A Preparatory Results for Analysis

We begin with some preparatory analysis results.

Lemma 1. *Assume that the noise satisfies the C_{noise}-sub-Gaussian condition in* (4)*, and assume that $||\theta_\star||_2 \le C_{\text{theta}}$. Then, the following results hold:*

1) For any $\delta \in (0,1)$, with probability at least $1-\delta$, for all $t \ge 0$, $\theta_\star \in \Big\{\theta \in \mathbb{R}^d : \|\theta_t - \theta_\star\|_{A_t} \le \det(I_d)^{\frac{1}{2}} C_{\text{theta}} + C_{\text{noise}}\sqrt{2\log\left(\frac{\det(A_t)^{\frac{1}{2}}\det(I_d)^{-\frac{1}{2}}}{\delta}\right)}\Big\}$.

2) Further, if $\|x_{t,a}\|_2 \le C_{\text{context}}$ holds for $\forall t, \forall a \in \mathcal{D}_t$, then, with probability at least $1-\delta$, $\theta_\star \in \Big\{\theta \in \mathbb{R}^d : \|\theta_t - \theta_\star\|_{A_t} \le C_{\text{theta}} + C_{\text{noise}}\sqrt{d\log\left(\frac{1+tC^2_{\text{context}}}{\delta}\right)}\Big\}$.

Proof. it simply follows from the fact that θ_t is the result of a ridge regression, and that the sub-Gaussian condition is assumed. The technique is as in [Abbasi-Yadkori *et al.*, 2011] (specifically, Theorem 2 in [Abbasi-Yadkori *et al.*, 2011]). □

Lemma 1 shows that the estimation θ_t is close to the unknown parameter $\theta_\star$ in an appropriate sense.

Lemma 2. *For $\forall t \ge 1$, $\forall a \in \mathcal{D}_t$, the following result holds,* $|\langle\theta_\star, x_{t,a}\rangle - \langle\theta_{t-1}, x_{t,a}\rangle| \le \|\theta_{t-1} - \theta_\star\|_{A_{t-1}} \cdot \|x_{t,a}\|_{A^{-1}_{t-1}}$.

Proof. We have the following,

$$
\begin{aligned}
&|\langle\theta_\star, x_{t,a}\rangle - \langle\theta_{t-1}, x_{t,a}\rangle| \\
=&\left|(\theta_\star - \theta_{t-1})^\top x_{t,a}\right| \\
=&\left|(\theta_\star - \theta_{t-1})^\top A^{\frac{1}{2}}_{t-1}A^{-\frac{1}{2}}_{t-1}x_{t,a}\right| \\
=&\left|\ \langle A^{\frac{1}{2}}_{t-1}(\theta_\star - \theta_{t-1}),\ A^{-\frac{1}{2}}_{t-1}x_{t,a}\rangle\ \right| \\
\le&\|A^{\frac{1}{2}}_{t-1}(\theta_\star - \theta_{t-1})\|_2 \cdot \|A^{-\frac{1}{2}}_{t-1}x_{t,a}\|_2 \\
=&\sqrt{(\theta_\star - \theta_{t-1})^\top A^{\frac{1}{2}}_{t-1}A^{\frac{1}{2}}_{t-1}(\theta_\star - \theta_{t-1})} \cdot \sqrt{x^\top_{t,a}A^{-\frac{1}{2}}_{t-1}A^{-\frac{1}{2}}_{t-1}x_{t,a}} \\
=&\|\theta_{t-1} - \theta_\star\|_{A_{t-1}} \cdot \|x_{t,a}\|_{A^{-1}_{t-1}},
\end{aligned}
$$

where the third equality holds by noting that a positive-definite matrix A_{t-1} is symmetric; the inequality holds by Cauchy–Schwarz inequality. □

Lemma 2 presents an upper bound on the estimation error of the reward corresponding to a given context vector. Combining Lemma 1, the first term of this upper bound, i.e., $\|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$, can be upper bounded with a high probability. To further consider the property of the second term, i.e., $\|x_{t,a}\|_{A^{-1}_{t-1}}$, we bound the summation by the following result.

Lemma 3. *Assume that $\|x_{t,a}\|_2 \le C_{\text{context}}$ holds for $\forall t, \forall a \in \mathcal{D}_t$, and assume that $\lambda_{\min}(I_d) \ge \max\{1, C^2_{\text{context}}\}$, then the following results hold,*

$$
\begin{aligned}
\sum_{t=1}^{T}\|x_{t,a_t}\|^2_{A^{-1}_{t-1}} &\le 2\log\Big(\det(A_T)/\det(I_d)\Big) \qquad (6)\\
&\le 2\left[d\log\left(\frac{\text{trace}(I_d) + TC^2_{\text{context}}}{d}\right) - \log\det(I_d)\right].
\end{aligned}
$$

Lemma 3 directly follows from [Abbasi-Yadkori *et al.*, 2011] (specifically, Lemma 11 of [Abbasi-Yadkori *et al.*, 2011]).

Remark 4. ***(Relax assumption iv. in Sec. 5.)*** *Now, we briefly discuss the way to relax the assumption $\lambda_{\min}(I_d) \ge \max\{1, C^2_{context}\}$ in Lemma 3 and the theorems using this Lemma. For this assumption, it can be relaxed by changing the initial value of matrix A in Algo. 1. Currently, we have the initial matrix value $A_0 = I_d$, leading to the current results in Lemma 3. If the initial value A_0 is changed to a positive-definite matrix with a higher minimum eigenvalue, then with a modified assumption $\lambda_{\min}(A_0) \ge \max\{1, C^2_{context}\}$, Lemma 3 still holds after substituting the matrix I_d by the new A_0. One example of such a new A_0 can be $C_{context} \cdot I_d$ with $C_{context} > 1$. Also, we note that Lemma 2 still holds after changing A_0 to another positive-definite matrix. Further, when changing A_0 to another positive-definite matrix, the first statement of Lemma 1 still holds after substituting the matrix I_d by the new matrix A_0, while the second statement following from the first one and can be changed accordingly. Thus, for assumption $\lambda_{\min}(I_d) \ge \max\{1, C^2_{context}\}$, we can modify the choice of A_0 to relax this assumption.*

B Proof for Theorem 1

Firstly, note that we have some preparatory analysis results, as shown in Appendix A.

We begin with some notations. Let $\mathbf{R}^{(\text{low})}_{\text{total}}(T)$ denote the accumulated regret regarding nominal rewards when the variation factor is low, i.e., $\mathbf{R}^{(\text{low})}_{\text{total}}(T) = \sum_{t=1}^{T} R_t \mathbb{1}\{L_t = \epsilon_0\}$, where $\mathbb{1}\{\cdot\}$ is the indicator function. The $\mathbf{R}^{(\text{high})}_{\text{total}}(T)$ is defined similarly as $\mathbf{R}^{(\text{high})}_{\text{total}}(T) = \sum_{t=1}^{T} R_t \mathbb{1}\{L_t = 1 - \epsilon_1\}$.

