AdaLinUCB: Opportunistic Learning for Contextual Bandits

Xueying Guo , **Xiaoxiao Wang** and **Xin Liu**
University of California, Davis
guoxueying@outlook.com, {xxwa, xinliu}@ucdavis.edu

Abstract

In this paper, we propose and study opportunistic contextual bandits - a special case of contextual bandits where the exploration cost varies under different environmental conditions, such as network load or return variation in recommendations. When the exploration cost is low, so is the actual regret of pulling a sub-optimal arm (e.g., trying a suboptimal recommendation). Therefore, intuitively, we could explore more when the exploration cost is relatively low and exploit more when the exploration cost is relatively high. Inspired by this intuition, for opportunistic contextual bandits with Linear payoffs, we propose an Adaptive Upper-Confidence-Bound algorithm (AdaLinUCB) to adaptively balance the exploration-exploitation trade-off for opportunistic learning. We prove that AdaLinUCB achieves $O\left((\log T)^2\right)$ problem-dependent regret upper bound, which has a smaller coefficient than that of the traditional LinUCB algorithm. Moreover, based on both synthetic and real-world dataset, we show that AdaLinUCB significantly outperforms other contextual bandit algorithms, under large exploration cost fluctuations.[1]

1 Introduction

In sequential decision making problems such as contextual bandits [Auer, 2002; Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011; Langford and Zhang, 2008], there exists an intrinsic trade-off between exploration (of unknown environment) and exploitation (of current knowledge). Existing algorithm design focuses on how to balance such a trade-off appropriately under the implicit assumption that the exploration cost remains the same over time. However, in a variety of application scenarios, the exploration cost is time varying and situation-dependent. Such scenarios present an opportunity to explore more when the exploration cost is relatively low and exploit more when that cost is high, thus adaptively balancing the exploration-exploitation trade-off to reduce the overall regret. Consider the following motivating examples.

[1]The supplementary material of this paper is available at: https://github.com/xiaoxiao01/IJCAI19/blob/master/Supplementary.pdf

Motivating scenario 1: return variation in recommendations. Contextual bandits have been widely used in recommendation systems [Li *et al.*, 2010]. In such scenarios, the candidate articles/products to be recommended are considered as the arms, the features of users as the context, and the click-through rate as the reward (i.e., the probability that a user accepts the recommendation). However, note that the monetary return of a recommendation (if accepted) can differ depending on 1) timing (e.g., holiday vs. non-holiday season) and 2) users with different levels of purchasing power or loyalty (e.g., diamond vs. silver status). Because the ultimate goal is to maximize the overall monetary reward, intuitively, when the monetary return of a recommendation (if accepted) is low, the monetary regret of pulling a suboptimal arm is low, leading to a low exploration cost, and correspondingly, high returns lead to high regret and high exploration cost.

Motivating scenario 2: load variation for network configuration. In computer networks, there are a number of parameters that can be configured and have a large impact on overall network performance. For example, in cellular networks, a cell tower can configure transmission power, radio spectrum, antenna, etc., that can affect network performance such as coverage, throughput, and quality of service. Contextual bandit can be applied in network configuration [Chuai *et al.*, 2019]. In such problems, the goal of network configuration can be improving network performance for peak load scenario. In such a scenario, a possible configuration of a cellular base station can be considered as an arm, the characteristics of the cell station such as coverage area as the context, and network performance such as throughput as reward. However, network traffic load fluctuates over time, and thus the actual regret of using a suboptimal configuration varies accordingly. Specifically, when the network load is low, dummy traffic can be injected into the network so that the total load (real plus dummy load) is the same as the peak load. In this manner, we can seek the optimal configuration under the peak load even in off-peak hours. Meanwhile, the regret of using a suboptimal configuration is low since the real load affected is low. In practice, the priority of the dummy traffic can be set to be lower than that of the real traffic. Because the network handles high priority traffic first, low priority traffic has little or no impact on the high priority traffic[Walraevens *et al.*, 2003]. Thus, the regret on the actual load can be further reduced, leading to a low or even negligible exploration cost.

Opportunistic Contextual Bandits. Motivated by these application scenarios, we study opportunistic contextual bandits in this paper, focusing on the contextual bandit setting with linear payoffs. Specifically, we define *opportunistic contextual bandit* as a contextual bandit problem with the following characteristic: 1) The exploration cost (regret) of selecting a suboptimal arm varies depending on a time-varying external factor that we called the variation factor. 2) The variation factor is revealed first so that the learning agent can decide which arm to pull depending on this variation factor. As suggested by its name, in opportunistic contextual bandits, the variation of this external variation factor can be leveraged to reduce the actual regret. Further, besides the previous two examples, opportunistic contextual bandit algorithms can be applied to other scenarios that share these characteristics. We also note that this can be considered as a special case of contextual bandits, by regarding the variation factor as part of context. However, the general contextual bandit algorithms do not take advantage of the opportunistic nature of the problem, and can lead to a less competitive performance.

Contributions. In this paper, we propose an Adaptive Upper-Confidence-Bound algorithm for opportunistic contextual bandits with Linear payoffs (AdaLinUCB). The algorithm is designed to dynamically balance the exploration-exploitation trade-off in opportunistic contextual bandits. To be best of our knowledge, this is the first work to study opportunistic learning for contextual bandits. We focus on the problem-dependent bound analysis here, which is a setting that allows a better bound to be achieved under stronger assumptions. To the best of our knowledge, such a bound does not exist for LinUCB in the existing literature. In this paper, we prove problem-dependent bounds for both the proposed AdaLinUCB and the traditional LinUCB algorithms. Both algorithms have a regret upper bound of $O\left((\log T)^2\right)$, and the coefficient of the AdaLinUCB bound is smaller than that of LinUCB. Furthermore, using both synthetic and real-world large-scale dataset, we show that AdaLinUCB significantly outperforms other contextual bandit algorithms, under large exploration cost fluctuations.

