Adaptive Exploration in Linear Contextual Bandit

Botao Hao
Princeton University

Tor Lattimore
Deepmind

Csaba Szepesvári
Deepmind and University of Alberta

Abstract

Contextual bandits serve as a fundamental model for many sequential decision making tasks. The most popular theoretically justified approaches are based on the optimism principle. While these algorithms can be practical, they are known to be suboptimal asymptotically. On the other hand, existing asymptotically optimal algorithms for this problem do not exploit the linear structure in an optimal way and suffer from lower-order terms that dominate the regret in all practically interesting regimes. We start to bridge the gap by designing an algorithm that is asymptotically optimal and has good finite-time empirical performance. At the same time, we make connections to the recent literature on when exploration-free methods are effective. Indeed, if the distribution of contexts is well behaved, then our algorithm acts mostly greedily and enjoys sub-logarithmic regret. Furthermore, our approach is adaptive in the sense that it automatically detects the nice case. Numerical results demonstrate significant regret reductions by our method relative to several baselines.

1 INTRODUCTION

Stochastic contextual linear bandits, the problem we consider, is interesting due to its rich structure and also because of its potential applications, e.g., in online recommendation systems (Agarwal et al., 2009; Li et al., 2010). In this paper we propose a new algorithm for this problem that is asymptotically optimal, computationally efficient and empirically well-behaved in finite-time regimes. As a consequence of asymptotic optimality, the algorithm adapts to easy instances where it achieves sub-logarithmic regret.

Popular approaches for regret minimisation in contextual bandits include ε-greedy (Langford and Zhang, 2007), explicit optimism-based algorithms (Dani et al., 2008; Rusmevichientong and Tsitsiklis, 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011), and implicit ones, such as Thompson sampling (Agrawal and Goyal, 2013). Although these algorithms enjoy near-optimal worst-case guarantees and can be quite practical, they are known to be arbitrarily suboptimal in the asymptotic regime, even in the non-contextual linear bandit (Lattimore and Szepesvári, 2017).

We propose an optimisation-based algorithm that estimates and tracks the optimal allocation for each context/action pair. This technique is most well known for its effectiveness in pure exploration (Chan and Lai, 2006; Garivier and Kaufmann, 2016; Degenne et al., 2019, and others). The approach has been used in regret minimisation in linear bandits with fixed action sets (Lattimore and Szepesvári, 2017) and structured bandits (Combes et al., 2017). The last two articles provide algorithms for the non-contextual case and hence cannot be applied directly to our setting. More importantly, however, the algorithms are not practical. The first algorithm uses a complicated three-phase construction that barely updates its estimates. The second algorithm is not designed to handle large action spaces and has a 'lower-order' term in the regret that depends linearly on the number of actions and dominates the regret in all practical regimes. This lower-order term is not merely a product of the analysis, but also reflected

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

in the experiments (see Section 5.4 for details).

The most closely related work is by Ok et al. (2018) who study a reinforcement learning setting. A stochastic contextual bandit can be viewed as a Markov decision process where the state represents the context and the transition is independent of the action. The structured nature of the mentioned paper means our setting is covered by their algorithm. Again, however, the algorithm is too general to exploit the specific structure of the contextual bandit problem. Their algorithm is asymptotically optimal, but suffers from lower-order terms that are linear in the number of actions and dominate the regret in all practically interesting regimes. In contrast, our algorithm is asymptotically optimal, but also practical in finite-horizon regimes, as will be demonstrated by our experiments.

The contextual linear bandit also serves as an interesting example where the asymptotics of the problem are not indicative of what should be expected in finite-time (see the second scenario in Section 5.2). This is in contrast to many other bandit models where the asymptotic regret is also roughly optimal in finite time (Lattimore and Szepesvári, 2019). There is an important lesson here. Designing algorithms that optimize for the asymptotic regret may make huge sacrifices in finite-time.

Another interesting phenomenon is related to the idea of 'natural exploration' that occurs in contextual bandits (Bastani et al., 2017; Kannan et al., 2018). A number of authors have started to investigate the striking performance of greedy algorithms in contextual bandits. In most bandit settings the greedy policy does not explore sufficiently and suffers linear regret. In some contextual bandit problems, however, the changing features ensure the algorithm cannot help but explore. Our algorithm and analysis highlights this effect (see Section 3.1 for details). If the context distribution is sufficiently rich, then the algorithm is eventually almost completely greedy and enjoys sub-logarithmic regret. As opposed to the cited previous works, our algorithm achieves this under the cited favourable conditions *while* at the same time it satisfies the standard optimality guarantees when the favourable conditions do not hold. As another contribution, we prove that algorithms based on optimism, similarly to the new algorithm, also enjoy sub-logarithmic regret in the rich-context distribution setting (Theorem 3.9), and hence differences appear in lower order terms only between these algorithms.