Then, we have that,

$$
\tilde{\mathbf{R}}_{\text{total}}(T) = \epsilon_0 \cdot \mathbf{R}^{(\text{low})}_{\text{total}}(T) + (1 - \epsilon_1) \cdot \mathbf{R}^{(\text{high})}_{\text{total}}(T).
$$

As a result, to prove this theorem, it is sufficient to prove that, with probability at least $1 - \tilde{\delta}$, both of the following results hold:

$$
\begin{aligned}
\mathbf{R}^{(\text{low})}_{\text{total}}(T) \le \frac{16C^2_{\text{noise}}C^2_{\text{theta}}}{\Delta_{\min}}\Bigg[&\log(C_{\text{context}}T) + 2\log\frac{2}{\tilde{\delta}} \\
&+ 2(d-1)\log\left(d\log\frac{d + TC^2_{\text{context}}}{d} + 2\log\frac{2}{\tilde{\delta}}\right) \\
&+ (d-1)\log\frac{64C^2_{\text{noise}}C^2_{\text{theta}}C_{\text{context}}}{\Delta^2_{\min}}\Bigg]^2, \qquad (7)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{R}^{(\text{high})}_{\text{total}}(T) \le\Bigg[&4d\frac{N-1}{\Delta_{\min}}\left(C_{\text{noise}}\sqrt{d\log\frac{2+2TC^2_{\text{context}}}{\tilde{\delta}}} + C_{\text{theta}}\right)^2 \\
&+ \Delta_{\max}C_{\text{slots}}\left(C_{\text{noise}}\sqrt{d\log\frac{2+2TC^2_{\text{context}}}{\tilde{\delta}}} + C_{\text{theta}}\right)^2\Bigg]. \qquad (8)
\end{aligned}
$$

B.1 Regret for slots with low variation factor:

Firstly, we focus on $\mathbf{R}^{(\text{low})}_{\text{total}}(T)$. For the binary-valued variation factor, let the lower threshold $l^{(-)} = \epsilon_0$. Then the variation factor $L_t = \epsilon_0$, while the normalized variation factor $\tilde{L}_t = 0$. As a result, the index $p_{t,a}$ in step 9 of Algo. 1 becomes,

$$p_{t,a} = \theta_{t-1}^\top x_{t,a} + \alpha\sqrt{x_{t,a}^\top A_{t-1}^{-1} x_{t,a}},$$

Then, for $\forall t \geq 1$ with $L_t = \epsilon_0$, if $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$, we have,

$$\begin{aligned}
R_t &= \langle x_{t,a_t^\star}, \theta_\star\rangle - \langle x_{t,a_t}, \theta_\star\rangle \\
&\leq \langle x_{t,a_t^\star}, \theta_{t-1}\rangle + \alpha\|x_{t,a_t^\star}\|_{A_{t-1}^{-1}} - \langle x_{t,a_t}, \theta_\star\rangle \\
&\leq \langle x_{t,a_t}, \theta_{t-1}\rangle + \alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}} - \langle x_{t,a_t}, \theta_\star\rangle \\
&= \langle x_{t,a_t}, \theta_{t-1}\rangle - \langle x_{t,a_t}, \theta_\star\rangle + \alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}} \\
&\leq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}\|x_{t,a_t}\|_{A_{t-1}^{-1}} + \alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}} \\
&\leq 2\alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}},
\end{aligned}$$

where the inequality in the second line holds by Lemma 2 and $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$; the inequality in the third line holds by the design of the AdaLinUCB algorithm, specifically, by step 11 of Algo. 1; the inequality in the fifth line holds by Lemma 2, and the last inequality holds by $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$. As a result, we have,

$$R_t \mathbb{1}\{L_t = \epsilon_0\} \leq 2\alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}},$$

with $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$. Then, we have,

$$\mathbf{R}^{(\text{low})}_{\text{total}}(T) = \sum_{t=1}^{T} R_t \mathbb{1}\{L_t = \epsilon_0\} \leq \sum_{t=1}^{T} 2\alpha\|x_{t,a_t}\|_{A_{t-1}^{-1}}, \quad (9)$$

with $\alpha \geq \|\theta_{T-1} - \theta_\star\|_{A_{T-1}}$.

We also note that, by Lemma 1, with probability at least $1 - \frac{\tilde{\delta}}{2}$, for all t,

$$\begin{aligned}
\theta_\star \in \Bigg\{\theta \in \mathbb{R}^d : \|\theta_t - \theta_\star\|_{A_t} \\
\leq C_{\text{theta}} + C_{\text{noise}}\sqrt{d\log\left(\frac{2 + 2tC^2_{\text{context}}}{\tilde{\delta}}\right)}\Bigg\}, \quad (10)
\end{aligned}$$

which substitutes the δ in Lemma 1 by $\frac{\tilde{\delta}}{2}$.

Further, we note that,

$$\mathbf{R}_{\text{total}}(T) = \sum_{t=1}^{T} R_t \leq \sum_{t=1}^{T} \frac{R_t^2}{\Delta_{\min}}, \quad (11)$$

where the inequality holds since either $R_t = 0$ or $\Delta_{\min} <= R_t$.

Then, by combining (9), (10), and (11), it follows from a similar argument as [Abbasi-Yadkori *et al.*, 2011] (specifically, the proof of Theorem 5 in [Abbasi-Yadkori *et al.*, 2011]) that (7) holds with probability at least $1 - \frac{\tilde{\delta}}{2}$. Note that the proof procedure uses Lemma 3 and the single optimal context condition.

B.2 Regret for slots with high variation factor

Now, we focus on $\mathbf{R}^{(\text{high})}_{\text{total}}(T)$. We begin with some notations. The N possible values of context vectors are denoted by $x_{(1)}, x_{(2)}, \cdots, x_{(N)}$ respectively. Without loss of generality, we assume that $x_{(1)}$ is the optimal context value, i.e., $x_{(1)} = x_\star$. Let $m_{t,\star}$ be the number of times that the arm with the optimal context value has been pulled before time slot t, i.e., $m_{t,\star} = \sum_{\tau=1}^{t} \mathbb{1}\{x_{\tau,a_\tau} = x_\star\}$. Similarly, let $m_{t,(n)}$ be the number of times that the arm with context value $x_{(n)}$ has been pulled before time slot t, i.e., $m_{t,(n)} = \sum_{\tau=1}^{t} \mathbb{1}\{x_{\tau,a_\tau} = x_{(n)}\}$. In addition, let $m^{(\text{low})}_{t,\star}$ be the number of times when the variation factor is low and the arm with the optimal context value $x_\star$ has been pulled during t-slot period, i.e., $m^{(\text{low})}_{t,\star} = \sum_{\tau=1}^{t} \mathbb{1}\{x_{\tau,a_\tau} = x_\star\} \cdot \mathbb{1}\{L_\tau = \epsilon_0\}$. Let $m^{(\text{low})}_{t,\text{all}}$ be the number of times when the variation factor is low during t-slot period, i.e., $m^{(\text{low})}_{t,\text{all}} = \sum_{\tau=1}^{t} \mathbb{1}\{L_\tau = \epsilon_0\}$. Further, let $m^{(\text{low})}_{t,\text{subopt}}$ be the number of times when the variation factor is low and the arm with a suboptimal context value has been pulled during t-slots, i.e., $m^{(\text{low})}_{t,\text{subopt}} = m^{(\text{low})}_{t,\text{all}} - m^{(\text{low})}_{t,\star}$.

Lemma 4. *For the AdaLinUCB algorithm, the following inequality holds, for any $n = 1, 2, \cdots, N$,*

$$\|x_{(n)}\|_{A_{t-1}^{-1}} \leq \sqrt{\frac{d}{m_{t-1,(n)}}}.$$

Proof. We note that,

$$\begin{aligned}
d &= \text{trace}(I_d) = \text{trace}\left(A_{t-1}^{-1}A_{t-1}\right) \\
&= \text{trace}\left(A_{t-1}^{-1}\left[\sum_{i=1}^{N} m_{t-1,(i)} \cdot x_{(i)}x_{(i)}^\top + I_d\right]\right) \\
&= \text{trace}\left(\sum_{i=1}^{N} m_{t-1,(i)} \cdot A_{t-1}^{-1}x_{(i)}x_{(i)}^\top + A_{t-1}^{-1}\right) \\
&= \sum_{i=1}^{N} m_{t-1,(i)} \cdot \text{trace}\left(A_{t-1}^{-1}x_{(i)}x_{(i)}^\top\right) + \text{trace}\left(A_{t-1}^{-1}\right) \\
&= \sum_{i=1}^{N} m_{t-1,(i)} \cdot \text{trace}\left(x_{(i)}^\top A_{t-1}^{-1}x_{(i)}\right) + \text{trace}\left(A_{t-1}^{-1}\right) \\
&= \sum_{i=1}^{N} m_{t-1,(i)} \cdot x_{(i)}^\top A_{t-1}^{-1}x_{(i)} + \text{trace}\left(A_{t-1}^{-1}\right) \\
&\geq m_{t-1,(n)} \cdot x_{(n)}^\top A_{t-1}^{-1}x_{(n)}, \quad \forall n = 1, 2, \cdots, N,
\end{aligned}$$

where the last inequality holds by noting that A_{t-1}^{-1} is a positive-definite matrix. Then, the results follow. □

In the following, we let,

$$\alpha_T = C_{\text{noise}}\sqrt{d\log\left(\frac{2 + 2TC^2_{\text{context}}}{\tilde{\delta}}\right)} + C_{\text{theta}}. \quad (12)$$

Thus, by (10), with probability at least $1 - \frac{\tilde{\delta}}{2}$, for all $t \leq T$, the following inequality holds.

$$\|\theta_t - \theta_\star\|_{A_t} \leq \alpha_T. \quad (13)$$

Let the α in AdaLinUCB algorithm be $\alpha = \alpha_T$.

