysis and evaluations establish guidelines on when to use UCB or AdaUCB. More discussions can be found in Appendix D.

4.3. AdaUCB under Continuous Load

Inspired by the insights obtained from the binary-valued load case, we discuss AdaUCB in opportunistic bandits under continuous load in this section.

Selection of truncation thresholds. When the load is continuous, we need to choose appropriate $l^{(-)}$ and $l^{(+)}$ for AdaUCB. We first assume that the load distribution is *a priori* known, and discuss how to choose the thresholds under unknown load distribution later. The analysis under binary-valued load indicates that, the explorations mainly occur in low load slots. To guarantee sufficient explorations for a logarithmic regret, we propose to select the thresholds such that:

- The lower threshold $l^{(-)}$ satisfies $\mathbb{P}\{L_t \leq l^{(-)}\} = \rho > 0$;
- The upper threshold $l^{(+)} \geq l^{(-)}$.

In the special case of $l^{(+)} = l^{(-)}$, we redefine the normalized load $\tilde{L}_t$ in (4) as $\tilde{L}_t = 0$ when $L_t \leq l^{(-)}$ and $\tilde{L}_t = 1$ when $L_t > l^{(-)}$.

Regret analysis. Under continuous load, it is hard to obtain regret bound as that in Theorem 2 for general $l^{(-)}$ and $l^{(+)}$ chosen above. Instead, we first show logarithmic regret for general $l^{(-)}$ and $l^{(+)}$, and then illustrate the advantages of AdaUCB for the special case with $l^{(-)} = l^{(+)}$.

First, we show that AdaUCB with appropriate truncation thresholds achieves logarithmic regret as below. This lemma is similar to Lemma 3, and the detailed outline of proof can be found in Appendix C.

Lemma 5. *In the opportunistic bandit with random continuous load and random rewards, under AdaUCB with $\mathbb{P}\{L_t \leq l^{(-)}\} = \rho > 0$ and $l^{(+)} \geq l^{(-)}$, we have*

$$\mathbb{E}[C_k(T)] \leq \frac{4\alpha \log T}{\Delta_k^2} + O(1). \tag{8}$$

Next, we illustrate the advantages of AdaUCB under continuous load by studying the regret bound for AdaUCB with special thresholds $l^{(+)} = l^{(-)}$.

Theorem 3. *In the opportunistic bandit with random continuous load and random rewards, under AdaUCB with $\mathbb{P}\{L_t \leq l^{(-)}\} = \rho > 0$ and $l^{(+)} = l^{(-)}$, we have*

$$R_{AdaUCB}(T) \leq 4\alpha \log T \mathbb{E}[L_t | L_t \leq l^{(-)}] \sum_{k>1} \frac{1}{\Delta_k} + O(1), \tag{9}$$

where $\mathbb{E}[L_t | L_t \leq l^{(-)}]$ is the expectation of L_t conditioned on $L_t \leq l^{(-)}$.

Sketch of Proof: Recall that for this special case $l^{(+)} = l^{(-)}$, we let $\tilde{L}_t = 0$ for $L_t \leq l^{(-)}$ and $\tilde{L}_t = 1$ for $L_t > l^{(+)}$. Then we can prove the theorem analogically to the proof of Theorem 2 for the binary-valued case. Specially, when $L_t \leq l^{(-)}$, we have $\tilde{L}_t = 0$ and it corresponds to the case of $L_t = \epsilon_0$ ($\tilde{L}_t = 0$) in the binary-valued load case. Similarly, the case of $L_t > l^{(+)}$ ($\tilde{L}_t = 1$) corresponds to the case of $L_t = 1 - \epsilon_1$ under binary-valued load with $\epsilon_1 = 0$. Then, we can obtain results similar to Lemma 4 and thus show that the regret under load $L_t > l^{(+)}$ is $O(1)$. Furthermore, the number of pulls under load level $L_t \leq l^{(-)}$ is bounded according to Lemma 5. The conclusion of the theorem then follows by using the fact that all load below $l^{(-)}$ are treated the same by AdaUCB, i.e., $\tilde{L}_t = 0$ for all $L_t \leq l^{(-)}$.

Remark 5: We compare the regret of AdaUCB and conventional bandit algorithms by an example, where the load level L_t is uniformly distributed in $[0, 1]$. In this simple example, the regret of AdaUCB with thresholds $l^{(+)} = l^{(-)}$ is bounded by $R_{\text{AdaUCB}}(T) \leq 4\alpha \log T \sum_{k>1} \frac{1}{\Delta_k} \cdot \frac{\rho}{2} + O(1)$, since $\mathbb{E}[L_t | L_t \leq l^{(-)}] = \rho/2$ and $\mathbb{E}[L_t | L_t > l^{(-)}] < 1$. However, for any load-oblivious bandit algorithm such as UCB(α) , the regret is lower bounded by $\log T \sum_{k>1} \frac{\Delta_k}{KL(u_k, u_1)} \cdot \frac{1}{2} + O(1)$. Thus, AdaUCB achieves much smaller regret when T is large and ρ is relatively small.

