A Parameter-Free Algorithm for Misspecified Linear Contextual Bandits

Kei Takemura
NEC Corporation
kei_takemura@nec.com

Shinji Ito
NEC Corporation
i-shinji@nec.com

Daisuke Hatano
RIKEN AIP
daisuke.hatano@riken.jp

Hanna Sumita
Tokyo Institute of Technology
sumita@c.titech.ac.jp

Takuro Fukunaga
Chuo University and JST PRESTO
fukunaga.07s@g.chuo-u.ac.jp

Naonori Kakimura
Keio University
kakimura@math.keio.ac.jp

Ken-ichi Kawarabayashi
National Institute of Informatics
k_keniti@nii.ac.jp

Abstract

We investigate the misspecified linear contextual bandit (MLCB) problem, which is a generalization of the linear contextual bandit (LCB) problem. The MLCB problem is a decision-making problem in which a learner observes d-dimensional feature vectors, called arms, chooses an arm from K arms, and then obtains a reward from the chosen arm in each round. The learner aims to maximize the sum of the rewards over T rounds. In contrast to the LCB problem, the rewards in the MLCB problem may not be represented by a linear function in feature vectors; instead, it is approximated by a linear function with additive approximation parameter $\varepsilon \geq 0$. In this paper, we propose an algorithm that achieves $\tilde{O}(\sqrt{dT\log(K)}+\varepsilon\sqrt{d}T)$ regret, where $\tilde{O}(\cdot)$ ignores polylogarithmic factors in d and T. This is the first algorithm that guarantees a high-probability regret bound for the MLCB problem without knowledge of the approximation parameter ε.

1 INTRODUCTION

The linear contextual bandit (LCB) problem is a sequential decision-making problem in which a learner iterates the following process T times. First, the learner observes d-dimensional vectors called arms, where the number of the arms is K. Each arm offers a reward defined by a common linear function over the arms, but the reward is not revealed to the learner at this point. Then the learner chooses an arm. At the end, the learner observes the reward of the chosen arm. The learner aims to maximize the sum of the rewards. We measure the performance of an algorithm by its regret, which is the difference between the sum of the rewards of the optimal choices and that of the algorithm's choices.

Over the last decade, the LCB problem has been extensively studied both theoretically and practically (Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2013; Auer, 2002; Chapelle and Li, 2011; Chu et al., 2011; Dani et al., 2008; Dimakopoulou et al., 2019; Li et al., 2010). The LCB problem has several advantages over the multi-armed bandit (MAB) problem for real-world applications such as recommender systems. First, the LCB problem enables feature vectors (arms) to change in each round. This enables us to model recommender systems where news articles may frequently change (Li et al., 2010). Second, $\sqrt{K}$ in the regret bound of the MAB problem is replaced with $\sqrt{d\log(K)}$ or d in that of the LCB problem. Since K is often much larger than d in practice, the LCB problem gives better performance.

The misspecified linear contextual bandit (MLCB) problem has been studied in the recent years (Foster and Rakhlin, 2020; Ghosh et al., 2017; Gopalan et al., 2016; Lattimore et al., 2020). In this problem, there is a possibility that the rewards are not represented by any linear function in the feature vectors but *approximated* by a linear function where the approximation error is at most ε. Note that the MLCB problem when

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Table 1: Regret Bounds for MLCB Problem

Upper bound (known ε)	Upper bound (unknown ε)	Lower bound
$\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$	$\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$	$\Omega(d\sqrt{T})$
(Lattimore et al., 2020)	$\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$	(Lattimore and Szepesvári, 2020)
$\tilde{O}(\sqrt{dKT} + \varepsilon\sqrt{K}T)$	(**This work**)	$\Omega(\varepsilon\sqrt{d/\log(K)}\min(K,T))$
(Foster and Rakhlin, 2020)		(Lattimore et al., 2020)

$\varepsilon = 0$ is identical to the LCB problem. The misspecification (i.e., approximation error) enables us to formulate a more complicated reward function than a linear function, e.g., a reward function which may change over the rounds. Such cases usually appear in real-world applications such as education, healthcare, and recommender systems (Dimakopoulou et al., 2019).

The existing regret bounds for the MLCB problem are summarized in Table 1. Here, $\tilde{O}(\cdot)$ ignores polylogarithmic factors in d and T. Lattimore et al. (2020) designed a modified version of the LinUCB algorithm which achieves $\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$ regret. Foster and Rakhlin (2020) proposed an algorithm for a more general problem that requires an online optimization oracle. Their algorithm achieves $\tilde{O}(\sqrt{dKT} + \varepsilon\sqrt{K}T)$ regret for the MLCB problem when the Vovk-Azoury-Warmuth forecaster (Azoury and Warmuth, 2001; Vovk, 1998) is chosen as the oracle.[1] On the other hand, Lattimore et al. (2020) showed $\Omega(\varepsilon\sqrt{d/\log(K)}\min(K,T))$ regret for the MLCB problem, based on the results of Du et al. (2020). This means that the MLCB problem does not admit any sub-linear regret with respect to T, in contrast to the LCB problem.

Although these algorithms achieve near-optimal regret bounds for the MLCB problem, they are based on the assumption that we are given (an upper bound of) the approximation parameter ε in advance. This assumption is unavoidable in the algorithms; Lattimore et al. (2020) use ε to compute confidence intervals, and Foster and Rakhlin (2020) define a distribution to sample arms with ε. Thus, to apply the algorithms, we need to estimate (an upper bound of) the approximation parameter ε. In the real-world applications mentioned above, however, it is difficult to compute the approximation parameter ε. If we only obtain a loose upper bound of ε, then the regret of their algorithms deteriorates. In particular, the algorithms cannot achieve sub-linear regret when $\varepsilon = 0$ if we do not know the fact that $\varepsilon = 0$.