Lemma 5. *When inequality* (13) *holds, for any slot t with variation factor $L_t = 1 - \epsilon_1$, if,*

$$\langle x_\star, \theta_\star\rangle - \langle x_{(n)}, \theta_\star\rangle > \alpha_T \|x_\star\|_{A_{t-1}^{-1}} + \alpha_T \|x_{(n)}\|_{A_{t-1}^{-1}}, \quad \forall n, \tag{14}$$

then the arm with the optimal context is pulled in slot t, i.e., $R_t = 0$.

Proof. For the binary-valued variation factor, let the higher threshold of variation factor $l^{(+)} = 1 - \epsilon_1$. Thus, when the variation factor is high, i.e., $L_t = 1 - \epsilon_1$, the truncated variation factor becomes $\tilde{L}_t = 1$. As a result, the index in step 9 of Algo. 1 becomes,

$$p_{t,a} = \theta_{t-1}^\top x_{t,a} = \langle x_{t,a}, \theta_{t-1}\rangle.$$

As a result, to prove that the arm with optimal context value is selected, it is sufficient to prove that,

$$\langle x_\star, \theta_{t-1}\rangle - \langle x_{(n)}, \theta_{t-1}\rangle > 0, \ \ \forall n = 2, \cdots, N.$$

When inequality (13) holds, by Lemma 2 , we have that,

$$\langle x_\star, \theta_{t-1}\rangle \geq \langle x_\star, \theta_\star\rangle - \alpha_T \|x_\star\|_{A_{t-1}^{-1}},$$

and,

$$\langle x_{(n)}, \theta_{t-1}\rangle \leq \langle x_{(n)}, \theta_\star\rangle - \alpha_T \|x_{(n)}\|_{A_{t-1}^{-1}}.$$

As a result, for any $n = 2, 3, \cdots, N$,

$$\begin{aligned}&\langle x_\star, \theta_{t-1}\rangle - \langle x_{(n)}, \theta_{t-1}\rangle \\ \geq &\langle x_\star, \theta_\star\rangle - \langle x_{(n)}, \theta_\star\rangle - \alpha_T \|x_\star\|_{A_{t-1}^{-1}} - \alpha_T \|x_{(n)}\|_{A_{t-1}^{-1}} \\ > &0,\end{aligned}$$

where the last inequality holds by the condition (14) of this Lemma, which completes the proof. □

By Lemma 5, when inequality (13) holds, for any slot t with variation factor $L_t = 1 - \epsilon_1$, if both of the following inequalities holds,

$$\alpha_T \|x_\star\|_{A_{t-1}^{-1}} \leq \frac{\Delta_{\min}}{2}, \tag{15}$$

$$\alpha_T \|x_{(n)}\|_{A_{t-1}^{-1}} < \frac{\langle x_\star, \theta_\star\rangle - \langle x_{(n)}, \theta_\star\rangle}{2}, \quad \forall n = 2, \cdots, N, \tag{16}$$

then the arm with the optimal context is selected with probability at least $1 - \tilde{\delta}$.

Now, we analyze when (16) holds. For any suboptimal context value $x_{(n)}$ with $n \neq 1$, by Lemma 4, (16) holds when,

$$m_{t-1,(n)} > \frac{4d\alpha_T^2}{\left[\langle x_\star, \theta_\star\rangle - \langle x_{(n)}, \theta_\star\rangle\right]^2}.$$

As a result, before (16) is satisfied, pulling the arms with the suboptimal context values increases $\mathbf{R}_{\text{total}}^{(\text{high})}(T)$ by at most,

$$\sum_{n=2}^{N} \frac{4d\alpha_T^2}{\langle x_\star, \theta_\star\rangle - \langle x_{(n)}, \theta_\star\rangle} \leq (N-1)\frac{4d}{\Delta_{\min}}\alpha_T^2. \tag{17}$$

Note that the r.h.s. of (17) is the first term of (8) by recalling α_T definition in (12).

Now, we focus on analyzing when (15) holds. For optimal context value $x_{(1)} = x_\star$, by Lemma 4, (15) holds when,

$$m_{t-1,\star} > \frac{4d\alpha_T^2}{\Delta_{\min}^2}. \tag{18}$$

To analyze when (18) holds, we can take advantage of (7), and note that,

$$m_{t,\text{subopt}}^{(\text{low})} \leq \frac{\mathbf{R}_{\text{total}}^{(\text{low})}(t)}{\Delta_{\min}}.$$

Thus, by (7), with probability at least $1 - \frac{\tilde{\delta}}{2}$, for all t,

$$\begin{aligned} m_{t,\text{subopt}}^{(\text{low})} \leq \frac{16 C_{\text{noise}}^2 C_{\text{theta}}^2}{\Delta_{\min}^2} \Bigg[& \log(C_{\text{context}} t) \\ & + 2(d-1)\log\left(d\log\frac{d + tC_{\text{context}}^2}{d} + 2\log\frac{2}{\tilde{\delta}}\right) \\ & + (d-1)\log\frac{64 C_{\text{noise}}^2 C_{\text{theta}}^2 C_{\text{context}}}{\Delta_{\min}^2} + 2\log\frac{2}{\tilde{\delta}}\Bigg]^2, \end{aligned} \tag{19}$$

where the probability is introduced by (10) when proving (7).

Further, for the binary-valued variation factor, by Hoeffding's inequality, we have that, with probability at least $1 - \frac{\tilde{\delta}}{2}$,

$$m_{t,\text{all}}^{(\text{low})} \geq \rho t - \sqrt{\frac{t}{2}\log\frac{2}{\tilde{\delta}}},$$

which also holds when $t = \alpha_T^2 C_{\text{slots}}$. Thus, with probability at least $1 - \frac{\tilde{\delta}}{2}$,

$$m_{\alpha_T^2 C_{\text{slots}},\text{all}}^{(\text{low})} \geq \rho \cdot \alpha_T^2 C_{\text{slots}} - \sqrt{\frac{\alpha_T^2 C_{\text{slots}}}{2}\log\frac{2}{\tilde{\delta}}}. \tag{20}$$

Then, by combining (19) and (20), and by recalling C_{slots} definition in (5), with probability at least $1 - \tilde{\delta}$,

$$\begin{aligned} m_{\alpha_T^2 C_{\text{slots}},\star} &\geq m_{\alpha_T^2 C_{\text{slots}},\star}^{(\text{low})} \\ &= m_{\alpha_T^2 C_{\text{slots}},\text{all}}^{(\text{low})} - m_{\alpha_T^2 C_{\text{slots}},\text{subopt}}^{(\text{low})} \\ &\geq \frac{4d\alpha_T^2}{\Delta_{\min}^2}. \end{aligned}$$

Thus, with probability at least $1 - \tilde{\delta}$, for $\forall t \geq \alpha_T^2 C_{\text{slots}}$, the inequality (15) holds. As a result, with probability at least $1 - \tilde{\delta}$, before (15) is satisfied, pulling the arms with the suboptimal context values increases $\mathbf{R}_{\text{total}}^{(\text{high})}(T)$ by at most $\alpha_T^2 C_{\text{slots}} \Delta_{\max}$. By combining (17), we have that, with probability at least $1 - \tilde{\delta}$, the inequality (8) for $\mathbf{R}_{\text{total}}^{(\text{high})}(T)$ holds.