The rest of the paper is organized as follows. We first introduce the problem setting (Section 2), which we follow by presenting our asymptotic lower bound (Section 3). Section 4 introduces our new algorithm, which is claimed to match the lower bound. A proof sketch of this claim is presented in the same section. Section 5 presents experiments to illuminate the behaviour of the new algorithm in comparison to its strongest competitors. Section 6 discusses remaining notable open questions.

Notation Let $[n] = \{1, 2, \dots, n\}$. For a vector x and positive semidefinite matrix A we let $\|x\|_A = \sqrt{x^\top A x}$. The cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

2 PROBLEM SETTING

We consider the stochastic K-armed contextual linear bandit with a horizon of n rounds and M possible contexts. The assumption that the contexts are discrete cannot be dropped but as we shall at least M will not play an important role in the regret bounds. This assumption would hold for example in a recommender system if users are clustered into finitely many groups. For each context $m \in [M]$ there is a known feature/action set $\mathcal{A}^m \subset \mathbb{R}^d$ with $|\mathcal{A}^m| = K$. The interaction protocol is as follows. First the environment samples a sequence of independent contexts $(c_t)_{t=1}^n$ from an unknown distribution p over $[M]$ and each context is assumed to appear with positive probability. At the start of round t the context c_t is revealed to the learner, who may use their observations to choose an action $X_t \in \mathcal{A}_t = \mathcal{A}^{c_t}$. The reward is

$$Y_t = \langle X_t, \theta \rangle + \eta_t \,,$$

where $(\eta_t)_{t=1}^n$ is a sequence of independent standard Gaussian random variables and $\theta \in \mathbb{R}^d$ is an unknown parameter. The Gaussian assumption can be relaxed to conditional sub-Gaussian assumption for the regret upper bound, but is necessary for the regret lower bound. Throughout, we consider a frequentist setting in the sense that θ is fixed. For simplicity, we assume each $\mathcal{A}^m$ spans $\mathbb{R}^d$ and $\|x\|_2 \leq 1$ for all $x \in \cup_m \mathcal{A}^m$.

The performance metric is the cumulative expected regret, which measures the difference between the expected cumulative reward collected by the omniscient policy that knows θ and the learner's expected cumulative reward. The optimal arm associated with context m is $x_m^* = \mathrm{argmax}_{x\in\mathcal{A}^m}\langle x,\theta\rangle$. Then the expected cumulative regret of a policy π when facing the bandit determined by θ is

$$R_\theta^\pi(n) = \mathbb{E}\left[\sum_{t=1}^n \langle x_{c_t}^*, \theta\rangle - \sum_{t=1}^n Y_t\right].$$

Note that this cumulative regret also depends on the context distribution p and action sets. They are omitted from the notation to reduce clutter and because there will never be ambiguity.

3 ASYMPTOTIC LOWER BOUND

We investigate the fundamental limit of linear contextual bandit by deriving its instance-dependent asymptotic lower bound. First, we define the class of policies that are taken into consideration.

Definition 3.1 (Consistent Policy)**.** A policy π is called consistent if the regret is subpolynomial for any bandit in that class and all context distributions:

$$R_\theta^\pi(n) = o(n^\varepsilon),\ \text{for all } \varepsilon > 0 \text{ and all } \theta \in \mathbb{R}^d. \quad (3.1)$$

The next lemma is the key ingredient in proving the asymptotic lower bound. Given a context m and $x \in \mathcal{A}^m$ let $\Delta_x^m = \langle x_m^* - x, \theta\rangle$ be the suboptimality gap. Furthermore, let $\Delta_{\min} = \min_{m\in[M]}\min_{x\in\mathcal{A}^m,\Delta_x^m>0}\Delta_x^m$.

Lemma 3.2. Assume that $p(m) > 0$ for all $m \in [M]$ and that x_m^* is uniquely defined for each context m and let π be consistent. Then for sufficiently large n the expected covariance matrix

$$\bar{G}_n = \mathbb{E}\left[\sum_{t=1}^n X_t X_t^\top\right], \quad (3.2)$$

is invertible. Furthermore, for any context m and any arm $x \in \mathcal{A}^m$,

$$\limsup_{n\to\infty} \log(n)\|x - x_m^*\|^2_{\bar{G}_n^{-1}} \le \frac{(\Delta_x^m)^2}{2}. \quad (3.3)$$

The proof is deferred to Appendix A.1 in the supplementary material. Intuitively, the lemma shows that any consistent policy must collect sufficient statistical evidence at confidence level $1 - 1/n$ that suboptimal arms really are suboptimal. This corresponds to ensuring that the width of an appropriate confidence interval $\sqrt{2\log(n)}\|x - x_m^*\|_{\bar{G}_n^{-1}}$ is approximately smaller than the sub-optimality gap Δ_x^m.