Our contribution is to propose the first algorithm for the MLCB problem that achieves $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$ regret without knowledge of the approximation parameter ε. The proposed algorithm is based on the SupLinUCB algorithm by Chu et al. (2011) for the LCB problem. The SupLinUCB algorithm introduces *stages*, which is used to reduce the set of arms in a round.[2] Specifically, as the stage progresses, the algorithm discards arms whose estimated rewards, together with confidence intervals, are less than a certain threshold. Our analysis reveals that the stages play an important role in bounding the estimation errors on the rewards without knowledge of the approximation parameter ε. We can show that the SupLinUCB algorithm achieves $\tilde{O}\left(\sqrt{d\log^2(K)T} + \varepsilon\sqrt{d\log(K)}T\right)$ regret. While the SupLinUCB algorithm achieves a near-optimal regret, the second term depends on the number of arms K, which is not optimal. To fill this gap, we propose two techniques for the SupLinUCB algorithm. One is to modify the conditions that determine whether or not the stage proceeds. This improves the bound of the estimation errors on the rewards. Specifically, the term with ε in the bound is no longer dependent on K and becomes $\tilde{O}(\varepsilon\sqrt{d})$. The other is to relax the threshold to reduce candidates for the chosen arm. On the basis of the improved estimation errors and the relaxed threshold, we show that a near-optimal arm that suffers $\tilde{O}(\varepsilon\sqrt{d})$ regret remains in the candidates during the stage progresses. This leads to $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$ regret in total. Note that by choosing a different parameter in our algorithm, it achieves $\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$ regret, which is minimax optimal if $\varepsilon = 0$. In contrast to the existing algorithms, the proposed algorithm automatically achieves the optimal regret if $\varepsilon = 0$, i.e., if a given instance is an instance of the LCB problem.

Very recently, a number of studies have proposed algorithms that achieve $\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$ regret for the MLCB problem without knowing the approximation

[1] This result does not contradict the lower bound of the LCB problem by Theorem 24.1 of Lattimore and Szepesvári (2020) because the lower bound holds when K is exponentially larger than d.

[2] This technique was originally proposed by Auer (2002) in the SupLinRel algorithm.

parameter ε (Foster et al., 2020; Pacchiano et al., 2020). We note that each algorithm deals with a more general problem. Our results are obtained independently of these studies, and our techniques are different, leading to a stronger regret bound for the MLCB problem. Specifically, whereas they showed the *expected* regret bounds with respect to the randomness in algorithms and the stochastic realization of the rewards, we show the *high-probability* regret bounds. Moreover, Pacchiano et al. (2020) assume that the sets of feature vectors are i.i.d. and the learner knows an upper bound of ε.

2 RELATED WORK

A special case of the MLCB problem is the non-contextual version, i.e., when given feature vectors are fixed over the rounds. Lattimore et al. (2020) showed that the elimination algorithm (Lattimore and Szepesvári, 2020) achieves $\tilde{O}(\sqrt{dT\log(K)}+\varepsilon\sqrt{d}T)$ regret without knowledge of the approximation parameter ε. The elimination algorithm, based on the experimental design technique, divides rounds into several stages and reduces the number of candidate arms to choose from as the stage progresses. In each stage, the algorithm computes a near-optimal design over the candidates and then chooses each arm in proportion to the design. From the resulting choices in the stage, the arms that seem sub-optimal are eliminated. The choices heavily depend on the assumption that the feature vectors are fixed. We remark that SupLinUCB and the proposed algorithms can be seen as a modification of the elimination algorithm to address the general MLCB problem.

A few studies have investigated the conditions under which sub-linear regret with respect to T is achievable for the non-contextual MLCB problem. Gopalan et al. (2016) showed that the OFUL algorithm (Abbasi-Yadkori et al., 2011) still enjoys a sub-linear regret if ε is very small. Ghosh et al. (2017) proposed an algorithm that achieves $\tilde{O}(d\sqrt{T}+\sqrt{KT})$ regret if ε is sufficiently large and most of the rewards of the arms cannot be represented by a common linear function. This algorithm tests the linearity of the rewards and then decides to use the OFUL algorithm or the UCB algorithm for the MAB problem. Meanwhile, Ghosh et al. (2017) showed that, for any algorithm that achieves the optimal regret $O(d\sqrt{T})$ for the LCB problem, an instance of the non-contextual MLCB problem can be constructed in which the algorithm suffers $\Omega(\varepsilon T)$ regret.

An alternative approach to overcoming the limitation of representing rewards by a linear function is the non-linear contextual bandit problem. In this problem, algorithms adaptively choose a policy, which is a map from the feature vectors to an arm, from given set of policies. Note that a reward function induces a policy that chooses the arm with the largest reward. The regret is defined as the difference between the sum of the rewards of choices by the best policy among given policies and that by the algorithms. Several studies have proposed computationally efficient algorithms that achieve sub-linear regret (Agarwal et al., 2014; Dudik et al., 2011; Foster et al., 2018; Foster and Rakhlin, 2020; Langford and Zhang, 2008). We note that the regret can be sub-linear even when given policies are induced by linear functions, but this fact does not contradicts to the lower bound by Lattimore et al. (2020). Generally, in agnostic setting (Agarwal et al., 2014; Dudik et al., 2011; Langford and Zhang, 2008), the definition of regret of the non-linear contextual bandit problem differs from that of the MLCB problem. These definitions coincide when the best policy achieves the optimal choices which maximize the sum of the rewards. Foster et al. (2018); Foster and Rakhlin (2020) assume that there exists a policy whose reward function can represent the true reward function.

3 PROBLEM SETTING

We formally define the MLCB problem. Let K denote the number of given arms. Each arm is indexed by an integer in $[K] := \{1, 2, \ldots, K\}$. The MLCB problem consists of T rounds. The learner proceeds with each round as follows. At the beginning of the t-th round, the learner observes the set of arms $\{x_t(i)\}_{i\in[K]} \subseteq \mathbb{R}^d$. Then, the learner chooses an arm $i_t \in [K]$. At the end of the round, the learner obtains reward $r_t(i_t)$, which is defined as follows.