B.3 Combine Results and Finish Proof

Further by noting that probabilities introduced in this proof procedure only comes from two events: i) confidence set for θ_t in (10); ii) lower bound for number of total slots with low variation factor as in (20). Note that each of these two events with probability at least $1 - \frac{\tilde{\delta}}{2}$ and that they are independent. Thus, with probability at least $1 - \tilde{\delta}$, both inequalities (7) and (8) hold, which completes the proof.

Algorithm 2 LinUCB(Extracted)

```
1: Inputs: $\alpha \in \mathbb{R}_+$, $d \in \mathbb{N}$.
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:     $\theta_{t-1} = A^{-1} b$
6:     Observe possible arm set $\mathcal{D}_t$, and observe associated context vectors $x_{t,a}, \forall a \in \mathcal{D}_t$.
7:     for $a \in \mathcal{D}_t$ do
8:         $p_{t,a} \leftarrow \theta_{t-1}^\top x_{t,a} + \alpha\sqrt{x_{t,a}^\top A^{-1} x_{t,a}}$ {Computes upper confidence bound}
9:     end for
10:    Choose action $a_t = \arg\max_{a \in \mathcal{D}_t} p_{t,a}$ with ties broken arbitrarily.
11:    Observe nominal reward $r_{t,a_t}$
12:    $A \leftarrow A + x_{t,a_t} x_{t,a_t}^\top$
13:    $b \leftarrow b + x_{t,a_t} r_{t,a_t}$
14: end for
```

C Performance Analysis of LinUCB

C.1 LinUCB Algorithm Notation

In opportunistic contextual bandit problem, one way to select bandits is to ignore the variation factor, i.e., L_t, and just employ the LinUCB algorithm, as shown in Algorithm 2. This algorithm is denoted as LinUCBExtracted in numerical restuls.

In Algo. 2, for each time slot, the algorithm updates an matrix A and a vector b, so that to estimate the unknown parameter for the linear function of context vector. To make the notation clear, denote $A_t = I_d + \sum_{\tau=1}^{t} x_{\tau,a_\tau} x_{\tau,a_\tau}^\top$, which is the matrix A updated in step 12 for each time slot. It directly follows that $A_t, \forall t \geq 0$ is a positive-definite matrix. Denote $b_t = \sum_{\tau=1}^{t} x_{\tau,a_\tau} r_{\tau,a_\tau}$, which is the vector b updated in step 13 for each time slot t.

As a result, the estimation of the unknown parameter $\theta_\star$ is denoted by θ_t, as shown in step 5, which satisfies,

$$\begin{aligned} \theta_t &= A_t^{-1} b_t \qquad (21) \\ &= \left(I_d + \sum_{\tau=1}^{t} x_{\tau,a_\tau} x_{\tau,a_\tau}^\top \right)^{-1} \sum_{\tau=1}^{t} x_{\tau,a_\tau} r_{\tau,a_\tau}. \end{aligned}$$

Note that θ_t is the result of a ridge regression. That is, θ_t is the coefficient that minimize a penalized residual sum of squares, i.e.,

$$\theta_t = \arg\min_{\theta} \left\{ \sum_{\tau=1}^{t} \left(r_{\tau,a_\tau} - \langle \theta, x_{\tau,a_\tau} \rangle \right)^2 + \|\theta\|_2^2 \right\} \qquad (22)$$

Here, the complexity parameter that controls the amount of shrinkage is chosen as 1.

Also, we note that the upper confidence index $p_{t,a}$, as shown in step 8 of Algo. 2 consists of two parts. The first part $\theta_{t-1}^\top x_{t,a} = \langle \theta_{t-1}, x_{t,a} \rangle$ is the estimation of the corresponding reward, using the up-to-date estimation of the unknown parameter, i.e., θ_{t-1}. The second part, i.e., $\alpha\sqrt{x_{t,a}^\top A_{t-1}^{-1} x_{t,a}} = \alpha \|x_{t,a}\|_{A_{t-1}^{-1}}$, is related to the uncertainty of reward estimation.

In the following, to analyze the performance of LinUCB algorithm, we assume the same assumptions as in Sec. 5.

C.2 General Performance Bound

Now, we analyze the performance of LinUCB algorithm. We note that the initial analysis effort of LinUCB[Chu *et al.*, 2011] presents analysis result for a modified version of LinUCB to satisfied the independent requirement by applying Azuma/Hoeffding inequality [Chu *et al.*, 2011]. As a result, we firstly provide the general performance analysis of LinUCB. We have used analysis technique as in [Abbasi-Yadkori *et al.*, 2011]. (Note that [Abbasi-Yadkori *et al.*, 2011] provides analysis for another algorithm instead of LinUCB, but its technique is helpful.)

Firstly, we note that since A_t, b_t, θ_t has the same definition as that in AdaLinUCB Algorithm, the previous Lemma 1, Lemma 2, and Lemma 3 also hold here for LinUCB algorithm. Then, we have the following results.

Theorem 4. *(The general regret bound of LinUCB). For the LinUCB algorithm in Algo. 2, consider traditional contextual bandits with linear payoffs, the following results hold.*

1) $\forall t \geq 1$, *if* $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$, *then the one-step regret (regarding nominal reward) satisfies,*

$$R_t \leq 2\alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}}.$$

2) $\forall \delta \in (0, 1)$, *with probability at least* $1 - \delta$, *the accumulated* T*-slot regret (regarding nominal reward) satisfies,*

$$\begin{aligned} \mathbf{R}_{\text{total}}(T) \leq & \sqrt{8T} \left[C_{\text{noise}} \sqrt{d \log\left(\frac{1+TC_{\text{context}}^2}{\delta} \right)} + C_{\text{theta}} \right] \\ & \cdot \sqrt{d \log\left[\frac{\text{trace}(I_d) + TC_{\text{context}}^2}{d} \right] - \log\det(I_d)}. \end{aligned} \qquad (23)$$

Proof. We begin by analyzing the one-step regret (regarding nominal reward) of LinUCB algorithm in Algo. 2. For $\forall t \geq 1$, with $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$, we have,

$$\begin{aligned} R_t &= \langle x_{t,a_t^\star}, \theta_\star \rangle - \langle x_{t,a_t}, \theta_\star \rangle \\ &\leq \langle x_{t,a_t^\star}, \theta_{t-1} \rangle + \alpha \|x_{t,a_t^\star}\|_{A_{t-1}^{-1}} - \langle x_{t,a_t}, \theta_\star \rangle \\ &\leq \langle x_{t,a_t}, \theta_{t-1} \rangle + \alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}} - \langle x_{t,a_t}, \theta_\star \rangle \\ &= \langle x_{t,a_t}, \theta_{t-1} \rangle - \langle x_{t,a_t}, \theta_\star \rangle + \alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}} \\ &\leq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}} \|x_{t,a_t}\|_{A_{t-1}^{-1}} + \alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}} \\ &\leq 2\alpha \|x_{t,a_t}\|_{A_{t-1}^{-1}}, \end{aligned}$$

where the inequality in the second line holds by Lemma 2 and $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$; the inequality in the third line holds by the design of the LinUCB algorithm, specifically, by step 10 of Algo. 2; the inequality in the fifth line holds by Lemma 2, and the last inequality holds by $\alpha \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$. As a result, the first statement is proved.