Theorem 3.3 (Asymptotic Lower Bound)**.** Under the same conditions as Lemma 3.2,

$$\liminf_{n\to\infty} \frac{R_\theta^\pi(n)}{\log(n)} \ge \mathcal{C}(\theta, \mathcal{A}^1, \dots, \mathcal{A}^M), \quad (3.4)$$

where $\mathcal{C}(\theta, \mathcal{A}^1, \dots, \mathcal{A}^M)$ is defined as the optimal value of the following optimisation problem:

$$\inf_{\alpha_{x,m}\in[0,\infty]} \sum_{m=1}^M \sum_{x\in\mathcal{A}^m} \alpha_{x,m}\Delta_x^m \quad (3.5)$$

subject to the constraint that for any context m and suboptimal arm $x \in \mathcal{A}^m$,

$$x^\top \left(\sum_{m=1}^M \sum_{x\in\mathcal{A}^m} \alpha_{x,m} x x^\top\right)^{-1} x \le \frac{(\Delta_x^m)^2}{2}. \quad (3.6)$$

Given the result in Lemma 3.2, the proof of Theorem 3.3 follows exactly the same idea of the proof of Corollary 2 in Lattimore and Szepesvári (2017) and thus is omitted here. Later on we will prove a matching upper bound in Theorem 4.3 and argue that our asymtotical lower bound is sharp.

Remark 3.4. In the above we adopt the convention that $\infty \times 0 = 0$ so that $\alpha_{x,m}\Delta_x^m = 0$ whenever $\Delta_x^m = 0$. The inverse of a matrix with infinite entries is defined by passing to the limit in the obvious way, and is not technically an inverse.

Remark 3.5. Let us denote $\{\alpha_{x,m}^*\}_{x\in\mathcal{A}^m,m\in[M]}$ as an optimal solution to the above optimisation problem. It serves as the *optimal allocation rule* for each arm such that the cumulative regret is minimized subject to the width of the confidence interval of each sub-optimal arm is small. Specifically, $\alpha_{x,m}^*\log(n)$ can be interpreted as the approximate optimal number of times arm x should be played having observed context m.

Remark 3.6. Our lower bound may also be derived from a more general bound of Ok et al. (2018), since a stochastic contextual bandit can be viewed as a kind of Markov decision process. We use an alternative proof technique and the two lower bound statements have different forms. The proof is included for completeness.

Example 3.7. When $M = 1$ and $\mathcal{A}^1 = \{e_1, \dots, e_d\}$ is the standard basis vectors, the problem reduces to classical multi-armed bandit and $\mathcal{C}(\theta, \mathcal{A}^1) = \sum_{x \in \mathcal{A}^1, \Delta_x > 0} 2/\Delta_x$, which matches the well-known asymptotic lower bound by Lai and Robbins (1985).

The constant $\mathcal{C}(\theta, \mathcal{A}^1, \dots, \mathcal{A}^M)$ depends on both the unknown parameter θ and the action sets $\mathcal{A}^1, \dots, \mathcal{A}^M$, but *not* the context distribution p. In this sense there is a certain discontinuity in the hardness measure $\mathcal{C}$ as a function of the context distribution. More precisely, problems where $p(m)$ is arbitrarily close to zero may have different regret asymptotically than the problem obtained by removing context m entirely. Clearly as $p(m)$ tends to zero the mth context is observed with vanishingly small probability in finite time and hence the asymptotically optimal regret may not be representative of the finite-time hardness.

3.1 Sub-logarithmic regret

Our matching upper and lower bounds reveal the interesting phenomenon that if the action sets satisfy certain conditions, then sub-logarithmic regret is possible. Consider the scenario that the set of optimal arms $\{x_1^*, \dots, x_M^*\}$ spans $\mathbb{R}^d$ [1]. Let $\Lambda \in \mathbb{R}$ be a large constant to be defined subsequently and for each context m and arm $x \in \mathcal{A}^m$, let $\alpha_{x,m}$ be 0 if $x \neq x_m^*$, and be Λ if else. Then,

$$\sum_{m=1}^{M} \sum_{x \in \mathcal{A}^m} \alpha_{x,m} x x^\top = \Lambda \sum_{m=1}^{M} x_m^* x_m^{*\top} . \tag{3.7}$$

Since the set of optimal arms spans $\mathbb{R}^d$ it holds that for any context m and arm $x \in \mathcal{A}^m$,

$$x^\top \left(\sum_{m=1}^{M} x_m^* x_m^{*\top} \right)^{-1} x < \infty . \tag{3.8}$$