We assume that the reward $r_t(i_t)$ has the expected value $\mu_t(i) = \mathbb{E}[r_t(i)]$ with R-sub-Gaussian noise. That is, we assume the following.

Assumption 1 (R-sub-Gaussian noise)**.** For all $t \in [T]$, $\eta_t = r_t(i_t) - \mu_t(i_t)$ is conditionally R-sub-Gaussian, i.e., for all $\lambda \in \mathbb{R}$,

$$\mathbb{E}\left[\exp(\lambda\eta_t) \mid \mathcal{F}_t\right] \leq \exp\left(\lambda^2 R^2/2\right),$$

where $\mathcal{F}_t = \sigma\left(\{x_s(i_s)\}_{s\in[t]}, \{\eta_s\}_{s\in[t-1]}\right)$.

Furthermore, we suppose that the expected reward $\mu_t(i)$ can be approximated by a linear function, where the approximation error is at most ε.

Assumption 2 (misspecified linear model)**.** There exist $\varepsilon \geq 0$ and $\theta \in \mathbb{R}^d$ such that for all $t \in [T]$ and $i \in [K]$,

$$\mu_t(i) = \theta^\top x_t(i) + \varepsilon_t(i),$$

where $|\varepsilon_t(i)| \leq \varepsilon$ for all $t \in [T]$ and $i \in [K]$.

Note that when $\varepsilon = 0$ is identical to the assumption in the LCB problem, and the unknown vector θ in Assumption 2 is generally not unique.

We evaluate the performance of an algorithm by the regret $R(T)$, which is defined as

$$R(T) = \sum_{t\in[T]} (\mu_t(i_t^*) - \mu_t(i_t)),$$

where $i_t^* \in \mathrm{argmax}_i \mu_t(i)$.

In addition, we define the following parameters of this problem: (i) $L > 0$ such that $\forall i \in [K]$ and $\forall t \in [T]$, $\|x_t(i)\|_2 \le L$, (ii) $M > 0$ such that $\|\theta\|_2 \le M$, and (iii) $B > 0$ such that $\forall i, j \in [K]$ and $\forall t \in [T]$, $|\mu_t(i) - \mu_t(j)| \le B$.

4 PROPOSED ALGORITHM

In this section, we propose an algorithm that achieves $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$ regret when the approximation parameter ε is not given. The proposed algorithm (Algorithm 1) is based on the SupLinUCB algorithm (Chu et al., 2011) for the LCB problem. In each round $t \in [T]$, our algorithm repeatedly reduces the set of arms in each *stage*, until an arm is chosen from the set. At the beginning of the s-th stage in round t, the algorithm estimates the reward using $\hat{r}_{t,s}(i)$ and its confidence interval using $w_{t,s}(i)$ for each arm $i \in I_{t,s}$, where $I_{t,s} \subseteq [K]$ denotes the set of remaining arms (lines 7–13). Here α is a parameter defined later, and we denote $\|x\|_V = \sqrt{x^\top V x}$ for any $x \in \mathbb{R}^d$ and any positive definite matrix $V \in \mathbb{R}^{d\times d}$. Then, the algorithm decides whether to proceed to the next stage based on the confidence intervals $w_{t,s}(i)$ of the arms $i \in I_{t,s}$. If all $w_{t,s}(i)$'s are smaller than $\alpha\sqrt{d/T}$, the algorithm chooses the arm that has the largest upper confidence bound of the estimated reward (line 15). If $w_{t,s}(i)$'s do not satisfy the condition above but are smaller than αc^{-s}, the algorithm proceeds to the next stage. In the next stage, our algorithm keeps the arms $i \in I_{t,s}$ that satisfy the following threshold:

$$\hat{r}_{t,s}(i) + w_{t,s}(i) \ge \max_{i'\in I_{t,s}} (\hat{r}_{t,s}(i') + w_{t,s}(i')) - 2\alpha c^{-s}, \tag{1}$$

and discards the other arms (line 18). Note that since the threshold in line 17 decreases exponentially as the stage progresses, it follows from the threshold in line 14 that the number of stages in each round is at most $S := \lceil \log_c(T/d)/2 \rceil$. If an arm i has confidence interval $w_{t,s}(i)$ larger than αc^{-s}, our algorithm chooses an arm with a large confidence interval (line 21), and keeps the current round t in the set $\Psi_{t,s}$ (line 22). The chosen arm i_t and the observed reward $r_t(i_t)$ of round $t \in \Psi_{t,s}$ will be used to compute $\hat{r}_{t',s}(i)$ and $w_{t',s}(i)$ for each arm $i \in I_{t',s}$ in stage s of the round $t' > t$, whereas the arm chosen in line 15 will not be used.

Algorithm 1 Proposed algorithm (a modified version of SupLinUCB (Chu et al., 2011))