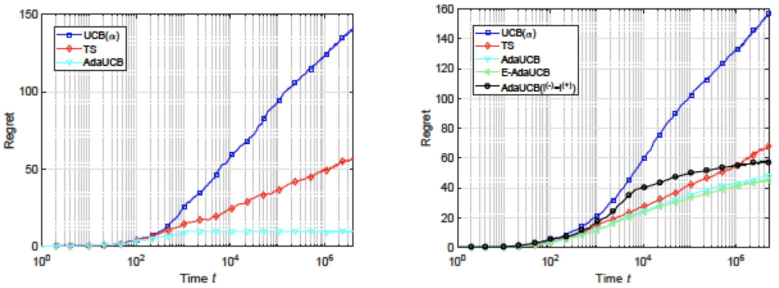
Remark 6: From the above analysis, we can see that the selection of $l^{(+)}$ does not affect the order of the regret ($O(\log T)$). However, for a fixed $l^{(-)}$, we can further adjust $l^{(+)}$ to control the explorations for the load in the range of $(l^{(-)}, l^{(+)})$. Specifically, with a larger $l^{(+)}$, more explorations happen under the load between $l^{(-)}$ and $l^{(+)}$. These explorations accelerate the learning speed but may increase the long term regret because we allow more explorations under load $l^{(-)} < L_t < l^{(+)}$. The behavior is opposite if we use a smaller $l^{(+)}$. In addition, appropriately chosen thresholds also handle the case when the load has little or no fluctuation, i.e., $L_t \approx c$. For example, if we set $l^{(-)} = c$ and $l^{(+)} = 2c$, AdaUCB degenerates to UCB(α).

E-AdaUCB. In practice, the load distribution may be unknown *a priori* and may change over time. To address this issue, we propose a variant, named Empirical-AdaUCB (E-AdaUCB), which adjusts the thresholds $l^{(-)}$ and $l^{(+)}$ based on the empirical load distribution. Specifically, the algorithm maintains the histogram for the load levels (or its moving average version for non-stationary cases), and then select $l^{(-)}$ and $l^{(+)}$ accordingly. For example, we can select $l^{(-)}$ and $l^{(+)}$ such that the empirical probability $\tilde{\mathbb{P}}\{L_t \leq l^{(-)}\} = \tilde{\mathbb{P}}\{L_t \geq l^{(+)}\} = 0.05$. We can see that, in most simulations, E-AdaUCB performs closely to AdaUCB with thresholds chosen offline.

5. Experiments

In this section, we evaluate the performance of AdaUCB using both synthetic data and real-world traces. We use the classic UCB(α) and TS (Thompson Sampling) algorithms as comparison baselines. In both AdaUCB and UCB(α), we set α as $\alpha = 0.51$, which is close to $1/2$ and performs better than a larger α. We note that the gap between AdaUCB and the classic UCB(α) clearly demonstrates the impact of opportunistic learning. On the other hand, TS is one of the most popular and robust bandit algorithms applied to a wide range of application scenarios. So we apply it here as a reference. However, because AdaUCB and TS (or other bandit algorithms) improve UCB on different fronts, so their comparison does not clearly show the impact of opportunistic bandit.

AdaUCB under synthetic scenarios. We consider a 5-armed bandit with Bernoulli rewards, where the expected reward vector is $[0.05, 0.1, 0.15, 0.2, 0.25]$. Fig. 1(a) shows the regrets for different algorithms under random binary-value load with $\epsilon_0 = \epsilon_1 = 0$ and $\rho = 0.5$. AdaUCB significantly reduces the regret in opportunistic bandits.



(a) Binary-valued load

(b) Beta distributed load

Figure 1. Regret under Synthetic Scenarios. In (a), $\epsilon_0 = \epsilon_1 = 0, \rho = 0.5$. In (b), for AdaUCB, $l^{(\ )} = l^{(\ )}_{0.05}, l^{(+)} = l^{(+)}_{0.05}$; for AdaUCB($l^{(\ )} = l^{(+)}$), $l^{(\ )} = l^{(+)} = l^{(\ )}_{0.05}$.

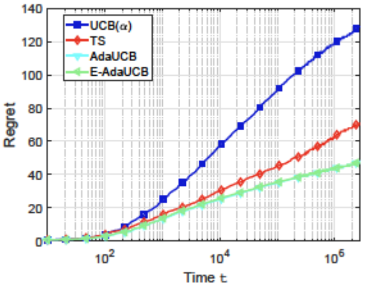


Figure 2. Regret in MVNO systems.

Specifically, the exploration cost in this case can be zero and AdaUCB achieves $O(1)$ regret. For continuous load, Fig. 1(b) shows the regrets for different algorithms with beta distributed load. AdaUCB still outperforms the UCB(α) or TS algorithms. Here, we define $l_\rho^{(-)}$ as the lower threshold such that $\mathbb{P}\{L_t \leq l_\rho^{(-)}\} = \rho$, and $l_\rho^{(+)}$ as the upper threshold such that $\mathbb{P}\{L_t \geq l_\rho^{(+)}\} = \rho$. These simulation results demonstrate that, with appropriately chosen parameters, the proposed AdaUCB and E-AdaUCB algorithms achieve good performance by leveraging the load fluctuation in opportunistic bandits. As a special case, with a single threshold $l^{(+)} = l^{(-)} = l_{0.05}^{(-)}$, AdaUCB still outperforms UCB(α) and TS, although it may have higher regret at the beginning. More simulation results can be found in Appendix D.1, where we study the impact of environment and algorithm parameters such as load fluctuation and the thresholds for load truncation. In particular, the results show that AdaUCB works well in continuous load when ρ is very small.