Combining Eq. (3.7) and Eq. (3.8),

$$x^\top \Big(\sum_{m=1}^{M} \sum_{x \in \mathcal{A}^m} \alpha_{x,m} x x^\top \Big)^{-1} x = \Lambda^{-1} x^\top \Big(\sum_{m=1}^{M} x_m^* x_m^{*\top} \Big)^{-1} x .$$

Hence, the constraint in Eq. (3.6) is satisfied for sufficiently large Λ. Since with this choice of $(\alpha_{x,m})$ we have $\sum_{m=1}^{M} \sum_{x \in \mathcal{A}^m} \alpha_{x,m} \Delta_x^m = 0$, it follows that $\mathcal{C}(\theta, \mathcal{A}^1, \dots, \mathcal{A}^M) = 0$. Therefore our upper bound will show that when the set of optimal actions $\{x_1^*, \dots, x_M^*\}$ spans $\mathbb{R}^d$ our new algorithm satisfies

$$\liminf_{n \to \infty} \frac{R_\theta^\pi(n)}{\log(n)} = 0 .$$

Remark 3.8. The choice of $\alpha_{x,m}$ above shows that when $\{x_1^*, \dots, x_M^*\}$ span $\mathbb{R}^d$, then an asymptotically optimal algorithm only needs to play suboptimal arms sub-logarithmically often, which means the algorithm is eventually very close to the greedy algorithm. Bastani et al. (2017); Kannan et al. (2018) also investigate the striking performance of greedy algorithms in contextual bandits. However, Bastani et al. (2017) assume the covariate diversity on the context distribution while Kannan et al. (2018) assume the context is artificially perturbed with noise – these assumptions make these works brittle. In addition, Bastani et al. (2017) only provide a rate-optimal algorithm while our algorithm is optimal in constants (see Theorem 4.3 for details).

As claimed in the introduction, we also prove that algorithms based on optimism can enjoy bounded regret when the set of optimal actions spans the space of all actions. The proof of the following theorem is given in Appendix B.7.

Theorem 3.9. Consider the policy π that plays optimistically by

$$X_t = \operatorname*{argmax}_{x \in \mathcal{A}^{c_t}} \langle \widehat{\theta}_{t-1}, x \rangle + \|x\|_{G_t^{-1}} \beta_t^{1/2} .$$

Suppose that θ is such that $\{x_1^*, \dots, x_M^*\}$ spans $\mathbb{R}^d$. Then, for suitable $(\beta_t)_{t=1}^n$ with $\beta_t = O(d \log(t))$, it holds that $\limsup_{n \to \infty} R_\theta^\pi(n) < \infty$.

Note, the choice of (β_t) for which the above theorem holds also guarantees the standard $\widetilde{O}(d\sqrt{n})$ minimax bound for this algorithm, showing that LinUCB can adapt online to this nice case.

4 OPTIMAL ALLOCATION MATCHING

The instance-dependent asymptotic lower bound provides an optimal allocation rule. However, the optimal allocation $\{\alpha_{x,m}^*\}_{x,m}$ depends on the unknown sub-optimality gap. In this section, we present a

[1] This condition is both sufficient and necessary. More precisely, sub-logarithmic regret is possible if and only if the optimal arms span the space that is spanned by all the available actions.

novel matching algorithm that simultaneously estimates the unknown parameter θ using least squares and updates the allocation rule.

4.1 Algorithm

Let $N_x(t) = \sum_{s=1}^t \mathbb{I}(X_s = x)$ be the number of pulls of arm x after round t and $G_t = \sum_{s=1}^t X_s X_s^\top$. The least squares estimator is $\widehat{\theta}_t = G_t^{-1} \sum_{x=1}^t X_s Y_s$. For each context m the estimated sub-optimality gap of arm $x \in \mathcal{A}^m$ is $\widehat{\Delta}_x^m(t) = \max_{y \in \mathcal{A}^m} \langle y - x, \widehat{\theta}_t \rangle$ and the estimated optimal arm is $\widehat{x}_m^*(t) = \operatorname{argmax}_{x \in \mathcal{A}^m} \langle x, \widehat{\theta}_t \rangle$. The minimum nonzero estimated gap is

$$\widehat{\Delta}_{\min}(t) = \min_{m \in [M]} \min_{x \in \mathcal{A}^m, \widehat{\Delta}_x^m(t) > 0} \widehat{\Delta}_x^m(t) \,.$$

Next, we define a similar optimisation problem as in (3.5) but with a different normalisation.