Now, we analyze the accumulated regret. Let,

$$\alpha_T = C_{\text{noise}}\sqrt{d\log\left(\frac{1+TC_{\text{context}}^2}{\delta}\right)} + C_{\text{theta}}.$$

Then, by Lemma 1, with probability at least $1-\delta$, for $\forall t \in [1,T]$, $\alpha_T \geq \|\theta_{t-1} - \theta_\star\|_{A_{t-1}}$. As a result, with probability at least $1-\delta$,

$$\begin{aligned}
\mathbf{R}_{\text{total}}(T) = \sum_{t=1}^{T} R_t &\leq \sqrt{T\sum_{t=1}^{T} R_t^2} \\
&\leq \sqrt{T\cdot 4\alpha_T^2 \sum_{t=1}^{T}\|x_{t,a_t}\|_{A_{t-1}^{-1}}^2} \\
&\leq \sqrt{8T\alpha_T^2} \\
&\cdot \sqrt{d\log\left[\frac{\text{trace}(I_d)+TC_{\text{context}}^2}{d}\right] - \log\det(I_d)},
\end{aligned}$$

where the first inequality holds by Jensen's inequality; the second inequality holds by statement 1); the third inequality holds by Lemma 3. Thus, by substituting the value of α_T, the inequality (23) holds. □

C.3 Problem-Dependent Bound

Now, we study the problem-dependent bound of LinUCB, and have the following results.

Theorem 5. *For the LinUCB algorithm in Algo. 2, consider traditional contextual bandit setting with linear payoffs, the accumulated T-slot regret (regarding nominal reward) satisfies,*

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

Proof. We note that,

$$\mathbf{R}_{\text{total}}(T) = \sum_{t=1}^{T} R_t \leq \sum_{t=1}^{T}\frac{R_t^2}{\Delta_{\text{min}}},$$

where the inequality holds since either $R_t = 0$ or $\Delta_{\text{min}} <= R_t$. Then, the results follows from same proof as in [Abbasi-Yadkori *et al.*, 2011] (see the proof of Theorem 5 in [Abbasi-Yadkori *et al.*, 2011]). Note that the proof procedure uses Lemma 3 and the single optimal context condition. □

C.4 Performance for Opportunistic Case - Proof of Theorem 3

Note that the arm selection strategy in LinUCB in Algo. 2 is independent of the value of L_t. Thus, when L_t is i.i.d. over time, we have that $\tilde{\mathbf{R}}_{\text{total}}(T) = \bar{L}\mathbf{R}_{\text{total}}(T)$. As a result, Theorem 3 directly follows from Theorem 5.

Algorithm 3 LinUCB(Multiply)

```
Inputs: $\alpha \in \mathbb{R}_+$, $d \in \mathbb{N}$.
$A \leftarrow \boldsymbol{I}_d$ {The $d$-by-$d$ identity matrix}
$b \leftarrow \mathbf{0}_d$
for $t = 1, 2, 3, \cdots, T$ do
    $\theta_{t-1} = A^{-1}b$
    Observe possible arm set $\mathcal{D}_t$, and observe associated context vectors $x_{t,a}, \forall a \in \mathcal{D}_t$.
    Observe $L_t$, and get $\tilde{x}_{t,a} = L_t \cdot x_{t,a}, \forall a \in \mathcal{D}_t$.
    for $a \in \mathcal{D}_t$ do
        $p_{t,a} \leftarrow \theta_{t-1}^\top \tilde{x}_{t,a} + \alpha\sqrt{\tilde{x}_{t,a}^\top A^{-1}\tilde{x}_{t,a}}$ {Computes upper confidence bound}
    end for
    Choose action $a_t = \arg\max_{a\in\mathcal{D}_t} p_{t,a}$ with ties broken arbitrarily.
    Observe nominal reward $r_{t,a_t}$ and get actual reward $\tilde{r}_{t,a_t} = L_t \cdot r_{t,a_t}$.
    $A \leftarrow A + \tilde{x}_{t,a_t}\tilde{x}_{t,a_t}^\top$
    $b \leftarrow b + \tilde{x}_{t,a_t}\tilde{r}_{t,a_t}$
end for
```

C.5 Another Way to Apply LinUCB in Opportunistic Linar Bandits

Beside the LinUCBExtracted algorithm in Algo. 2, we also note that there is another way to directly apply in LinUCB in opportunistic contextual bandit environment. Recall that the LinUCBExtracted algorithm in Algo. 2 is based on the linear relationship, $\mathbb{E}[r_{t,a}|x_{t,a}] = \langle x_{t,a}, \theta_\star\rangle$. We can also apply the LinUCBMultiply algorithm in Algo. 3, which is based on the linear relationship, $\mathbb{E}[L_t \cdot r_{t,a}|x_{t,a}, L_t] = \langle L_t \cdot x_{t,a}, \theta_\star\rangle$, i.e., regarding $L_t \cdot x_{t,a}$ as context vector.

Thus, we have also implemented LinUCBMultiply in the numerical results. However, from the experiment results, LinUCBExtracted algorithm has a better performance than LinUCBMultiply.

D AdaLinUCB for Disjoint Model

In above, we focus on the design and analysis of opportunistic contextual bandit for the joint model. However, it should be noted that, the AdaLinUCB algorithm in Algo. 1 can be modified slightly and then be applied to the disjoint model, which is shown in the Algo. 4.

Here, we note that the joint model is the model introduced in Sec. 3:, which assumes that,

$$\mathbb{E}[r_{t,a}|x_{t,a}] = \langle x_{t,a}, \theta_\star\rangle,$$

where $x_{t,a}$ is a context vector and $\theta_\star$ is the unknown coefficient vector. 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.

The joint and disjoint models correspond to different models for linear contextual bandit problems, as introduced in the seminal paper on LinUCB [Li *et al.*, 2010].

Algorithm 4 AdaLinUCB - Disjoint Model

```
Inputs: $\alpha \in \mathbb{R}_+$, $d \in \mathbb{N}$, $l^{(+)}$, $l^{(-)}$.
$A^{(a)} \leftarrow \boldsymbol{I}_d$, $\forall a$
$b^{(a)} \leftarrow \mathbf{0}_d$, $\forall a$
for $t = 1, 2, 3, \cdots, T$ do
    Observe possible arm set $\mathcal{D}_t$, and observe associated context vectors $x_{t,a}, \forall a \in \mathcal{D}_t$.
    Observe $L_t$ and calculate $\tilde{L}_t$ by (3).
    for $a \in \mathcal{D}_t$ do
        $\theta_{t-1}^{(a)} = [A^{(a)}]^{-1} b^{(a)}$
        $p_{t,a} \leftarrow [\theta_{t-1}^{(a)}]^\top x_{t,a} + \alpha \sqrt{(1 - \tilde{L}_t) x_{t,a}^\top [A^{(a)}]^{-1} x_{t,a}}$
    end for
    Choose action $a_t = \arg\max_{a \in \mathcal{D}_t} p_{t,a}$ with ties broken arbitrarily.
    Observe nominal reward $r_{t,a_t}$.
    $A^{(a)} \leftarrow A^{(a)} + x_{t,a_t} x_{t,a_t}^\top$
    $b^{(a)} \leftarrow b^{(a)} + x_{t,a_t} r_{t,a_t}$
end for
```

E More Numerical Results

We have implemented AdaLinUCB (as in Algo. 1), LinUCBExtracted (as in Algo. 2), and LinUCBMultiply (as in Algo. 3). We have also implemented **E-AdaLinUCB** algorithm, which is an algorithm that adjusts the threshold $l^{(+)}$ and $l^{(-)}$ based on the empirical distribution of L_t. Specifically, the E-AdaLinUCB algorithm maintains the empirical histogram for the variation factors (or its moving average version for non-stationary cases), and selects $l^{(+)}$ and $l^{(-)}$ accordingly. Furthermore, the results for **KernelUCB** is shown in Appendix E.3.