Input: $T > 0$, $\lambda > 0$, $\alpha > 0$, and $c > 1$.
1: Let $S = \lceil \log_c(T/d)/2 \rceil$.
2: $\Psi_{1,s} \leftarrow \emptyset$ for $s \in [S]$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Observe $\{x_t(i)\}_{i\in[K]}$.
5: $s \leftarrow 1$ and $I_{t,1} \leftarrow [K]$.
6: **repeat**
7: $V_{t-1,s} \leftarrow \lambda I + \sum_{\tau\in\Psi_{t,s}} x_\tau(i_\tau)x_\tau(i_\tau)^\top$.
8: $b_{t-1,s} \leftarrow \sum_{\tau\in\Psi_{t,s}} r_\tau(i_\tau)x_\tau(i_\tau)$.
9: $\hat{\theta}_{t,s} \leftarrow V_{t-1,s}^{-1} b_{t-1,s}$.
10: **for** $i \in I_{t,s}$ **do**
11: $\hat{r}_{t,s}(i) \leftarrow \hat{\theta}_{t,s}^\top x_t(i)$.
12: $w_{t,s}(i) \leftarrow \alpha\|x_t(i)\|_{V_{t-1,s}^{-1}}$.
13: **end for**
14: **if** $w_{t,s}(i) \le \alpha\sqrt{d/T}$ for all $i \in I_{t,s}$ **then**
15: $i_t \in \mathrm{argmax}_{i\in I_{t,s}}(\hat{r}_{t,s}(i) + w_{t,s}(i))$.
16: $\Psi_{t+1,s'} \leftarrow \Psi_{t,s'}$ for all $s' \in [S]$.
17: **else if** $w_{t,s}(i) \le \alpha c^{-s}$ for all $i \in I_{t,s}$ **then**
18: $I_{t,s+1} \leftarrow$ arms that satisfy (1).
19: $s \leftarrow s + 1$.
20: **else**
21: Choose $i_t \in I_{t,s}$ s.t. $w_{t,s}(i) > \alpha c^{-s}$.
22: $\Psi_{t+1,s} \leftarrow \Psi_{t,s} \cup \{t\}$.
23: $\Psi_{t+1,s'} \leftarrow \Psi_{t,s'}$ for all $s' \in [S] \setminus \{s\}$.
24: **end if**
25: **until** an arm i_t is chosen.
26: Observe $r_t(i_t)$.
27: **end for**

The crucial difference between the proposed and the SupLinUCB algorithm is that our algorithm introduces α in lines 14, 17, 18, and 21. In fact, our algorithm coincides with the SupLinUCB algorithm if we set $\alpha = 1$ and $c = 2$ in those lines.[3] Note that in this case, the parameter α remains to define $w_{t,s}(i)$. The difference improves the regret bound by a $\tilde{\Theta}(\sqrt{\log(K)})$ factor, as will be seen in the next section.

5 REGRET ANALYSIS

In this section, we show that the proposed algorithm has $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$ regret. We define $\beta(\delta) = R\sqrt{2\log(2KST/\delta)} + \sqrt{\lambda}M$.

Theorem 1. If $\alpha = \beta(\delta)$, $c - 1 = \Theta(1)$, and $\lambda =$

[3] Strictly speaking, the RHS in line 14 is $1/\sqrt{T}$ in the SupLinUCB algorithm, but this minor difference does not affect the regret bound of Chu et al. (2011).

R^2M^{-2}, then Algorithm 1 has the following regret bound with probability $1-\delta$:

$$R(T) = \tilde{O}\left(R\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T + Bd\right),$$

where $\tilde{O}(\cdot)$ ignores the polylogarithmic factors in d and T.

We remark that Theorem 1 holds for any pair of θ and ε that satisfies Assumption 2, which means that the regret bound in Theorem 1 is obtained for the smallest ε that satisfies Assumption 2. In particular, we can achieve a sub-linear regret when $\varepsilon = 0$, i.e., when a given instance is that of the LCB problem, even if we do not know this fact.

To prove Theorem 1, we first decompose the regret based on $\Psi_{T+1,s}$'s. Let Ψ_0 be the set of rounds that succeeds to choose an arm with small confidence interval (lines 14–16), i.e., $\Psi_0 = [T] \setminus \bigcup_{s\in[S]} \Psi_{T+1,s}$. Then the regret can be decomposed as follows:

$$R(T) = R^{first}(T) + R^{main}(T) + R^{confident}(T),$$

where

$$R^{first}(T) = \sum_{t\in\Psi_{T+1,1}} (\mu_t(i_t^*) - \mu_t(i_t)),$$

$$R^{main}(T) = \sum_{s=2}^{S} \sum_{t\in\Psi_{T+1,s}} (\mu_t(i_t^*) - \mu_t(i_t)), \text{ and}$$

$$R^{confident}(T) = \sum_{t\in\Psi_0} (\mu_t(i_t^*) - \mu_t(i_t)).$$

In what follows, we bound each term.

The term $R^{first}(T)$ is the regret of the rounds when the algorithm chooses an arm with a large confidence interval in the first stage. We will show $R^{first}(T) = \tilde{O}(d)$ in the next subsection by proving $|\Psi_{T+1,1}| = \tilde{O}(d)$ (Lemma 1).

The main part of the proof is to bound the term $R^{main}(T)$, as $R^{confident}(T)$ can be bounded in a similar way. As mentioned in the previous section, we introduce α when we determine whether to proceed to the next stage (lines 14, 17, and 21). This enables us to bound the estimation error of $\hat{r}_{t,s}(i)$ by $\beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}} + \tilde{O}(\varepsilon\sqrt{d})$ (Lemma 2). On the basis of Lemma 2, with the threshold (1), we show that our algorithm keeps a near-optimal arm that suffers $\tilde{O}(\varepsilon\sqrt{d})$ regret as the stage progresses (Lemma 4). This implies that the regret in a round is at most $5\beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}} + \tilde{O}(\varepsilon\sqrt{d})$ (Lemma 6). Summing up the regrets from all rounds, we obtain a desired regret bound $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$. The details of our proof are given in the following subsections, and additional proofs may be found in the appendix.

Before describing the details of our analysis, we compare our analysis with that of Chu et al. (2011). We may be able to perform the SupLinUCB algorithm for the MLCB problem, but the regret bound would be worse. Suppose that $\alpha = \beta(\delta)$, similarly to our algorithm. Since the SupLinUCB algorithm does not use α in the thresholds of the algorithm, we cannot use Lemma 1. Instead, we use Lemma 6 from Chu et al. (2011), which implies $R^{first}(T) = \tilde{O}(\sqrt{dT\log(K)^2})$. Moreover, by using the Lemma, the estimation error of $\hat{r}_{t,s}(i)$ is bounded by $\beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}} + \tilde{O}(\varepsilon\sqrt{d\log(K)})$, which is worse than that by Lemma 2. Therefore, the regret bound in total will be $\tilde{O}(\sqrt{dT\log(K)^2} + \varepsilon\sqrt{d\log(K)}T)$.