Definition 4.1. Let $f_{n,\delta}$ be the constant given by

$$f_{n,\delta} = 2(1 + 1/\log(n)) \log(1/\delta) + cd \log(d \log(n)) \,, \tag{4.1}$$

where c is an absolute constant. We write $f_n = f_{n,1/n}$. For any $\widetilde{\Delta} \in [0, \infty)^{|\cup_m \mathcal{A}^m|}$ define $T(\widetilde{\Delta})$ as a solution of the following optimisation problem:

$$\min_{(T_x^m)_{x,m} \in [0,\infty]} \sum_{m=1}^M \sum_{x \in \mathcal{A}^m} T_x^m \widetilde{\Delta}_x^m, \tag{4.2}$$

subject to

$$\|x\|^2_{H_T^{-1}} \leq \frac{\Delta_x^2}{f_n}, \forall x \in \mathcal{A}^m, m \in [M].$$

and that $H_T = \sum_{m=1}^M \sum_{x \in \mathcal{A}^m} T_x^m x x^\top$ is invertible.

If $\widetilde{\Delta}$ is an estimate of Δ, we call the solution $T(\widetilde{\Delta})$ an *approximated allocation rule* in contrast to the *optimal allocation rule* defined in Remark 3.5. Our algorithm alternates between exploration and exploitation, depending on whether or not all the arms have satisfied the approximated allocation rule. We are now ready to describe the algorithm, which starts with a brief initialisation phase.

Initialisation In the first d rounds the algorithm chooses any action X_t in the action set such that X_t is not in the span of $\{X_1, \ldots, X_{t-1}\}$. This is always possible by the assumption that $\mathcal{A}^m$ spans $\mathbb{R}^d$ for all contexts m. At the end of the initialisation phase G_t is guaranteed to be invertible.

Main phase In each round after the initialisation phase the algorithm checks if the following criterion holds for any $x \in \mathcal{A}^{c_t}$:

$$\|x\|^2_{G_{t-1}^{-1}} \leq \max\left\{ \frac{(\widehat{\Delta}_{\min}(t-1))^2}{f_n}, \frac{(\widehat{\Delta}_x^{c_t}(t-1))^2}{f_n} \right\}. \tag{4.3}$$

The algorithm exploits if Eq. (4.3) holds and explores otherwise, as explained below.

Exploitation. The algorithm exploits by taking the greedy action:

$$X_t = \operatorname*{argmax}_{x \in \mathcal{A}^{c_t}} x^\top \widehat{\theta}_{t-1}. \tag{4.4}$$

Exploration. The algorithm explores when Eq. (4.3) does not hold. This means that some actions have not been explored sufficiently. There are two cases to consider. First, when there exists an arm $x' \in \mathcal{A}^{c_t}$ such that

$$N_{x'}(t-1) < \min(T_{x'}^{c_t}(\widehat{\Delta}(t-1)), f_n / \widehat{\Delta}^2_{\min}(t-1)),$$

the algorithm then computes two actions

$$\begin{aligned} b_1 &= \operatorname*{argmin}_{x \in \mathcal{A}^{c_t}} \frac{N_x(t-1)}{\min(T_x^{c_t}(\widehat{\Delta}(t-1)), f_n / \widehat{\Delta}^2_{\min}(t-1))} \\ b_2 &= \operatorname*{argmin}_{x \in \mathcal{A}^{c_t}} N_x(t-1). \end{aligned} \tag{4.5}$$

Let $s(t)$ be the number of exploration rounds defined in Algorithm 1. If $N_{b_2}(t-1) \leq \varepsilon_t s(t)$ the algorithm plays arm $X_t = b_2$ – a form of forced exploration. Otherwise the algorithm plays arm $X_t = b_1$. Finally, rounds where an $x' \in \mathcal{A}^{c_t}$ with the required property does not exist are called *wasted*. In these rounds the algorithm acts optimistically as LinUCB (Abbasi-Yadkori et al., 2011):

$$X_t = \operatorname*{argmax}_{x \in \mathcal{A}^{c_t}} x^\top \widehat{\theta}_{t-1} + \sqrt{f_{n,1/(s(t))^2}} \|x\|_{G_{t-1}^{-1}}, \tag{4.6}$$

where $f_{n,1/(s(t))^2}$ is defined in Eq. (4.1). The complete algorithm is presented in Algorithm 1.

Remark 4.2. The naive forced exploration can be improved by calculating a barycentric spanner (Awerbuch and Kleinberg, 2008) for each action set and then playing the least played action in the spanner. In normal practical setups this makes very little difference, where the forced exploration plays a limited role. For finite-time worst-case analysis, however, it may be crucial, since otherwise the regret may depend linearly on the number of actions, while using the spanner guarantees the forced exploration is sample efficient.