2 Related Work

Contextual bandit algorithms have been applied to many real applications, such as display advertising [Li *et al.*, 2011] and content recommendation [Li *et al.*, 2010; Bouneffouf *et al.*, 2012]. In contrast to the classic K-arm bandit problem [Auer *et al.*, 2002; Chapelle and Li, 2011; Agrawal, 1995], side information called context is provided in contextual bandit problem before arm selection [Auer, 2002; Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011; Langford and Zhang, 2008]. The contextual bandits with linear payoffs was first introduced in [Auer, 2002]. In [Li *et al.*, 2010], LinUCB algorithm is introduced based on the "optimism in the face of Uncertainty" principal for linear bandits. The LinUCB algorithm and its variances are reported to be effective in real application scenarios [Li *et al.*, 2010; Wu *et al.*, 2016; Wang *et al.*, 2016; Wang *et al.*, 2017]. Compared to the classic K-armed bandits, the contextual bandits achieves superior performance in various application scenarios [Filippi *et al.*, 2010].

Although LinUCB is effective and widely applied, its analysis is challenging. In the initial analysis effort [Chu *et al.*, 2011], instead of analyzing LinUCB, it presents an $O(\sqrt{T\ln^3(T)})$ regret bound for a modified version of LinUCB. The modification is needed to satisfy the independent requirement by applying Azuma/Hoeffding inequality. In another line of analysis effort, the authors in [Abbasi-Yadkori *et al.*, 2011] design another algorithm for contextual bandits with linear payoffs and provide its regret analysis without independent requirement. Although the algorithm proposed in [Abbasi-Yadkori *et al.*, 2011] is different from LinUCB and suffers from a higher computational complexity, the analysis techniques are helpful.

The opportunistic learning has been introduced in [Wu *et al.*, 2018] for classic K-armed bandits. However, we note that opportunistic learning exists for any sequential decision making problem. In [Bouneffouf *et al.*, 2012], the authors study into contextual bandits with HLCS (High-Level Critical Situations) set, and proposes a contextual-ϵ-greedy policy, a policy that has an opportunistic nature since the ϵ (exploration level) is adaptively adjusted based on the similarity to HLCSs (importance level). However, it only introduces a heuristic algorithm, and does not present a clearly formulation of opportunistic learning. Furthermore, the policy design in [Bouneffouf *et al.*, 2012] implicitly makes the assumption that the contexts in HLCS have already been explored sufficiently beforehand, which is not a cold-start problem. To the best of our knowledge, no prior work has made formal mathematical formulation and rigorous performance analysis for opportunistic contextual bandits.

The opportunistic linear contextual bandits can be regarded as a special case of non-linear contextual bandits. However, general contextual bandit algorithms such as KernelUCB [Valko *et al.*, 2013] do not take advantage of the opportunistic nature of the problem, and thus can lead to a less competitive performance, as shown in Appendix E.3 of the supplementary material for more details. Moreover, KernelUCB suffers from the sensitivity to hyper-parameter tuning, and the extremely high computational complexity for even moderately large dataset, which limits its application in real problems.

3 System Model

We use the following notation conventions. We use $\|x\|_2$ to denote the 2-norm of a vector $x \in \mathbb{R}^d$. For a positive-definite matrix $A \in \mathbb{R}^{d\times d}$, the weighted 2-norm of vector $x \in \mathbb{R}^d$ is defined by $\|x\|_A = \sqrt{x^\top A x}$. The inner product of vectors is denoted by $\langle\cdot,\cdot\rangle$, that is, $\langle x, y\rangle = x^\top y$. Denote by $\lambda_{\min}(A)$ the minimum eigenvalue of a positive-definite matrix A. Denote by $\det(A)$ the determinant of matrix A. Denote by $\mathrm{trace}(A)$ the trace of matrix A.

Now, we present system model. We first introduce the setting of a standard linear contextual bandit problem. The time is slotted. In each time slot t, there exists a set of possible arms, denoted by set $\mathcal{D}_t$. For each arm $a \in \mathcal{D}_t$, there is an associated context vector $x_{t,a} \in \mathbb{R}^d$, and a nominal reward $r_{t,a}$. In each slot t, the learner can observe context vectors of all possible arms, and then choose an arm a_t and receive the corresponding nominal reward r_{t,a_t}. Note that only the

nominal reward of the chosen arm is revealed for the learner in each time slot t. Further, the nominal rewards of arms are assumed to be a noisy version of an unknown linear function of the context vectors. Specifically, $r_{t,a} = \langle x_{t,a}, \theta_\star \rangle + \eta_t$, where $\theta_\star \in \mathbb{R}^d$ is an unknown parameter, and η_t is a random noise with zero mean, i.e., $\mathbb{E}[\eta_t | x_{t,a_t}, \mathcal{H}_{t-1}] = 0$, with $\mathcal{H}_{t-1} = (x_{1,a_1}, \eta_1, \cdots, x_{t-1,a_{t-1}}, \eta_{t-1})$ representing historical observations.

The goal of a standard contextual bandit problem is to minimize the total regret in T slots, in terms of the nominal rewards. Particularly, the accumulated T-slot regret regarding nominal reward is defined as,

$$\mathbf{R}_{\text{total}}(T) = \sum_{t=1}^{T} R_t = \sum_{t=1}^{T} \mathbb{E}[r_{t,a_t^\star} - r_{t,a_t}], \tag{1}$$

where R_t is the one-slot regret regarding nominal reward for time slot t, $a_t^\star$ is the optimal arm at time slot t. Here, the optimal arm is the one with the largest expected reward, i.e., $a_t^\star = \arg\max_{a \in \mathcal{D}_t} \mathbb{E}[r_{t,a}]$. To simplify the notation, we denote $r_{t,\star} = r_{t,a_t^\star}$ in the following. That is, $r_{t,\star}$ is the optimal nominal reward at slot t.