5.1 Bound of $R^{first}(T)$

We bound the regret when an arm with large confidence interval is chosen in the first stage. Since the regret in a round is at most B, we can bound $R^{first}(T)$ by bounding the number of chosen arms in the first stage.

Lemma 1. For all $t \in [T]$ and $s \geq 1$, we have

$$|\Psi_{t,s}| \leq 2c^{2s}d\log(1 + L^2|\Psi_{t,s}|/(d\lambda)).$$

Using Lemma 1 and the assumption that $c = \Theta(1)$, we obtain

$$\begin{aligned} R^{first}(T) &\leq B|\Psi_{T+1,1}| \\ &\leq 2c^2Bd\log(1 + L^2|\Psi_{T+1,1}|/(d\lambda)) \\ &= O(Bd\log(1 + L^2T/(d\lambda)). \end{aligned}$$

5.2 Bound of $R^{main}(T)$

We bound $R^{main}(T)$, which is the main part of our regret analysis. We first show that the estimation error of $\hat{\theta}_{t,s}$ is at most $\beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}} + \tilde{O}(\varepsilon\sqrt{d})$. Let s_t be the stage in which i_t is chosen for all $t \in [T]$. Recall that $S = \lceil \log_c(T/d)/2 \rceil$.

Lemma 2. For all $\delta \in (0,1)$, with probability at least $1 - \delta(S-1)/S$, for all $t \in [T]$, $s \in [s_t - 1]$, and $i \in I_{t,s}$, we have

$$\begin{aligned} &|(\hat{\theta}_{t,s} - \theta)^\top x_t(i)| \\ &\leq \beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}} + \varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}. \end{aligned} \quad (2)$$

Our confidence bound relies on the following lemma.

Lemma 3. Let $\tilde{\theta}_{t,s} = V_{t-1,s}^{-1}\sum_{\tau\in\Psi_{t,s}}(\theta^\top x_\tau(i_\tau) + \eta_\tau)x_\tau(i_\tau)$ for all $t \in [T]$ and $s \in [S]$. For all $\delta \in (0,1)$, for any $s \in [S]$, $t \in [T]$, and $i \in I_{t,s}$, with probability at least $1 - \delta/(KST)$, we have

$$\left|(\tilde{\theta}_{t,s} - \theta)^\top x_t(i)\right| \leq \beta(\delta)\|x_t(i)\|_{V_{t-1,s}^{-1}}.$$

Proof of Lemma 2. We arbitrarily fix $t \in [T]$, $s \in [s_t - 1]$, and $i \in I_{t,s}$. From the definition of $\hat{\theta}_{t,s}$ and $b_{t,s}$, we have

$$\begin{aligned}
&|(\hat{\theta}_{t,s} - \theta)^\top x_t(i)| \\
&= \left| \left(V_{t-1,s}^{-1} b_{t-1,s} - \theta\right)^\top x_t(i) \right| \\
&= \left| \left(V_{t-1,s}^{-1} \sum_{\tau \in \Psi_{t,s}} r_\tau(i_\tau) x_\tau(i_\tau) - \theta \right)^\top x_t(i) \right|.
\end{aligned}$$

Then, from Assumption 2 and the definition of $\tilde{\theta}_{t,s}$, we have

$$\begin{aligned}
&\left| \left(V_{t-1,s}^{-1} \sum_{\tau \in \Psi_{t,s}} r_\tau(i_\tau) x_\tau(i_\tau) - \theta \right)^\top x_t(i) \right| \\
&\le \left| \left(\tilde{\theta}_{t,s} - \theta\right)^\top x_t(i) \right| \qquad (3) \\
&+ \left| \left(V_{t-1,s}^{-1} \sum_{\tau \in \Psi_{t,s}} \varepsilon_\tau(i_\tau) x_\tau(i_\tau) \right)^\top x_t(i) \right|. \qquad (4)
\end{aligned}$$

Applying Lemma 3 to the term (3), we have

$$\left| \left(\tilde{\theta}_{t,s} - \theta\right)^\top x_t(i) \right| \le \beta(\delta) \|x_t(i)\|_{V_{t-1,s}^{-1}}$$

with probability at least $1 - \delta/(KST)$. Taking the union bound over rounds, stages, and arms, with probability at least $1 - \delta(S-1)/S$, the above bound holds for all $t \in [T]$, $s \in [s_t - 1]$, and $i \in I_{t,s}$. For the term (4), from Assumption 2, we have

$$\begin{aligned}
&\left| \left(V_{t-1,s}^{-1} \sum_{\tau \in \Psi_{t,s}} \varepsilon_\tau(i_\tau) x_\tau(i_\tau) \right)^\top x_t(i) \right| \\
&= \left| \sum_{\tau \in \Psi_{t,s}} \varepsilon_\tau(i_\tau) x_\tau(i_\tau)^\top V_{t-1,s}^{-1} x_t(i) \right| \\
&\le \varepsilon \sum_{\tau \in \Psi_{t,s}} \left| x_\tau(i_\tau)^\top V_{t-1,s}^{-1} x_t(i) \right|.
\end{aligned}$$

Then, applying the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\varepsilon \sum_{\tau \in \Psi_{t,s}} \left| x_\tau(i_\tau)^\top V_{t-1,s}^{-1} x_t(i) \right| \\
&\le \varepsilon \sqrt{|\Psi_{t,s}| \sum_{\tau \in \Psi_{t,s}} \left(x_\tau(i_\tau)^\top V_{t-1,s}^{-1} x_t(i) \right)^2} \\
&= \varepsilon \sqrt{|\Psi_{t,s}| x_t(i)^\top V_{t-1,s}^{-1} (V_{t-1,s} - \lambda I) V_{t-1,s}^{-1} x_t(i)} \\
&\le \sqrt{|\Psi_{t,s}| x_t(i)^\top V_{t-1,s}^{-1} x_t(i)}.
\end{aligned}$$