4.2 Asymptotic Upper Bound

Our main theorem states that Algorithm 1 is asymptotically optimal under mild assumptions.

Theorem 4.3. *Suppose that $T_x^m(\Delta)$ is uniquely defined and $T_x^m(\cdot)$ is continuous at Δ for all contexts m and actions $x \in \mathcal{A}^m$. Then the policy π_{oam} proposed in Algorithm 1 with $\varepsilon_t = 1/\log(\log(t))$ satisfies*

$$\limsup_{n\to\infty} \frac{R_\theta^{\pi_{\text{oam}}}(n)}{\log(n)} \le \mathcal{C}(\theta, \mathcal{A}^1, \dots, \mathcal{A}^M). \qquad (4.7)$$

Together with the asymptotic lower bound in Theorem 3.3, we can argue that optimal-allocation matching algorithm is asymptotical optimal and the lower bound in Eq. (3.4) is sharp.

Remark 4.4. The assumption that $T_x^m(\cdot)$ is continuous at Δ is used to ensure the stability of our algorithm. We prove that the uniqueness assumption actually implies continuity (Lemma C.5 in the supplementary material) and thus the continuity assumption could be omitted. There are, however, certain corner cases where uniqueness does not hold. For example when $\theta = (1,0)^\top, \mathcal{A} = \{(1,0),(0,1),(0,-1)\}$.

4.3 Proof Sketch

The complete proof is deferred to Appendix A.2 in the supplementary material. At a high level the analysis of the optimisation-based approach consists of three parts. (1) Showing that the algorithm's estimate of the true parameter is close to the truth in finite time. (2) Showing that the algorithm subsequently samples arms approximately according to the unknown optimal allocation and (3) Showing that the greedy action when arms have been sampled sufficiently according to the optimal allocation is optimal with high probability. Existing optimisation-based algorithms suffer from dominant 'lower-order' terms because they use simple empirical means for Part (1), while here we use the data-efficient least-squares estimator.

We let Explore = F-Explore ∪ UW-Explore ∪ W-Explore be the set of exploration rounds, decomposed into disjoint sets of forced exploration ($X_t = b_1$), unwasted exploration ($X_t = b_2$) and wasted exploration (LinUCB), and let Exploit be the set of exploitation rounds.

Algorithm 1 Optimal Allocation Matching

```
Input: exploration parameter $\varepsilon_t$, exploration
    counter $s(d) = 0$.
for $t = 1$ to $d$ do
    Observe an action set $\mathcal{A}^{c_t}$, pull arm $X_t$ such that
      $X_t$ is not in the span of $\{X_1, \dots, X_{t-1}\}$.
end
for $t = d+1$ to $n$ do
    Observe an action set $\mathcal{A}^{c_t}$ and solve the optimi-
      sation problem (4.2) based on the estimated gap
      $\widehat{\Delta}(t-1)$.
    if $\|x\|^2_{G_{t-1}^{-1}} \le \max\{\frac{\widehat{\Delta}^2_{\min}(t-1)}{f_n}, \frac{(\widehat{\Delta}_x^{c_t}(t-1))^2}{f_n}\}, \forall x \in$
      $\mathcal{A}^{c_t}$, then
        Pull arm $X_t = \operatorname{argmax}_{x\in\mathcal{A}^{c_t}} x^\top \widehat{\theta}_{t-1}$.
    else
        $s(t) = s(t-1) + 1$
        if $N_x(t-1) \ge \min(T_x(\widehat{\Delta}(t-1)), f_n/(\widehat{\Delta}_{\min}(t-$
          $1)))^2, \forall x \in \mathcal{A}^{c_t}$, then
            Pull arm according to LinUCB in (4.6).
        else
            Calculate $b_1, b_2$ as in Eq. (4.5).
            if $N_{b_2}(t-1) \le \varepsilon_t s(t-1)$ then
                Pull arm $X_t = b_2$.
            else
                Pull arm $X_t = b_1$.
            end
        end
    end
    Update $\widehat{\theta}_t, \widehat{\Delta}_x^{c_t}(t), \widehat{\Delta}_{\min}(t)$.
end
```

Regret while exploiting The criterion in Eq. (4.3) guarantees that the greedy action is optimal with high probability in exploitation rounds. To see this, note that if t is an exploitation round, then the sub-optimality gap of greedy action X_t satisfies the following with high probability:

$$\Delta_{X_t}^{c_t} \lesssim \sqrt{\frac{\log(\log(n))}{1 \vee N_{X_t}(t-1)}} < \Delta_{\min}\,.$$