In the **opportunistic learning environment**, let L_t be an external variation factor for time slot t. The **actual reward** $\tilde{r}_{t,a}$ that the agent receives has the following relationship with the **nominal reward**:

$$\tilde{r}_{t,a} = L_t r_{t,a}, \forall t, \forall a \in \mathcal{D}_t.$$

At each time slot, the learner first observes the context vectors associated with all possible arms, i.e., $x_{t,a}, \forall a \in \mathcal{D}_t$, as well as the current value of L_t. Based on which the learner selects current arm a_t, observes a nominal reward r_{t,a_t}, and receives the actual reward $\tilde{r}_{t,a} = L_t r_{t,a}$.

This model captures the essence of the opportunistic contextual bandits. For example, in the recommendation scenario, and L_t can be a seasonality factor, which captures the general purchase rate in current season. Or L_t can be purchasing power (based on historical information) or loyalty level of users (e.g., diamond vs. silver status). In the network configuration example, when the nominal reward $r_{t,a}$ captures the impact of a configuration at the peak load, the total load (the dummy load plus the real load) resembles the peak load. Then, L_t can be the amount of real load, and thus the actual reward is modulated by L_t as $L_t r_{t,a}$.

The goal of the learner is to minimize the total regret in T slots, in terms of the actual rewards. Particularly, the accumulated T-slot regret regarding actual reward is defined as,

$$\tilde{\mathbf{R}}_{\text{total}}(T) = \sum_{t=1}^{T} \mathbb{E}[R_t L_t] = \sum_{t=1}^{T} \mathbb{E}[L_t r_{t,\star} - L_t r_{t,a_t}]. \tag{2}$$

In a special case, equation (2) has an equivalent form: when L_t is i.i.d. over time with mean value $\bar{L}$ and r_{t,a_t} is independent of L_t conditioned on a_t, the total regret regarding actual reward is $\tilde{\mathbf{R}}_{\text{total}}(T) = \bar{L} \sum_{t=1}^{T} \mathbb{E}[r_{t,\star}] - \sum_{t=1}^{T} \mathbb{E}[L_t r_{t,a_t}]$. Note that in general, it is likely that $\mathbb{E}[L_t r_{t,a_t}] \neq \bar{L}\mathbb{E}[r_{t,a_t}]$, because the action a_t can depend on L_t.

Algorithm 1 AdaLinUCB

1: Inputs: $\alpha \in \mathbb{R}_+$, $d \in \mathbb{N}$, $l^{(+)}$, $l^{(-)}$.
2: $A \leftarrow \boldsymbol{I}_d$ {The d-by-d identity matrix}
3: $b \leftarrow \mathbf{0}_d$
4: **for** $t = 1, 2, 3, \cdots, T$ **do**
5: $\quad \theta_{t-1} = A^{-1} b$
6: $\quad$ Observe possible arm set $\mathcal{D}_t$, and observe associated context vectors $x_{t,a}, \forall a \in \mathcal{D}_t$.
7: $\quad$ Observe L_t and calculate $\tilde{L}_t$ by (3).
8: $\quad$ **for** $a \in \mathcal{D}_t$ **do**
9: $\quad\quad p_{t,a} \leftarrow \theta_{t-1}^\top x_{t,a} + \alpha \sqrt{(1 - \tilde{L}_t) x_{t,a}^\top A^{-1} x_{t,a}}$
10: $\quad$ **end for**
11: $\quad$ Choose action $a_t = \arg\max_{a \in \mathcal{D}_t} p_{t,a}$ with ties broken arbitrarily.
12: $\quad$ Observe nominal reward r_{t,a_t}.
13: $\quad A \leftarrow A + x_{t,a_t} x_{t,a_t}^\top$
14: $\quad b \leftarrow b + x_{t,a_t} r_{t,a_t}$
15: **end for**

4 Adaptive LinUCB

We note that the conventional LinUCB algorithm assumes that the exploration cost factor does not change over time, i.e., $L_t = 1$. Therefore, to minimize the the nominal reward is equivalent to that of the actual reward. When L_t is time-varying and situation dependent as discussed earlier, we need to maximize the total actual reward, which is affected by the variation factor L_t. Motivated by this distinction, we design the adaptive LinUCB algorithm (AdaLinUCB) as in Algo. 1.

In Algo. 1, α is a hyper-parameter, which is an input of the algorithm, and $\tilde{L}_t$ is the normalized variation factor, defined as,

$$\tilde{L}_t = \left([L_t]_{l^{(-)}}^{l^{(+)}} - l^{(-)} \right) / \left(l^{(+)} - l^{(-)} \right), \tag{3}$$

where $l^{(-)}$ and $l^{(+)}$ are the lower and upper thresholds for truncating the variation factor, and $[L_t]_{l^{(-)}}^{l^{(+)}} = \max\{l^{(-)}, \min\{L_t, l^{(+)}\}\}$. That is, $\tilde{L}_t$ normalizes L_t into $[0, 1]$ to capture different ranges of L_t. To achieve good performance, the truncation thresholds should be appropriately chosen to achieve sufficient exploration. Empirical results show that a wide range of threshold values can lead to good performance of AdaLinUCB. Furthermore, these thresholds can be learned online in practice without prior knowledge on the distribution of L_t, as discussed in Sec. 6 and Appendix E of the supplementary material. Note that $\tilde{L}_t$ is only used in AdaLinUCB algorithm. The actual rewards and regrets are based on L_t, not $\tilde{L}_t$.