E.1 Synthetic Scenario with Binary-Valued variation Factor

Fig. 3 shows the performance of different algorithms with binary-valued variation factor for different value of ρ. From the simulation result, the AdaLinUCB algorithm significantly outperforms other algorithms for different values of ρ.

E.2 Synthetic Scenario with Beta Distributed variation Factor

Here, we define $l_\rho^{(-)}$ as the lower threshold such that $\mathbb{P}\{L_t \leq l_\rho^{(-)} = \rho\}$, and $l_\rho^{(+)}$ as the lower threshold such that $\mathbb{P}\{L_t \geq l_\rho^{(+)} = \rho\}$. The simulation results demonstrate that, with appropriately chosen parameters, the proposed AdaLinUCB algorithm (and its empirical version E-AdaLinUCB) achieves good performance by leveraging the variation factor fluctuation in opportunistic contextual bandits. Furthermore, it turns out that, for a large range of $l^{(+)}$ and $l^{(-)}$ values, AdaLinUCB performs well. Meanwhile, E-AdaLinUCB has a similar performance as that of AdaLinUCB in different scenarios.

In Fig. 4, we implement both AdaLinUCB with a single threshold $l^{(-)} = l^{(+)}$, and AdaLinUCB (and E-AdaLinUCB) with two different threshold values. We find that AdaLinUCB and E-AdaLinUCB perform well for all these appropriate choices of $l^{(-)}$ and $l^{(+)}$. In addition, even in the special case with a single threshold $l^{(-)} = l^{(+)}$, AdaLinUCB has a better performance than other algorithms.

We evaluate the impact of $l^{(-)}$ and $l^{(+)}$ separately with the other one fixed in Fig. 5 and Fig. 6, respectively. Compared them, we can see that the impact of threshold values under continuous variation factor is insignificant (when $l^{(+)}$ and $l^{(-)}$ are changing in wide appropriate ranges), and the regret of AdaLinUCB is significantly lower than that of LinUCBExtracted and LinUCBMultiple.

E.3 Compare with KernelUCB

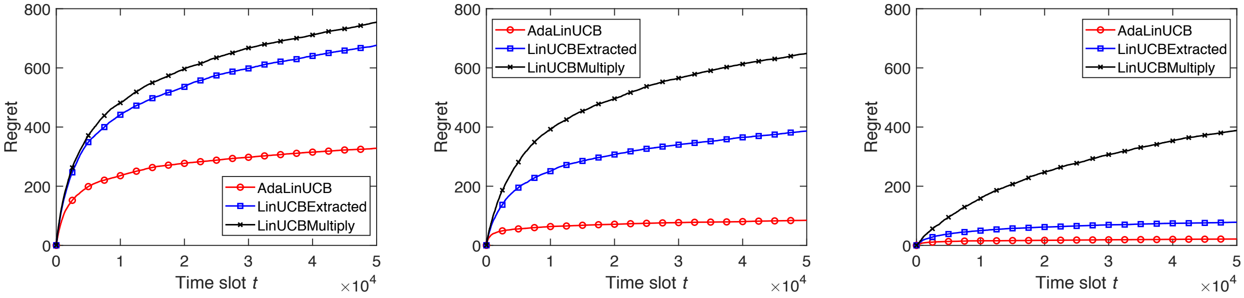
We have also implemented KernelUCB [Valko *et al.*, 2013] which is a kernel-based upper confidence bound algorithm. It applies for general contextual bandits with non-linear payoffs. It can characterize general non-linear relationship between the context vector and reward based on the kernel that defines the similarity between two data points. There are many widely used kernels, such as Gaussian kernel, Laplacian kernel and polynomial kernel [Rasmussen, 2004].

We demonstrate KernelUCB in Algo. 5. The algorithm is based on paper [Valko *et al.*, 2013]. Furthermore, in line 10, we have actually used the technique of Schur complement [Zhang, 2006] to update of kernel matrix $\boldsymbol{K}_t$ so as to boost the implementation of KernelUCB.

Fig. 7 demonstrates the performance of AdaLinUCB, LinUCBExtracted, and KernelUCB (with carefully selected hyper-parameters) for different scenarios. Note that the performance of KernelUCB highly depends on the choice of hyper-parameter. To make a fair comparison, we test the performance of KernelUCB for different hyper-parameter values, and chooses the hyper-parameters with the best performance (among the hyper-parameter values that we have experimented), i.e., $\Gamma_{\text{kernel}} = 2$ for Gaussian kernel $k(z_1, z_2) = \exp(-\Gamma_{\text{kernel}}||z_1 - z_2||^2)$, $\lambda_{\text{regularization}} = 0.5$ for kernel ridge regression. As shown in Fig. 7, AdaLinUCB outperforms KernelUCB under both binary-valued variation factor and continuous variation factor.

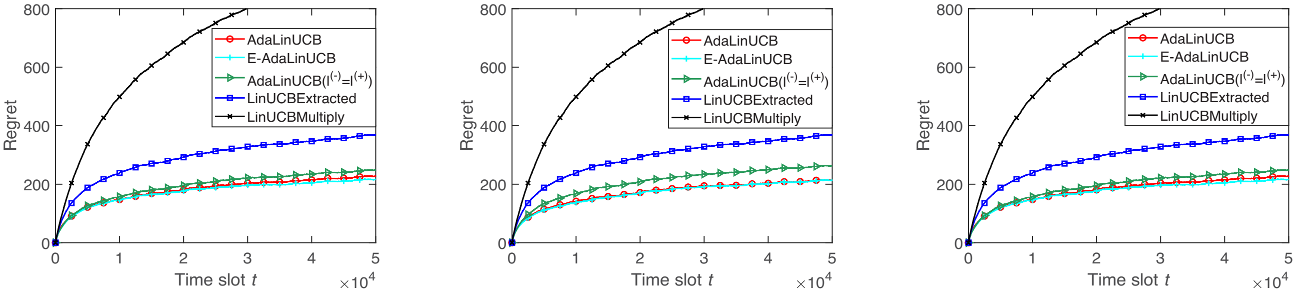
Besides the less competitive performance, as show in Fig. 7, there are two other severe drawbacks that prevents the application of KernelUCB in many practical scenarios. Firstly, its performance is highly sensitive to the choice of hyper-parameters. As discussed above, we have tested the performance of KernelUCB for different hyper-parameter values, and chooses the hyper-parameters with the best performance for a fair comparison. However, even when the hyper-parameters just changes slightly (or environment such as variation factor fluctuation changes slightly), the performance of KernelUCB can deteriorate severely such that it performs even worse then LinUCBExtracted.

Secondly, KernelUCB suffers from the high computational complexity problem. Even if we have used the technique of Schur complement [Zhang, 2006] to update of $\boldsymbol{K}_t$ so as to boost the implementation of KernelUCB as paper [Valko *et al.*, 2013] , it still suffers from prohibitively high computational complexity even for moderately long time horizon. This is also the reason why Fig. 7 has a shorter time horizon than other figures. Specifically, even to run a 10^4-slot simulation, the time to run KernelUCB algorithm is at least 70 times longer than the time to run AdaLinUCB algorithm. In



(a) $\rho = 0.1$ (b) $\rho = 0.5$ (c) $\rho = 0.9$

Figure 3: Regret under binary-valued variation factor.