It follows from $s < s_t$ that $x_t(i')^\top V_{t-1,s}^{-1} x_t(i') \le c^{-2s}$ for all $i' \in I_{t,s}$. Thus, we obtain

$$\varepsilon \sqrt{|\Psi_{t,s}| x_t(i)^\top V_{t-1,s}^{-1} x_t(i)} \le \varepsilon \sqrt{|\Psi_{t,s}| c^{-2s}}.$$

Applying Lemma 1 to the above, we finish the proof. □

It is worth mentioning that, if we construct an estimator from all observations (i.e., an estimator without producing stages) without knowledge of ε, then the estimation error becomes larger due to the misspecification. When we consider an estimator with a single stage, it seems hard to bound the term (4) by $\tilde{O}(\varepsilon\sqrt{d})$ because $|\Psi_{t,1}| = t-1$ and it cannot show that $\|x_t(i_t)\|^2_{V_{t-1,1}^{-1}}$ is $\tilde{O}(d/t)$ for any $t \in [T]$ and any sequence $\{x_{t'}(i_{t'})\}_{t' \in [t]}$. The instance in Appendix E of Lattimore et al. (2020) shows that an estimator that uses all observations has $\Omega(1)$ estimation error after $\Theta(T)$ rounds when $d = O(1)$ and $\varepsilon = \Theta(T^{-1/4})$ while $\beta_t \|x_t(i)\|_{V_{t-1,s}^{-1}} + \varepsilon\sqrt{d} = \tilde{O}(T^{-1/4})$ in that setting, where $\beta_t = \tilde{O}(\sqrt{d})$. We overcome this by introducing stages.

For the special case of the MLCB problem when given arms do not change over the rounds, Lattimore et al. (2020) analyzed the elimination algorithm. They showed that the dependence on ε in the estimation errors is $\tilde{O}(\varepsilon\sqrt{d})$ without knowledge of ε. The key ingredient of their proof is to control the number of rounds and the confidence intervals in a stage so that they are not too large (see Proposition 5.1 of Lattimore et al. (2020) for details). One of our contributions is to design stages to have this property for the general MLCB problem. Specifically, we define thresholds of confidence intervals so that $|\Psi_{T+1,s}| \|x_t(i)\|^2_{V_{t-1,s}^{-1}} = \tilde{O}(d)$, where $|\Psi_{T+1,s}|$ corresponds to the number of rounds in a stage of the elimination algorithm. Thus, our algorithm can be seen as a modification of the elimination algorithm for the MLCB problem.

Before proceeding with our analysis, we define the probabilistic event that we use in our proofs.

Definition 1. We define E_t as the event where the estimation error can be bounded as in Lemma 2, i.e.,

$$E_t = \{\forall s \in [s_t - 1], \forall i \in I_{t,s},\ (2) \text{ holds.}\}.$$

We next show that, due to our threshold (1) and Lemma 2, a near-optimal arm i such that $\mu_t(i_t^*) - \mu_t(i) = \tilde{O}(\varepsilon\sqrt{d})$ remains in $I_{t,s}$ for all $t \in [T]$ and $s \in [s_t]$. This is in contrast to the SupLinUCB algorithm, which guarantees that $i_t^* \in I_{t,s}$ for the LCB problem. Let $i_{t,s}^* \in \operatorname{argmax}_{i \in I_{t,s}} \mu_t(i)$ for all $t \in [T]$ and $s \le s_t$.

Lemma 4. For all $t \in [T]$, under the event E_t, we have

$$\begin{aligned}&\mu_t(i_t^*) - \mu_t(i_{t,s}^*)\\ &\leq 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))})(s-1)\end{aligned}$$

for all $s \in [s_t]$.

Proof. We prove this lemma by induction. We fix $t \in [T]$ arbitrarily. For $s = 1$, since $i_t^* = i_{t,1}^*$, we have the bound. Assume that the bound holds in a stage $s < s_t$. It is sufficient to show that $\mu_t(i_{t,s}^*) - \mu_t(i_{t,s+1}^*) \leq 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))})$. If $i_{t,s}^* = i_{t,s+1}^*$, the desired bound holds. Hence, we assume that $i_{t,s}^* \notin I_{t,s+1}$. Let $\hat{i}_{t,s} \in \mathrm{argmax}_{i \in I_{t,s}}(\hat{r}_{t,s}(i) + w_{t,s}(i))$. From the fact that $\hat{i}_{t,s} \in I_{t,s+1}$ and Assumption 2, we have

$$\begin{aligned}\mu_t(i_{t,s}^*) - \mu_t(i_{t,s+1}^*) &\leq \mu_t(i_{t,s}^*) - \mu_t(\hat{i}_{t,s})\\ &\leq \theta^\top(x_t(i_{t,s}^*) - x_t(\hat{i}_{t,s})) + 2\varepsilon.\end{aligned} \tag{5}$$

Then, from the definition of E_t, we obtain

$$\begin{aligned}&\theta^\top(x_t(i_{t,s}^*) - x_t(\hat{i}_{t,s}))\\ &\leq \hat{r}_{t,s}(i_{t,s}^*) + w_{t,s}(i_{t,s}^*) - (\hat{r}_{t,s}(\hat{i}_{t,s}) - w_{t,s}(\hat{i}_{t,s}))\\ &+ 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}\\ &= \hat{r}_{t,s}(i_{t,s}^*) + w_{t,s}(i_{t,s}^*) - (\hat{r}_{t,s}(\hat{i}_{t,s}) + w_{t,s}(\hat{i}_{t,s}))\\ &+ 2w_{t,s}(\hat{i}_{t,s}) + 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}.\end{aligned}$$

Since $s < s_t$, we have $w_{t,s}(i_{t,s}) \leq \beta(\delta)c^{-s}$ for all $i \in I_{t,s}$. Thus, from the assumption that $i_{t,s}^* \notin I_{t,s+1}$ and the threshold (1), we have

$$\begin{aligned}&2w_{t,s}(\hat{i}_{t,s})\\ &\leq 2\beta(\delta)c^{-s}\\ &< \hat{r}_{t,s}(\hat{i}_{t,s}) + w_{t,s}(\hat{i}_{t,s}) - (\hat{r}_{t,s}(i_{t,s}^*) + w_{t,s}(i_{t,s}^*))\end{aligned}$$

and we can obtain

$$\theta^\top(x_t(i_{t,s}^*) - x_t(\hat{i}_{t,s})) \leq 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}.$$

Substituting this into (5) completes the proof. □

In the following lemma, using the threshold (1) and Lemma 2 again, we show that the largest difference of the arms' rewards in the set $I_{t,s}$ is $\tilde{O}(c^{-s})\beta(\delta) + \tilde{O}(\varepsilon\sqrt{d})$ for stages such that $s \geq 2$.