In Algo. 1, for each time slot, the algorithm updates a matrix A and a vector b. The A is updated in step 13, which is denoted as $A_t = I_d + \sum_{\tau=1}^{t} x_{\tau,a_\tau} x_{\tau,a_\tau}^\top$ in the following analysis. Note that A_t is a positive-definite matrix for any t, and that $A_0 = I_d$. The b is updated in step 14, which is denoted as $b_t = \sum_{\tau=1}^{t} x_{\tau,a_\tau} r_{\tau,a_\tau}$ in the following analysis. Then, we have $\theta_t = A_t^{-1} b_t$ (see step 5), which is the estimation of the unknown parameter $\theta_\star$ based on historical observations.

Specifically, θ_t is the result of a ridge regression for estimating $\theta_\star$, which minimizes a penalized residual sum of squares, i.e., $\theta_t = \arg\min_\theta \left\{\sum_{\tau=1}^{t} \left(r_{\tau,a_\tau} - \langle \theta, x_{\tau,a_\tau}\rangle\right)^2 + \|\theta\|_2^2\right\}$.

In general, the AdaLinUCB algorithm explores more when the variation factor is relatively low, and exploits more when the variation factor is relatively high. To see this, note that the first term of the index $p_{t,a}$ in step 9, i.e., $\theta_{t-1}^\top x_{t,a}$, is the estimation of the corresponding reward; while the second part is an adaptive upper confidence bound modulated by $\tilde{L}_t$, which determines the level of exploration. At one example, when L_t is at its lowest level with $L_t \leq l^{(-)}$, $\tilde{L}_t = 0$, and the index $p_{t,a}$ is the same as that of the LinUCB algorithm, and then the algorithm selects arm in the same way as the conventional LinUCB. At the other extreme, when $\tilde{L}_t = 1$, i.e., $L_t \geq l^{(+)}$, the index $p_{t,a} = \theta_{t-1}^\top x_{t,a}$, which is the estimation of the corresponding reward. That is, when the variation factor is at its highest level, the AdaLinUCB algorithm purely exploits the existing knowledge and selects the current best arm.

5 Performance Analysis

We first summarize the technical assumptions needed for performance analysis: i. Noise satisfies C_{noise}-sub-Gaussian condition, as explained later in (4); ii. The unknown parameter $\theta_\star$ satisfies $||\theta_\star||_2 \leq C_{\text{theta}}$; iii. For $\forall t, \forall a \in \mathcal{D}_t$, $\|x_{t,a}\|_2 \leq C_{\text{context}}$ holds; iv. $\lambda_{\min}(I_d) \geq \max\{1, C_{\text{context}}^2\}$; v. the nominal reward r_{t,a_t} is independent of the variation factor L_t, conditioned on a_t.

We note that assumptions i.-iv. are widely used in contextual bandit analysis [Auer, 2002; Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011; Wu *et al.*, 2016; Wang *et al.*, 2016].

Specifically, the sub-Gaussian condition in assumption i. is a constraint on the tail property of the noise distribution, as that in [Abbasi-Yadkori *et al.*, 2011]. That is, for the noise η_t, we assume that,

$$\forall \zeta \in \mathbb{R}, \quad \mathbb{E}[e^{\zeta \eta_t} | x_{t,a_t}, \mathcal{H}_{t-1}] \leq \exp\left(\frac{\zeta^2 C_{\text{noise}}^2}{2}\right), \quad (4)$$

with $\mathcal{H}_{t-1} = (x_{1,a_1}, \eta_1, \cdots, x_{t-1,a_{t-1}}, \eta_{t-1})$ and $C_{\text{noise}} > 0$. Note that the sub-Gaussian condition requires both (4) and $\mathbb{E}[\eta_t | x_{t,a_t}, \mathcal{H}_{t-1}] = 0$. Further, this condition indicates that $\text{Var}[\eta_t | F_{t-1}] \leq C_{\text{noise}}^2$, where $\{F_t\}_{t=0}^\infty$ is the filtration of σ-algebras for selected context vectors and noises, i.e., $F_t = \sigma(x_{1,a_1}, x_{2,a_2}, \cdots, x_{t+1,a_{t+1}}, \eta_1, \eta_2, \cdots, \eta_t)$. Thus, C_{noise}^2 can be viewed as the (conditional) variance of the noise.

Examples for the distributions that satisfies the sub-Gaussian condition are: 1) A zero-mean Gaussian noise with variance at most C_{noise}^2; 2) A bounded noise with zero-mean and lying in an interval of length at most $2C_{\text{noise}}$.

Assumption iv. can be relaxed by changing the value of A_0 in Algo. 1 from the current identity matrix I_d to a positive-definite matrix with a higher minimum eigenvalue (see Appendix A of the supplementary material for more details).

Assumption v. is valid in many application scenarios. For example, in the network configuration scenario, since the total load resembles the peak load, the network performance, i.e., the nominal reward r_{t,a_t}, is independent of the real load L_t, conditioned on configuration a_t. Also, in the recommendation scenario, the click-through rate (i.e., reward $r_{t,a}$) can be independent of the user influence (i.e., variation factor L_t).

5.1 Problem-Dependent Bounds

We focus on problem-dependent performance analysis here because it can lead to a tighter bound albeit under stronger assumptions. To derive the problem-dependent bound, we assume that there are a finite number of possible context values, and denote this number as N. Then, let $\Delta_{\min}$ denote the minimum nominal reward difference between the best and the "second best" arms. That is, $\Delta_{\min} = \min_t \left\{r_{t,\star} - \max_{a \in \mathcal{D}_t, r_{t,a} \neq r_{t,\star}} r_{t,a}\right\}$. Similarly, let $\Delta_{\max}$ denote the maximum nominal reward difference between arms. That is, $\Delta_{\max} = \max_t \left\{r_{t,\star} - \min_{a \in \mathcal{D}_t} r_{t,a}\right\}$.