Since the instantaneous regret either vanishes or is larger than $\Delta_{\min}$, we have

$$\mathbb{E}\Big[\sum_{t\in\text{Exploit}} \Delta_t^{c_t}\Big] = o(\log(n)).$$

Regret while exploring Based on the design of our algorithm, the regret while exploring is decomposed into three terms,

$$\mathbb{E}\Big[\sum_{t\in\text{Explore}} \Delta_{X_t}^{c_t}\Big] = \mathbb{E}\Big[\sum_{t\in\text{F-Explore}} \Delta_{X_t}^{c_t}\Big] + \mathbb{E}\Big[\sum_{t\in\text{W-Explore}} \Delta_{X_t}^{c_t}\Big] + \mathbb{E}\Big[\sum_{t\in\text{UW-Explore}} \Delta_{X_t}^{c_t}\Big]\,.$$

Shortly we argue that the regret incurred in W-Explore$\cup$UW-Explore is at most logarithmic and hence the regret in rounds associated with forced exploration is sub-logarithmic:

$$\mathbb{E}\Big[\sum_{t\in\text{F-Explore}} \Delta_{X_t}^{c_t}\Big] = O(\varepsilon_n|\text{Explore}|) = o(\log(n))\,.$$

The regret in W-Explore is also sub-logarithmic. To see this, we first argue that $|\text{W-Explore}| = O(|\text{UW-Explore}|)$ since each context has positive probability. Combining with the fact that $|\text{UW-Explore}|$ is logarithmic in n and the regret of LinUCB is square root in time horizon,

$$\mathbb{E}\Big[\sum_{t\in\text{W-Explore}} \Delta_t^{c_t}\Big] = o(\log(n))\,.$$

The regret in UW-Explore is logarithmic in n with the asymptotically optimal constant using the definition of the optimal allocation:

$$\limsup_{n\to\infty} \frac{\mathbb{E}\left[\sum_{t\in\text{UW-Explore}} \Delta_t^{c_t}\right]}{\log(n)} = \mathcal{C}(\theta, \mathcal{A}^1, \ldots, \mathcal{A}^M)\,.$$

Of course many details have been hidden here, which are covered in detail in the supplementary material.

5 EXPERIMENTS

In this section, we first compare our proposed algorithm and LinUCB (Abbasi-Yadkori et al., 2011) on some specific problem instances to showcase their strengths and weaknesses. We examine OSSB (Combes et al., 2017) on instances with large action sets to illustrate its weakness due to not using the linear structure everywhere. Since Combes et al. (2017) demonstrated that OSSB dominates the algorithm of Lattimore and Szepesvári (2017), we omit this algorithm from our experiments. In the end, we include the comparison with LinTS (Agrawal and Goyal, 2013). Some additional experiments are deferred to Appendix D in the supplementary material.

To save computation, we follow the lazy-update approach, similar to that proposed in Section 5.1 of (Abbasi-Yadkori et al., 2011): The idea is to resolve the optimisation problem (4.2) whenever $\det(G_t)$ increases by a constant factor $(1+\zeta)$ and in all scenarios we choose (the arbitrary value) $\zeta = 0.1$. All codes were written in Python. To solve the convex optimisation problem (4.2), we use the CVXPY library (Diamond and Boyd, 2016).

5.1 Fixed Action Set

Finite-armed linear bandits with fixed action set are a special case of linear contextual bandits. Let $d = 2$ and let the true parameter be $\theta = (1, 0)^\top$. The action set $\mathcal{A} = \{x_1, x_2, x_3\}$ is fixed and $x_1 = (1, 0)^\top$, $x_2 = (0, 1)^\top$, $x_3 = (1 - u, 5u)^\top$. We consider $u = \{0.1, 0.2\}$. By construction, x_1 is the optimal arm. From Figure 1, we observe that LinUCB suffers significantly more regret than our algorithm. The reason is that if u is very small, then x_1 and x_3 point in almost the same direction and so choosing only these arms does not provide sufficient information to quickly learn which of x_1 or x_3 is optimal. On the other hand, x_2 and x_1 point in very different directions and so choosing x_2 allows a learning agent to quickly identify that x_1 is in fact optimal. LinUCB stops pulling x_2 once it is optimistic and thus fails to find the right balance between information and reward. Our algorithm, however, takes this into consideration by tracking the optimal allocation ratios.

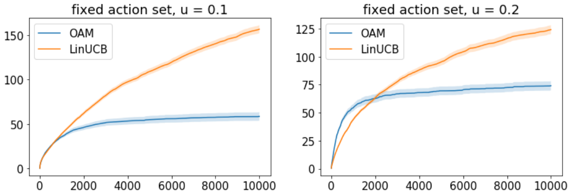

Figure 1: Fixed action set. The results are averaged over 100 realisations. Here and also later, the shaded areas show the standard errors.