Lemma 5. For all $t \in [T]$, under the event E_t, we have

$$\begin{aligned}&\theta^\top(x_t(i_{t,s}^*) - x_t(i))\\ &\leq 5\beta(\delta)c^{1-s} + 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}\end{aligned}$$

for all s such that $2 \leq s \leq s_t$ and $i \in I_{t,s}$.

Proof. We fix $t \in [T]$, s such that $2 \leq s \leq s_t$, and $i \in I_{t,s}$ arbitrarily. From the definition of $I_{t,s}$, we have

$$\begin{aligned}&\hat{r}_{t,s-1}(i_{t,s}^*) + w_{t,s-1}(i_{t,s}^*) - (\hat{r}_{t,s-1}(i) + w_{t,s-1}(i))\\ &\leq 2\beta(\delta)c^{1-s}.\end{aligned}$$

Since $s - 1 < s_t$, we have $0 \leq w_{t,s-1}(i) \leq \beta(\delta)c^{1-s}$ for all $i \in I_{t,s-1}$. Thus, we have

$$\begin{aligned}&\hat{\theta}_{t,s-1}^\top(x_t(i_{t,s}^*) - x_t(i))\\ &\leq (w_{t,s-1}(i) - w_{t,s-1}(i_{t,s}^*)) + 2\beta(\delta)c^{1-s}\\ &\leq 3\beta(\delta)c^{1-s}.\end{aligned}$$

Then, from the definition of E_t, we have

$$\begin{aligned}3\beta(\delta)c^{1-s} &\geq \hat{\theta}_{t,s-1}^\top(x_t(i_{t,s}^*) - x_t(i))\\ &\geq \theta^\top(x_t(i_{t,s}^*) - x_t(i))\\ &- \beta(\delta)(\|x_t(i_{t,s}^*)\|_{V_{t,s-1}^{-1}} + \|x_t(i)\|_{V_{t,s-1}^{-1}})\\ &- 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))}.\end{aligned}$$

Since $\|x_t(i')\|_{V_{t,s-1}^{-1}} \leq c^{1-s}$ for all $i' \in I_{t,s}$, we obtain the desired result. □

Combining Lemma 4 and Lemma 5, we can obtain an upper bound of the regret in a round.

Lemma 6. For all $t \in [T]$ such that $s_t \geq 2$, under the event E_t, we have

$$\begin{aligned}&\mu_t(i_t^*) - \mu_t(i)\\ &\leq 5\beta(\delta)c^{1-s_t} + 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))})s_t\end{aligned}$$

for all $i \in I_{t,s_t}$.

Proof. We arbitrarily fix $t \in [T]$ such that $s_t \geq 2$ and $i \in I_{s_t}$. It follows from Assumption 2 and Lemma 5 that

$$\begin{aligned}\mu_t(i) &\geq \theta^\top x_t(i) - \varepsilon\\ &\geq \theta^\top x_t(i_{t,s_t}^*) - 5\beta(\delta)c^{1-s_t}\\ &- 2\varepsilon\sqrt{2d\log(1 + L^2t/(d\lambda))} - \varepsilon\\ &\geq \mu_t(i_{t,s_t}^*) - 5\beta(\delta)c^{1-s_t}\\ &- 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))}).\end{aligned}$$

From Lemma 4, we obtain

$$\begin{aligned}&\mu_t(i_{t,s_t}^*) - 5\beta(\delta)c^{1-s_t}\\ &- 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))})\\ &\geq \mu_t(i_t^*) - 5\beta(\delta)c^{1-s_t}\\ &- 2\varepsilon(1 + \sqrt{2d\log(1 + L^2t/(d\lambda))})s_t.\end{aligned}$$

□

We can now bound $R^{main}(T)$. Let $E = \bigcup_{t\in[T]} E_t$. From Lemma 2, we have $\mathbb{P}(E) \geq 1-\delta(S-1)/S$. Conditioned on the event E, from Lemma 6, we have

$$\sum_{s=2}^{S} \sum_{t\in\Psi_{T+1,s}} (\mu_t(i_t^*) - \mu_t(i_t))$$
$$\leq \sum_{s=2}^{S} 5\beta(\delta)c^{1-s}|\Psi_{T+1,s}| \quad (6)$$
$$+ \sum_{s=2}^{S} 2\varepsilon(1+2\sqrt{d\log(1+L^2T/(d\lambda))})s|\Psi_{T+1,s}|. \quad (7)$$

We bound the term (6). From Lemma 1, we have

$$|\Psi_{T+1,s}| \leq \sqrt{2c^s d\log(1+L^2T/(d\lambda))|\Psi_{T+1,s}|}.$$

Applying this to the term (6), we have

$$\sum_{s=2}^{S} 5\beta(\delta)c^{1-s}|\Psi_{T+1,s}|$$
$$\leq 5\beta(\delta)c\sqrt{2d\log(1+L^2T/(d\lambda))}\sum_{s=2}^{S}\sqrt{|\Psi_{T+1,s}|}.$$

Using the Cauchy-Schwarz inequality, we have

$$\sum_{s=2}^{S}\sqrt{|\Psi_{T+1,s}|} \leq \sqrt{S\sum_{s=2}^{S}|\Psi_{T+1,s}|}$$
$$\leq \sqrt{ST}.$$