As in existing literature for problem-dependent analysis of linear bandits[Abbasi-Yadkori *et al.*, 2011], we assume that single optimal context condition holds here. Specifically, for different time slot $t = 1, 2, \cdots$, there is a single optimal context value. That is, there exists $x_\star \in \mathbb{R}^d$, such that, $x_\star = x_{t,a_t^\star}, \forall t$.

5.2 AdaLinUCB under Binary-Valued Variation

We first introduce the result under a random binary-valued variation factor. We assume that the variation factor L_t is i.i.d. over time, with $L_t \in \{\epsilon_0, 1-\epsilon_1\}$, where $\epsilon_0, \epsilon_1 \geq 0$ and $\epsilon_0 < 1 - \epsilon_1$. Let ρ denote the probability that the variation factor is low, i.e., $\mathbb{P}\{L_t = \epsilon_0\} = \rho$.

Firstly, we note that, for a $\tilde{\delta} \in (0,1)$, there exists a positive integer C_{slots} such that,

$$\forall t \geq C_{\text{slots}},\ \rho t - \sqrt{\frac{t}{2}\log\frac{\tilde{\delta}}{2}} - \frac{16 C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}^2}\Bigg[\log(C_{\text{context}} t) + 2(d-1)\log\left(d\log\frac{d + tC_{\text{context}}^2}{d} + 2\log\frac{2}{\tilde{\delta}}\right) + 2\log\frac{2}{\tilde{\delta}} + (d-1)\log\frac{64 C_{\text{noise}}^2 C_{\text{theta}}^2 C_{\text{context}}}{\Delta_{\min}^2}\Bigg]^2 \geq \frac{4d}{\Delta_{\min}^2}. \quad (5)$$

To see such an integer C_{slots} exists, note that for large enough t, in the left-hand side of the inequality (5), the dominant positive term is $O(t)$ while the dominant negative term is $O(\sqrt{t})$.

To interpret C_{slots}, it is an integer that is large enough so that during C_{slots}-slot period, enough exploration is done in the time slots when variation factor is relatively low, such that to have a relatively tight bound for the estimation of the optimal reward.

Then, we have the following results.

Theorem 1. *Consider the opportunistic contextual bandits with linear payoffs and binary-valued variation factor. With probability at least* $1 - \tilde{\delta}$*, the accumulated regret (regarding*

actual reward) of AdaLinUCB algorithm satisfies,

$$\tilde{\mathbf{R}}_{\text{total}}(T) \leq \epsilon_0 \cdot \frac{16C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}} \Bigg[\log(C_{\text{context}} T) + 2\log\frac{2}{\tilde{\delta}} + 2(d-1)\log\Big(d\log\frac{d+TC_{\text{context}}^2}{d} + 2\log\frac{2}{\tilde{\delta}}\Big) + (d-1)\log\frac{64C_{\text{noise}}^2 C_{\text{theta}}^2 C_{\text{context}}}{\Delta_{\min}^2}\Bigg]^2 + (1-\epsilon_1)\Big[\big(\Delta_{\max} C_{\text{slots}} + 4d\frac{N-1}{\Delta_{\min}}\big) \cdot \Big(C_{\text{noise}}\sqrt{d\log\frac{2+2TC_{\text{context}}^2}{\tilde{\delta}}} + C_{\text{theta}}\Big)^2\Big],$$

where C_{slots} is a constant satisfying (5)*.*

Proof Sketch: Although the proof for Theorem 1 is complicated, the key is to treat the slots with low variation factor and the slots with high variation factor separately. For slots with low variation factor, the one-step regret is upper bounded by the weighted 2-norm of the selected context vectors, i.e., $R_t \mathbb{1}\{L_t = \epsilon_0\} \leq 2\alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}}$, and then the accumulated regret can be analyzed accordingly. For the slots with high variation factor, by matrix analysis, we can show that when a particular context value has been selected enough times, its estimated reward is accurate enough in an appropriate sense. Further, it can benefit from regret bound for low variation factor slots that the optimal context has been selected enough time with high probability. Then, we combine these to prove the result. More details are shown in Appendix B of the supplementary material.

Remark 1. *For the regret bound in Theorem 1, the first three lines cover the accumulated regret that is incurred during time slots when the variation factor is relatively low, i.e., during slots t with $L_t = \epsilon_0$, while the last two lines cover the accumulated regret that is incurred during time slots when the variation factor is relatively high, i.e., during slots t with $L_t = 1 - \epsilon_1$. Further, when T is large enough, the dominant term for the first three lines is $O\left((\log T)^2\right)$, while the dominant term for the last two lines is $O(\log T)$. That is, the bound for the accumulated regret during slots when the variation factor is relatively high actually increases slower than the bound for the accumulated regret during slots when the variation factor is relatively low. This is in consistent with the motivation of AdaLinUCB design: explore more when the variation factor is relatively low, and exploit more when the variation factor is relatively high.*

Furthermore, beside parameter T, which is the time horizon, the regret bound in Theorem 1 is also affected by problem-dependent parameters: it is affected by N, which is the number of possible context values, $\Delta_{\min}$, which is the minimum nominal reward difference between the best and the "second best" arms, and $\Delta_{\max}$, which is the maximum nominal reward difference between arms. In general, a larger number of possible context values, i.e., a larger N, may lead to a larger $\Delta_{\max}$ and a smaller $\Delta_{\min}$, and in this way, results in a larger regret bound.

5.3 AdaLinUCB under Continuous Variation

We now study AdaLinUCB in opportunistic contextual bandits under continuous variation factor. Under continuous variation factor, it is difficult to obtain regret bound for general values of $l^{(-)}$ and $l^{(+)}$ because exploration and exploitation mix in a complex fashion when $l^{(-)} < L_t < l^{(+)}$. Instead, inspired by the insights obtained from the binary-valued variation factor case, we illustrate the advantages of AdaLinUCB for special case with $l^{(-)} = l^{(+)}$.