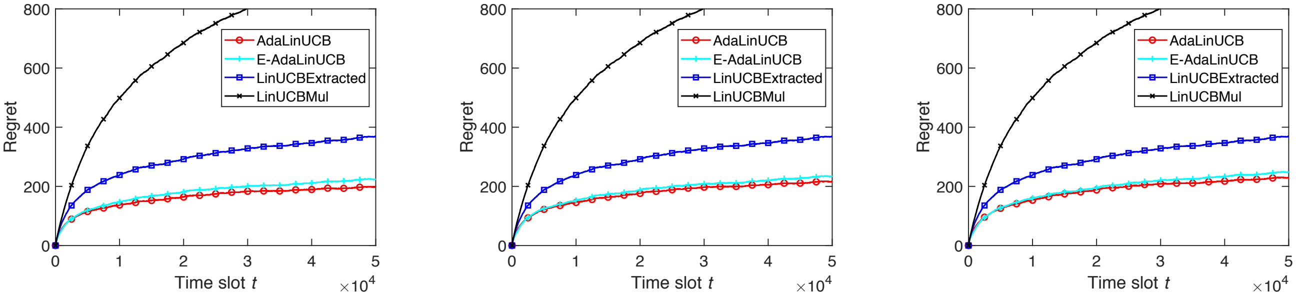


(a) AdaLinUCB: $l^{(-)} = l^{(-)}_{0.05}, l^{(+)} = l^{(+)}_{0.05}$; AdaLinUCB($l^{(-)} = l^{(+)}$):$l^{(-)} = l^{(+)} = 0.45$

(b) AdaLinUCB: $l^{(-)} = l^{(-)}_{0}, l^{(+)} = l^{(+)}_{0}$; AdaLinUCB($l^{(-)} = l^{(+)}$): $l^{(-)} = l^{(+)} = 0.5$

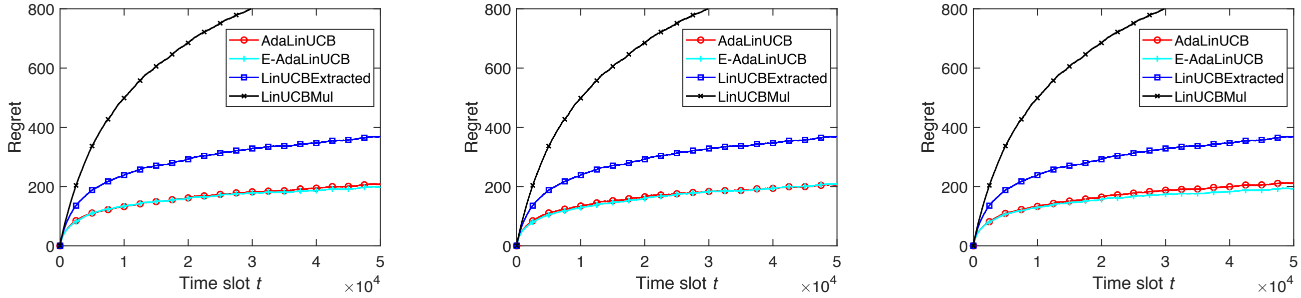
(c) AdaLinUCB: $l^{(-)} = l^{(-)}_{0.1}, l^{(+)} = l^{(+)}_{0.1}$; AdaLinUCB($l^{(-)} = l^{(+)}$): $l^{(-)} = l^{(+)} = 0.55$

Figure 4: Regret under beta distributed variation factor with a single threshold.



(a) $l^{(-)} = l^{(-)}_{0.05}, l^{(+)} = l^{(+)}_{0}$ (b) $l^{(-)} = l^{(-)}_{0.1}, l^{(+)} = l^{(+)}_{0}$ (c) $l^{(-)} = l^{(-)}_{0.15}, l^{(+)} = l^{(+)}_{0}$

Figure 5: Regret under beta distributed variation factor with different values of $l^{(-)}$.



(a) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.05}^{(+)}$ (b) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.1}^{(+)}$ (c) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.15}^{(+)}$

Figure 6: Regret under beta distributed variation factor with different values of $l^{(+)}$.

Algorithm 5 KernelUCB

```
Inputs: $\alpha \in \mathbb{R}_+$, $d \in \mathbb{N}$, $k(\cdot,\cdot)$, $\lambda = \lambda_{\text{regularization}}$.
for $t = 1, 2, 3, \cdots, T$ do
    Observe possible arm set $\mathcal{D}_t$, and observe associated context vectors $x_{t,a}, \forall a \in \mathcal{D}_t$.
    Observe $L_t$, and get augment context $\tilde{x}_{t,a} = [L_t, x_{t,a}^\top]^\top, \forall a \in \mathcal{D}_t$.
    if $t = 1$ then
        Choose the first actions $a_t \in \mathcal{D}_t$ (at start first action is pulled)
    else
        for $a \in \mathcal{D}_t$ do
            $k_{t,a} \leftarrow [k(\tilde{x}_{t,a}, \tilde{x}_{1,a_1}), k(\tilde{x}_{t,a}, \tilde{x}_{2,a_2}), \cdots, k(\tilde{x}_{t,a}, \tilde{x}_{t-1,a_{t-1}})]^\top$
            $\boldsymbol{K}_t \leftarrow$ kernel matrix of $(\tilde{x}_{1,a_1}, \cdots, \tilde{x}_{t-1,a_{t-1}})$
            $p_{t,a} \leftarrow k_{t,a}^\top [\boldsymbol{K}_t + \lambda \boldsymbol{I}]^{-1} y_{t-1} + \alpha \sqrt{k(\tilde{x}_{a,t}, \tilde{x}_{a,t}) - k_{t,a}^T [\boldsymbol{K}_t + \lambda \boldsymbol{I}]^{-1} k_{t,a}}$
        end for
        Choose action $a_t = \arg\max_{a \in \mathcal{D}_t} p_{t,a}$ with ties broken arbitrarily.
    end if
    Observe nominal reward $r_{t,a_t}$.
    $y_t \leftarrow [r_{1,a_1}, r_{2,a_2}, \cdots, r_{t,a_t}]^\top$
end for
```

addition, when the time horizon is even larger, the time to run KernelUCB can be prohibitively long. This is because KernelUCB needs more computation with more existing data samples. As a result, KernelUCB is not applicable for applications with large number of data samples in practice.

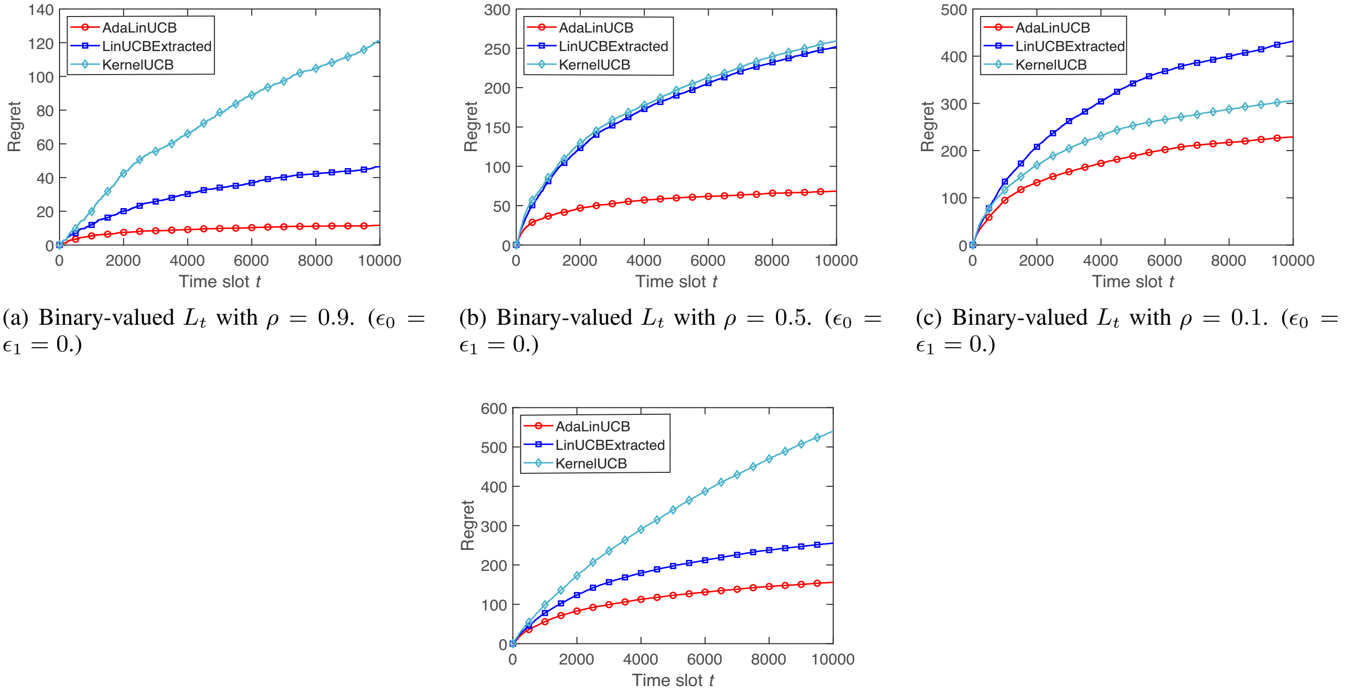
E.4 More for experiments on Yahoo! Today Module

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].