Recall that $S = \lceil \log_c(T/d)/2 \rceil$ and $\beta(\delta) = \tilde{O}(R\sqrt{\log(K)})$. From the assumption $c-1 = \Theta(1)$, there exists a universal constant $C > 0$ such that $\frac{1}{2}\log_c(T/d) \leq C\log(T/d)$. This implies that $S = O(\log(T))$. Thus, we have

$$\sum_{s=2}^{S} 5\beta(\delta)c^{1-s}|\Psi_{T+1,s}| = \tilde{O}(R\sqrt{dT\log(K)}).$$

Furthermore, from the fact that $\sum_{s\in[S]}|\Psi_{T+1,s}| \leq T$, we have that the term (7) is $\tilde{O}\left(\varepsilon\sqrt{d}T\right)$. Hence, we obtain

$$R^{main}(T) = \tilde{O}\left(R\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T\right)$$

with probability at least $1-\delta(S-1)/S$.

5.3 Bound of $R^{confident}(T)$

Lastly, we bound $R^{confident}(T)$. We first show an analogy of Lemma 2. We can prove this lemma in a similar way to the proof for Lemma 2.

Lemma 7. We have

$$|(\hat{\theta}_{t,s_t} - \theta)^\top x_t(i)| \leq \beta(\delta)\|x_t(i)\|_{V_{t-1,s_t}^{-1}} + \varepsilon\sqrt{d}$$

for all $t \in \Psi_0$ and $i \in I_{t,s_t}$, with probability at least $1-\delta/S$.

Using Lemma 7, we obtain an analogy of Lemma 5.

Lemma 8. We have

$$\mu_t(i_{t,s_t}^*) - \mu_t(i_t) \leq 2\beta(\delta)\sqrt{d/T} + 2\varepsilon(1+\sqrt{d})$$

for all $t \in \Psi_0$, with probability at least $1-\delta/S$.

We now bound $R^{confident}(T)$. We can decompose $R^{confident}(T)$ as follows:

$$R^{confident}(T) = \sum_{t\in\Psi_0}(\mu_t(i_t^*) - \mu_t(i_{t,s_t}^*)) \quad (8)$$
$$+ \sum_{t\in\Psi_0}(\mu_t(i_{t,s_t}^*) - \mu_t(i_t)). \quad (9)$$

Since $|\Psi_0| \leq T$, by applying Lemma 4 and Lemma 8 to (8) and (9), respectively, we obtain

$$R^{confident}(T) = \tilde{O}\left(R\sqrt{dT} + \varepsilon\sqrt{d}T\right)$$

with probability at least $1-\delta/S$.

6 CONCLUSION

We proposed the first parameter-free algorithm that achieves $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$ regret for the MLCB problem. Similar to the SupLinUCB algorithm, the proposed algorithm reduces the set of arms as the stage progresses. By introducing the parameter α to the conditions that determine the arms in each stage, we improved the $\tilde{\Theta}(\sqrt{\log(K)})$ factor from the regret of the SupLinUCB algorithm. More precisely, we showed that our algorithm keeps a near-optimal arm that suffers $\tilde{O}(\varepsilon\sqrt{d})$ regret in any stage, and thus the total regret bound is $\tilde{O}(\sqrt{dT\log(K)} + \varepsilon\sqrt{d}T)$.

We note that our analysis of the proposed algorithm with a different parameter α gives an alternative regret bound. Specifically, when we adopt $\alpha = R\sqrt{d\log\left(\frac{1+STL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}M$, our algorithm achieves $\tilde{O}(d\sqrt{T} + \varepsilon\sqrt{d}T)$ regret, which is optimal if $\varepsilon = 0$. To prove this bound, we use Theorem 2 in Abbasi-Yadkori et al. (2011) for each stage instead of Lemma 3 and follow the same line of the proof of Theorem 1. Theorem 1 matches the result when K is exponentially larger than d.

Acknowledgements

SI was supported by JST, ACT-I, Grant Number JPMJPR18U5, Japan. TF was supported by JST, PRESTO, Grant Number JPMJPR1759, Japan. NK and KK were supported by JSPS, KAKENHI, Grant Number JP18H05291, Japan.

References

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

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1638–1646.

Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 127–135.

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

Azoury, K. S. and Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246.

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366.

Dimakopoulou, M., Zhou, Z., Athey, S., and Imbens, G. (2019). Balanced linear contextual bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3445–3453.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. (2020). Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*.

Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., and Zhang, T. (2011). Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 169–178.

Foster, D., Agarwal, A., Dudik, M., Luo, H., and Schapire, R. (2018). Practical contextual bandits with regression oracles. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1539–1548.

Foster, D. and Rakhlin, A. (2020). Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3199–3210.

Foster, D. J., Gentile, C., Mohri, M., and Zimmert, J. (2020). Adapting to misspecification in contextual bandits. In *Advances in Neural Information Processing Systems*, pages 11478–11489.

Ghosh, A., Chowdhury, S. R., and Gopalan, A. (2017). Misspecified linear bandits. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3761–3767.

Gopalan, A., Maillard, O.-A., and Zaki, M. (2016). Low-rank bandits with latent mixtures. *arXiv preprint arXiv:1609.01508*.

Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pages 817–824.

Lattimore, T. and Szepesvári, C. (2020). *Bandit Algorithms*. Cambridge University Press.

Lattimore, T., Szepesvári, C., and Weisz, G. (2020). Learning with good feature representations in bandits and in rl with a generative model. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5662–5670.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670.

Pacchiano, A., Phan, M., Abbasi Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., and Szepesvari, C. (2020). Model selection in contextual stochastic bandit problems. In *Advances in Neural Information Processing Systems*, pages 10328–10337. Curran Associates, Inc.

Vovk, V. (1998). Competitive on-line linear regression. In *Advances in Neural Information Processing Systems*, pages 364–370.