In the special case of $l^{(-)} = l^{(+)}$, the normalized variation factor $\tilde{L}_t$ in (3) is redefined as $\tilde{L}_t = 0$ when $L_t \leq l^{(-)}$ and as $\tilde{L}_t = 1$ when $L_t > l^{(+)} = l^{(-)}$.

Theorem 2. *In the opportunistic contextual bandits with linear payoffs and continuous variation factor that is i.i.d. over time, under AdaLinUCB with $\mathbb{P}\{L_t \leq l^{(-)}\} = \rho > 0$ and $l^{(-)} = l^{(+)}$, with probability at least $1 - \tilde{\delta}$, the accumulated regret (regarding actual reward) satisfies,*

$$\tilde{\mathbf{R}}_{\text{total}}(T) \leq \mathbb{E}\big[L_t | L_t \leq l^{(-)}\big] \frac{16C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}} \Bigg[\log(C_{\text{context}} T) + 2(d-1)\log\Big(d\log\frac{d+TC_{\text{context}}^2}{d} + 2\log\frac{2}{\tilde{\delta}}\Big) + (d-1)\log\frac{64C_{\text{noise}}^2 C_{\text{theta}}^2 C_{\text{context}}}{\Delta_{\min}^2} + 2\log\frac{2}{\tilde{\delta}}\Bigg]^2 + \mathbb{E}[L_t | L_t > l^{(-)}] \cdot \Bigg[\Big(\Delta_{\max} C_{\text{slots}} + 4d\frac{N-1}{\Delta_{\min}}\Big) \cdot \Big(C_{\text{noise}}\sqrt{d\log\frac{2+2TC_{\text{context}}^2}{\tilde{\delta}}} + C_{\text{theta}}\Big)^2\Bigg],$$

where C_{slots} is a constant satisfying (5)*.*

Proof. Recall that for the special case with $l^{(+)} = l^{(-)}$, we have $\tilde{L}_t = 0$ when $L_t \leq l^{(-)}$ and as $\tilde{L}_t = 1$ when $L_t > l^{(+)}$. Thus, this theorem can be proved analogically to the proof of Theorem 1, by noting the following: When $L_t \leq l^{(-)}$, we have $\tilde{L}_t = 0$ which corresponds to the case of $L_t = \epsilon_0$ ($\tilde{L}_t = 0$) in the binary-valued variation factor case; while when $L_t > l^{(+)}$ ($\tilde{L}_t = 1$) corresponds to the case of $L_t = 1 - \epsilon_1$ under binary-valued variation factor case. The conclusion of the theorem then follows by using the fact that all variation factor below $l^{(-)}$ are treated same by AdaLinUCB, i.e., $\tilde{L}_t = 0$ for $L_t \leq l^{(-)}$; while all variation factor above $l^{(-)}$ are treated same by AdaLinUCB, i.e., $\tilde{L}_t = 1$ for $L_t \leq l^{(+)}$. □

Remark 2. *Similar to Remark 1 for Theorem 1, the regret bound in Theorem 2 can be divided into two parts: the first three lines cover the accumulated regret that is incurred during time slots when $L_t \leq l^{(-)}$ and is $O\left((\log T)^2\right)$, while the last two lines cover the accumulated regret for time slots when $L_t > l^{(-)}$ and is $O\left((\log T)\right)$. Furthermore, a larger N, i.e., a larger number of possible context values, can lead to a larger regret bound.*

5.4 Regret Bound of LinUCB

To the best of our knowledge, there exists no problem-dependent bound on LinUCB. (The initial analysis of LinUCB presents a more general and looser performance bound for a modified version of LinUCB. The modification is needed to satisfy the independent requirement by applying Azuma/Hoeffding inequality [Chu *et al.*, 2011].) Furthermore, we note that one can directly apply LinUCB to opportunistic contextual bandits using the linear relationship $\mathbb{E}[r_{t,a}|x_{t,a}] = \langle x_{t,a}, \theta_\star \rangle$, which is called LinUCBExtracted in numerical results. Therefore, we derive the regret upper bound for LinUCB here, both as an individual contribution as well as for comparison purpose.

Theorem 3. *In the opportunistic contextual bandits with linear payoffs and continuous variation factor that is i.i.d. over time with mean $\bar{L}$, with probability at least $1-\delta$, the accumulated T-slot regret (regarding actual reward) of LinUCB satisfies,*

$$\tilde{\mathbf{R}}_{\text{total}}(T) \leq \frac{16\bar{L}C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}} \Bigg[\log(C_{\text{context}} T) + 2\log\frac{1}{\delta} + 2(d-1)\log\left(d\log\frac{d+TC_{\text{context}}^2}{d} + 2\log\frac{1}{\delta}\right) + (d-1)\log\frac{64C_{\text{noise}}^2 C_{\text{theta}}^2 C_{\text{context}}}{\Delta_{\min}^2}\Bigg]^2.$$

The regret bound for LinUCB under non-opportunistic case can be shown by simply having $\bar{L} = 1$ in the above result. Here, note that problem-dependent bound analysis is a setting that allows a better bound to be achieved with stronger assumptions. Recall that the assumptions are discussed in Sec. 5.1. As a result, the problem-dependent bound of LinUCB is much better than its general bound, such as the bound for a modified version of LinUCB in [Chu *et al.*, 2011]. More results for LinUCB and the proof of Theorem 3 can be found in Appendix C of the supplementary material.