In Yahoo! Today Module, for each user visit, there are 10 candidate articles to be selected. The candidate articles are updated in a timely manner and are different for different time slots. Further, both the user and each of the candidate articles are associated with a 6-dimensional feature vector, which are generated by a conjoint analysis with a bilinear model [Chu *et al.*, 2009].

For the variation factor, we use a real trace - the sales of a popular store . It includes everyday turnover in two years [Rossman, 2015]. The normalized variation factor variation is demonstrated in Fig. 9.

Similarly to the experiments in Fig. 4, Fig. 5 and Fig. 6, we have evaluated the impact of of $l^{(-)}$ and $l^{(+)}$ in this data set in Fig. 9. We can see that the impact of threshold values for experiments on this real-world dataset is insignificant (when they are changing in a relatively large appropriate range) and the rewards of AdaLinUCB and E-AdaLinUCB are always higher than that of LinUCBExtracted and LinUCBMultiple.



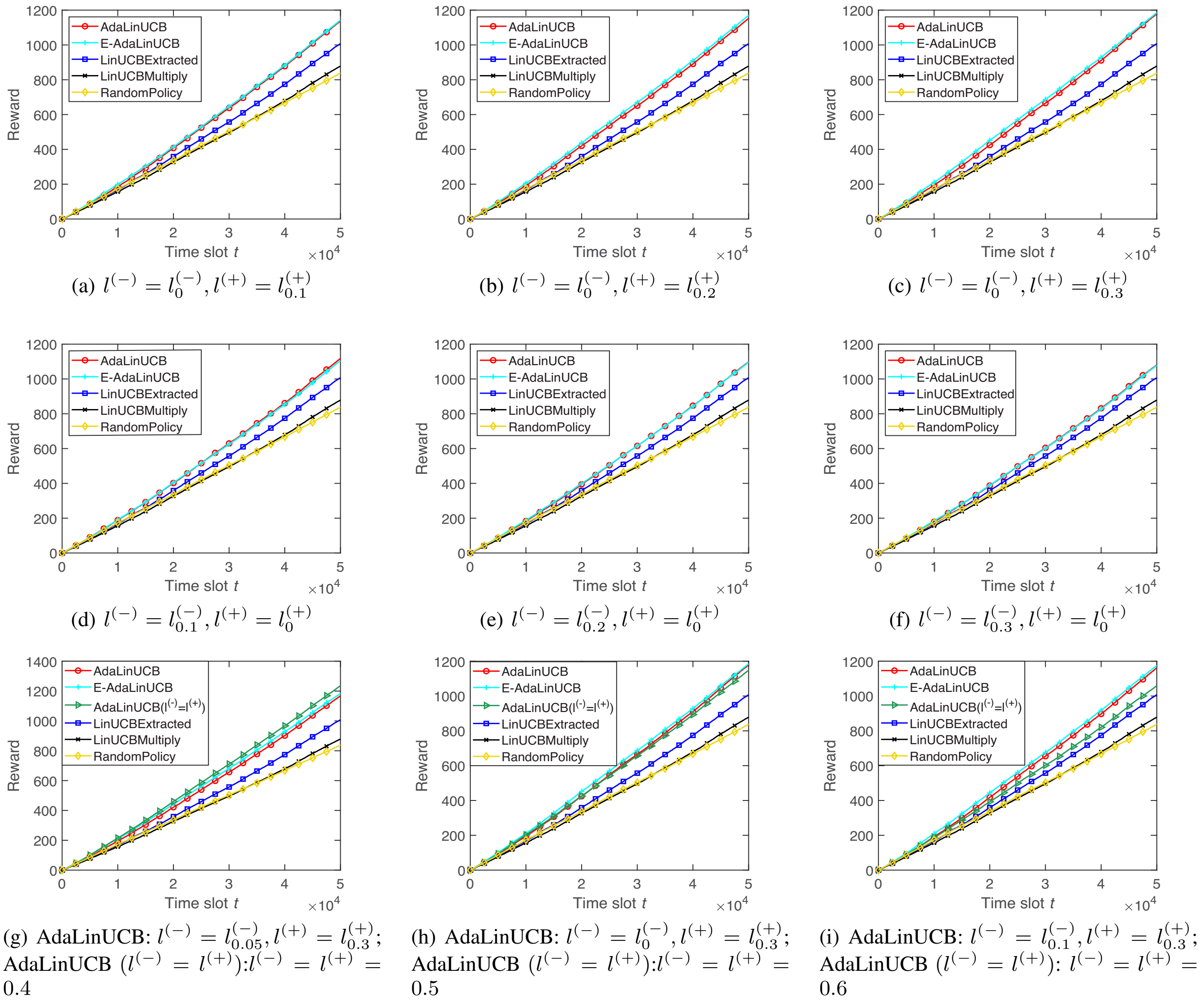
(a) Binary-valued L_t with $\rho = 0.9$. ($\epsilon_0 = \epsilon_1 = 0$.)

(b) Binary-valued L_t with $\rho = 0.5$. ($\epsilon_0 = \epsilon_1 = 0$.)

(c) Binary-valued L_t with $\rho = 0.1$. ($\epsilon_0 = \epsilon_1 = 0$.)

(d) Beta distributed variation factor; AdaLinUCB with $l^{(-)} = 0, l^{(+)} = l_0^{(+)}$.

Figure 7: Performance Comparison with KernelUCB.



(a) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.1}^{(+)}$

(b) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.2}^{(+)}$

(c) $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.3}^{(+)}$

(d) $l^{(-)} = l_{0.1}^{(-)}, l^{(+)} = l_0^{(+)}$

(e) $l^{(-)} = l_{0.2}^{(-)}, l^{(+)} = l_0^{(+)}$

(f) $l^{(-)} = l_{0.3}^{(-)}, l^{(+)} = l_0^{(+)}$

(g) AdaLinUCB: $l^{(-)} = l_{0.05}^{(-)}, l^{(+)} = l_{0.3}^{(+)}$; AdaLinUCB ($l^{(-)} = l^{(+)}$):$l^{(-)} = l^{(+)} = 0.4$

(h) AdaLinUCB: $l^{(-)} = l_0^{(-)}, l^{(+)} = l_{0.3}^{(+)}$; AdaLinUCB ($l^{(-)} = l^{(+)}$):$l^{(-)} = l^{(+)} = 0.5$

(i) AdaLinUCB: $l^{(-)} = l_{0.1}^{(-)}, l^{(+)} = l_{0.3}^{(+)}$; AdaLinUCB ($l^{(-)} = l^{(+)}$): $l^{(-)} = l^{(+)} = 0.6$

Figure 8: Performance comparison with different $l^{(-)}$ and $l^{(+)}$ values on Yahoo! Today Module.

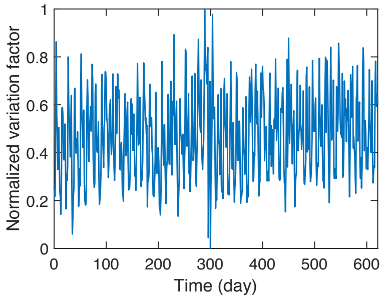


Figure 9: Normalized variation factor demonstration.