AdaUCB applied in MVNO systems. We now evaluate the proposed algorithms using real-world traces. In an MVNO (Mobile Virtual Network Operator) system, a virtual operator, such as Google Fi (Project Fi, *https://fi.google.com*), provides services to users by leasing network resources from real mobile operators. In such a system, the virtual operator would like to provide its users high quality service by accessing the network resources of the real operator with the best network performance. Therefore, we view each real mobile operator as an arm, and the quality of user experienced on that operator network as the reward. We use experiment data from Speedometer (Speedometer, *https://storage.cloud.google.com/speedometer*) and another anonymous operator to conduct the evaluation. More details about the MVNO system can be found in Appendix D.2. Here, using insights obtained from simulations based on the synthetic data, we choose $l^{(-)}$ and $l^{(+)}$ such that $\mathbb{P}\{L_t \leq l^{(-)}\} = \mathbb{P}\{L_t \geq l^{(+)}\} = 0.05$. As shown in Fig. 2, the regret of AdaUCB is only about 1/3 of UCB(α), and the performance of E-AdaUCB is indistinguishable from that of AdaUCB. This experiment demonstrates the effectiveness of AdaUCB and E-AdaUCB in practical situations, where the load and the reward are continuous and are possibly non-stationary. It also demonstrates the practicality of E-AdaUCB without ***a priori*** load distribution information.

6. Conclusions and Future Work

In this paper we study opportunistic bandits where the regret of pulling a suboptimal arm depends on external conditions such as traffic load or produce price. We propose AdaUCB that opportunistically chooses between exploration and exploitation based on the load level, i.e., taking the slots with low load level as opportunities for more explorations. We analyze the regret of AdaUCB, and show that AdaUCB can achieve provable lower regret than the traditional UCB algorithm, and even $O(1)$ regret with respect to time horizon T, under certain conditions. Experimental results based on both synthetic and real data demonstrate the significant benefits of opportunistic exploration under large load fluctuations.

This work is a first attempt to study opportunistic bandits, and several open questions remain. First, although AdaUCB achieves promising experimental performance under general settings, rigorous analysis with tighter performance bound remains challenging. Furthermore, opportunistic TS-type algorithms are also interesting because TS-type algorithms often performs better than UCB-type algorithms in practice. Last, we hope to investigate more general relations between the load and actual reward.

Acknowledgements

This research was supported in part by NSF Grants CCF-1423542, CNS-1547461, CNS-1718901. The authors would like to thank Prof. Peter Auer (University of Leoben) for his helpful suggestions, and the reviewers for their valuable feedback.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

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

Besbes, O., Gur, Y., and Zeevi, A. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems*, pp. 199–207, 2014.

Brandt, J. A. and Bessler, D. A. Price forecasting and evaluation: An application in agriculture. *Journal of Forecasting*, 2(3):237–248, 1983.

Bubeck, S. *Bandits games and clustering foundations*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I, 2010.

Bubeck, S. and Cesa-Bianchi, N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

Chu, W., Li, L., Reyzin, L., and Schapire, R. E. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Garivier, A. and Moulines, E. On upper-confidence bound policies for non-stationary bandit problems. In *International Conference on Algorithmic Learning Theory*, 2011.

Lai, T. L. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pp. 1091–1114, 1987.

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

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *ACM International Conference on World Wide Web (WWW)*, pp. 661–670, 2010.

Press, W. H. Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *Proceedings of the National Academy of Sciences*, 106(52):22387–22392, 2009.

Project Fi, *https://fi.google.com*.

Salomon, A., Audibert, J.-Y., and Alaoui, I. E. Regret lower bounds and extended upper confidence bounds policies in stochastic multi-armed bandit problem. *arXiv preprint arXiv:1112.3827*, 2011.

Salomon, A., Audibert, J.-Y., and Alaoui, I. E. Lower bounds and selectivity of weak-consistent policies in stochastic multi-armed bandit problem. *Journal of Machine Learning Research*, 14(Jan):187–207, 2013.

Speedometer, *https://storage.cloud.google.com/speedometer*.

Villar, S. S., Bowden, J., Wason, J., et al. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical Science*, 30(2):199–215, 2015.

Walraevens, J., Steyaert, B., and Bruneel, H. Performance analysis of a single-server atm queue with a priority scheduling. *Computers & Operations Research*, 30(12): 1807 – 1829, 2003.

Wu, H., Srikant, R., Liu, X., and Jiang, C. Algorithms with logarithmic or sublinear regret for constrained contextual bandits. In *The 29th Annual Conference on Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec. 2015.

Zhou, L. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.