Remark 3. *Theorem 3 and Theorem 1 show that the problem-dependent regret bounds (regarding actual reward) for LinUCB and AdaLinUCB are both $O\left((\log T)^2\right)$. Further, for binary-valued variation factor, the asymptotically dominant term for the bound of LinUCB is $\frac{1-\epsilon_1+\epsilon_0}{2} \cdot \frac{16C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}} (\log T)^2$. In comparison, for AdaLinUCB, it is $\epsilon_0 \cdot \frac{16C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}} (\log T)^2$. Because $\epsilon_0 < 1-\epsilon_1$, in the scenario of binary-valued variation factor, the AdaLinUCB algorithm has a better asymptotic problem-dependent upper bound than that of the LinUCB algorithm. Similarly, in scenario with continuous variation factor, the AdaLinUCB algorithm with $l^{(+)} = l^{(-)}$ has a better problem-dependent bound than LinUCB algorithm as long as $\mathbb{E}[L_t | L_t \leq l^{(-)}] < \bar{L}$, which holds in most cases.*

5.5 Discussions on the Disjoint Model

The seminal paper on LinUCB [Li *et al.*, 2010] introduces different models for contextual bandits. The opportunistic learning applies to these different models. One of them is the joint model discussed above. Another model is the disjoint model, which assumes that, $\mathbb{E}[r_{t,a}|x_{t,a}] = \langle x_{t,a}, \theta_\star^{(a)} \rangle$, where $x_{t,a}$ is a context vector and $\theta_\star^{(a)}$ is the unknown coefficient vector for arm a. This model is called disjoint since the parameters are not shared among different arms. There is also a hybrid model that combines the joint model and the disjoint model.

In this paper, we focus on the design and analysis of opportunistic contextual bandits using the joint model. However, it should be noted that, the AdaLinUCB algorithm in Algo. 1 can be modified slightly and applied to the disjoint model, see Appendix D of the supplementary material for more details. Also, the analysis of the joint model can be extended to the disjoint one. Note that the disjoint model can be converted to a joint model when the number of possible arms is finite. Specifically, for an arbitrary disjoint-model opportunistic contextual bandit problem with $\theta_\star^{(a)}\ \forall a$, an equivalent joint-model problem exists with the joint unknown parameter as $\theta_\star = ([\theta_\star^{(1)}]^\top, [\theta_\star^{(2)}]^\top, \cdots)^\top$ and the context vectors modified accordingly. Thus, the previous analytical results are valued for the disjoint model with appropriate modifications.

6 Numerical Results

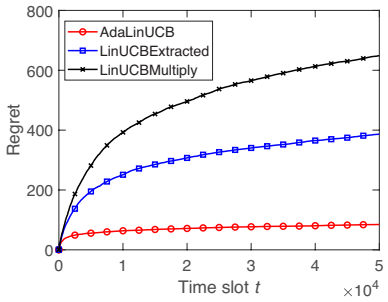
We present numerical results to demonstrate the performance of the AdaLinUCB algorithm using both synthetic scenario and real-world datasets. We have implemented the following algorithms: 1) AdaLinUCB in Algo. 1; 2) LinUCB(Extracted) in Sec. 5.4; 3) **LinUCBMultiply**, another way to directly apply LinUCB in opportunistic case, where we use $L_t \cdot x_{t,a}$ as context vector; 4) **E-AdaLinUCB**, an algorithm that adjusts the threshold $l^{(+)}$ and $l^{(-)}$ based on the empirical distribution of L_t. In all the algorithms, we set $\alpha = 1.5$ to make a fair comparison.

We have also experimented **LinUCBCombine** algorithm, where we use $\tilde{x}_{t,a} = [L_t, x_{t,a}^\top]^\top$ as context vector to directlty apply LinUCB, and find that LinUCBCombine has a much worse performance compared to other algorithms.

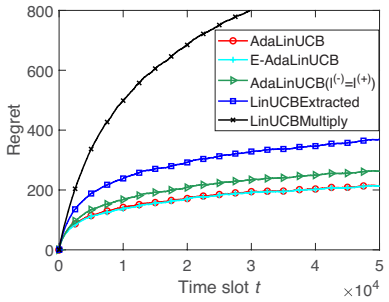
Meanwhile, we also notice that the opportunistic linear contextual bandits can be regarded as a special case of non-linear contextual bandits by viewing the variation factor L_t as a part of context vector. Along this line of thinking, we have also experimented **KernelUCB** algorithm [Valko *et al.*, 2013], which is a general algorithm for non-linear contextual bandits. However, we find that KernelUCB is less competitive in performance and suffers from extremely high computational complexity (see Appendix E.3 of the supplementary material for more details). One reason is that a general contextual bandit algorithm such as KernelUCB does not take advantage of the opportunistic nature of the problem, and can, therefore, have a worse performance than AdaLinUCB.

6.1 Experiments on Synthetic Scenarios

The synthetic scenario has a total of 20 possible arms, each associated with a disjoint unknown coefficient $\theta_\star^{(a)}$. The simulator generates 5 possible groups of context vectors, and each group has context vectors associated with all the possible arms. At each time slot, a context group is presented before the decision. Further, each unknown coefficient $\theta_\star^{(a)}$ and each context $x_{t,a}$ is a 6-dimension vector, with elements in



(a) Binary-valued variation factor



(b) Beta distributed variation factor

Figure 1: Regret under Synthetic Scenarios. In (a), $\epsilon_0 = \epsilon_1 = 0, \rho = 0.5$. In (b), AdaLinUCB: $l^{(-)}$=$l_0^{(-)}$, $l^{(+)}$= $l_0^{(+)}$; AdaLinUCB: ($l^{(-)}$= $l^{(+)}$), $l^{(-)}$= $l^{(+)}$= $l_{0.5}^{(-)}$

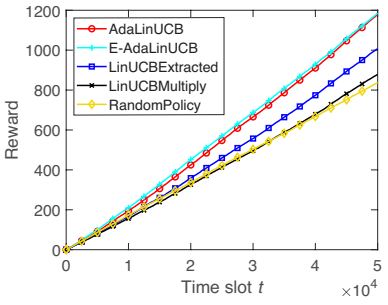


Figure 2: Rewards for Yahoo! Today Module $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.3}^{(+)}$

each dimension generated randomly, and is normalized such that the L2-norm of $\theta_\star^{(a)}$ or $x_{t,a}$ is 1.

Fig. 1(a) shows the regret for different algorithms under random binary-value variation factor with $\epsilon_0 = \epsilon_1 = 0$ and $\rho = 0.5$. AdaLinUCB significantly reduces the regret in this scenario. Specifically, at time slots $t = 5 \times 10^4$, AdaLinUCB achieves a regret that is only 10.3% of that of LinUCBMultiply, and 17.6% of that of LinUCBExtracted.

For continuous variation factor, Fig. 1(b) compares the regrets for the algorithms under a beta distributed variation factor. Here, we define $l_\rho^{(-)}$ as the lower threshold such that $\mathbb{P}\{L_t \le l_\rho^{(-)} = \rho\}$, and $l_\rho^{(+)}$ as the higher threshold such that $\mathbb{P}\{L_t \ge l_\rho^{(+)} = \rho\}$. It is shown that AdaLinUCB still outperforms other algorithms, and AdaLinUCB has a regret 41.8% lower than that of LinUCBExtracted. Furthermore, its empirical version, E-AdaLinUCB has a similar performance to that of AdaLinUCB. Even in the special case with a single threshold $l^{(-)} = l^{(+)} = l_{0.5}^{(-)}$, AdaLinUCB still outperforms LinUCBExtracted, reducing the regret by 28.6%.

We have conducted more simulations to evaluate the impact of environment and algorithm parameters such as variation factor fluctuation and the thresholds for variation factor truncation, and find that AdaLinUCB works well in different scenarios (see Appendix E.1 and E.2 of the supplementary material).

6.2 Experiments on Yahoo! Today Module

We also test the performance of the algorithms using the data from Yahoo! Today Module. This dataset contains over 4 million user visits to the Today module in a ten-day period in May 2009 [Li *et al.*, 2010]. To evaluate contextual bandits using offline data, the experiment uses the unbiased offline evaluation protocol proposed in [Li *et al.*, 2011]. For the variation factor, we use a real trace - the sales of a popular store. It includes everyday turnover in two years [Rossman, 2015].

In this real recommendation scenario, because we do not know the ground truth; i.e., which article is best for a specific user, we cannot calculate the regret. Therefore, all the results are measured using the reward, as shown in Fig. 2. We note that AdaLinUCB increases the reward by 17.0%, compared to LinUCBExtracted, and by 40.8% compared to the random policy. We note that an increase in accumulated reward is typically much more substantial than the same decrease in regret. We also note that E-AdaLinUCB, where one does not assume prior knowledge on the variation factor distribution, achieves a similar performance. This experiment demonstrates the effectiveness of AdaLinUCB and E-AdaLinUCB in practical situations, where the variation factor are continuous and are possibly non-stationary, and the candidate arms are time-varying. More details on the datasets and evaluation under different parameters can be found in Appendix E.4 of the supplementary material.

7 Conclusions

In this paper, we study opportunistic contextual bandits where the exploration cost is time-varying depending on external conditions such as network load or return variation in recommendations. We propose AdaLinUCB that opportunistically chooses between exploration and exploitation based on that external variation factor, i.e., taking the slots with low variation factor as opportunities for more explorations. We prove that AdaLinUCB achieves $O\left((\log T)^2\right)$ problem-dependent regret upper bound, which has a smaller coefficient thatn that of the traditional LinUCB algorithm. Extensive experiment results based on both synthetic and real-world database demonstrate the significant benefits of opportunistic exploration under large exploration cost fluctuations.

Acknowledgments

The work is partially supported supported by NSF through grants CNS-1547461, CNS-1718901, and IIS-1838207.

References

[Abbasi-Yadkori *et al.*, 2011] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2312–2320, 2011.

[Agrawal, 1995] Rajeev Agrawal. Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

[Auer *et al.*, 2002] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002.

[Auer, 2002] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2002.

[Bouneffouf *et al.*, 2012] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *International conference on Neural Information Processing (ICONIP)*, pages 324–331, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Chapelle and Li, 2011] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.

[Chu *et al.*, 2011] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *AISTATS*, pages 208–214, 2011.

[Chuai *et al.*, 2019] Jie Chuai, Zhitang Chen, Guochen Liu, Xueying Guo, Xiaoxiao Wang, Xin Liu, Chongming Zhu, and Feiyi Shen. A collaborative learning based approach for parameter configuration of cellular networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2019.

[Filippi *et al.*, 2010] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 586–594. Curran Associates, Inc., 2010.

[Langford and Zhang, 2008] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 817–824. Curran Associates, Inc., 2008.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *the 19th International Conference on World Wide Web (WWW)*, 2010.

[Li *et al.*, 2011] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 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*, WSDM '11, pages 297–306. ACM, 2011.

[Rossman, 2015] Rossman. *Rossmann Store sales data*, 2015. https://www.kaggle.com/c/rossmann-store-sales/data.

[Valko *et al.*, 2013] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[Walraevens *et al.*, 2003] Joris Walraevens, Bart Steyaert, and Herwig Bruneel. Performance analysis of a single-server atm queue with a priority scheduling. *Computers & Operations Research*, 30(12):1807 – 1829, 2003.

[Wang *et al.*, 2016] Huazheng Wang, Qingyun Wu, and Hongning Wang. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1633–1642. ACM, 2016.

[Wang *et al.*, 2017] Huazheng Wang, Qingyun Wu, and Hongning Wang. Factorization bandits for interactive recommendation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 2017.

[Wu *et al.*, 2016] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 529–538. ACM, 2016.

[Wu *et al.*, 2018] Huasen Wu, Xueying Guo, and Xin Liu. Adaptive exploration-exploitation tradeoff for opportunistic bandits. In *ICML*, 2018.