5.2 Changing Action Set

We consider a simple but representative case when there are only two action sets $\mathcal{A}^1$ and $\mathcal{A}^2$ available.

Scenario One. In each round, $\mathcal{A}^1$ is drawn with probability 0.3 while $\mathcal{A}^2$ is drawn with probability 0.7. Set $\mathcal{A}^1$ contains $x_1^1 = (1, 0, 0)^\top$, $x_2^1 = (0, 1, 0)^\top$, and $x_3^1 = (0.9, 0.5, 0)^\top$, while set $\mathcal{A}^2$ contains $x_1^2 = (0, 1, 0)^\top$, $x_2^2 = (0, 0, 1)^\top$, and $x_3^2 = (0, 0.5, 0.9)^\top$. The true parameter θ is $(1, 0, 1)^\top$. From the left

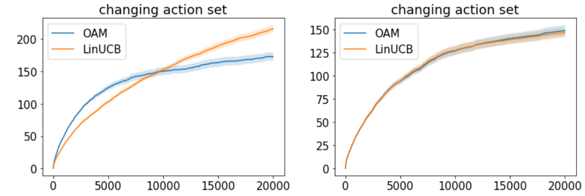

Figure 2: Changing action sets. The left panel is for scenario one and the right panel is for scenario two. The results are averaged over 100 realisations.

panel of Figure 2, we observe that LinUCB, while starts better, eventually again suffers more regret than our algorithm.

Scenario Two. In each round, $\mathcal{A}^1$ is drawn with probability 0.99, while $\mathcal{A}^2$ is drawn with probability 0.01. Set $\mathcal{A}^1$ contains three actions: $x_1^1 = (1,0)^\top$, $x_2^1 = (0,1)^\top$, $x_3^1 = (0.9, 0.5)^\top$, while set $\mathcal{A}^2$ contains three actions: $x_1^2 = (0,1)^\top$, $x_2^2 = (-1,0)^\top$, $x_3^2 = (-1,0)$. Apparently, x_1^1 and x_1^2 are the optimal arms for each action set and they span $\mathbb{R}^2$. Based on the allocation rule in Section 3.1, the algorithm is advised to pull actions x_1^1 and x_1^2 very often based on asymptotics. However, since the probability that $\mathcal{A}^2$ is drawn is extremely small, we are very likely to fall back to wasted exploration and use LinUCB to explore. Thus, in the short term, our algorithm will suffer from the drawback that optimistic algorithms also suffer from and what is described in Section 5.1. Although, the asymptotics will eventually "kick in", it may take extremely long time to see the benefits of this and the algorithm's finite-time performance will be poor. Indeed, this is seen on the right panel of Figure 2, which shows that in this case LinUCB and our algorithm are nearly indistinguishable.

5.3 Sublinear/Bounded Regret

Earlier we have argued that when the optimal arms of all action sets span $\mathbb{R}^d$, our algorithm achieves sub-logarithmic regret. Here, we experimentally study this interesting case. We consider $M = 2$. In each round, $\mathcal{A}^1$ is drawn with probability 0.8 while $\mathcal{A}^2$ is drawn with probability 0.2 and the true parameter θ is $(1,0)^\top$.Set $\mathcal{A}^1$ contains three actions: $x_1^1 = (1,0)^\top$, $x_2^1 = (0,1)^\top$, $x_3^1 = (0.9, 0.5)^\top$, while set $\mathcal{A}^2$ contains three actions: $x_1^2 = (0,1)^\top$, $x_2^2 = (-1,0)^\top$, $x_3^2 = (-1,0)$. As discussed before, x_1^1 and x_1^2 are the optimal arms for each action set and they span $\mathbb{R}^2$. The results are shown in the left subpanel of Figure 3. The regret of our algorithm appears to have stopped growing after a short period of increase. In line with Theorem 3.9, LinUCB is seen to achieve bounded regret in this problem.

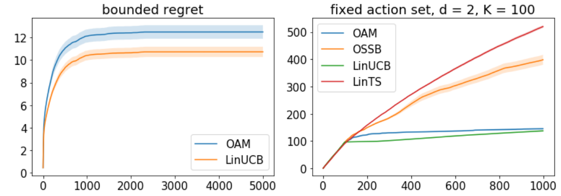

Figure 3: The left panel is for bounded regret and right panel is for large action space. The results are averaged over 100 realisations.

5.4 Large, Fixed Action Set

We let $d = 2$ and $\theta = (1,0)^\top$. We generate 100 uniformly distributed on the d-dimensional unit sphere (fixed action set). The results are shown in the right subfigure of Figure 3. When the action space is large, OSSB suffers significantly large regret and becomes unstable due to not using the linear structure everywhere. The regret of (the theoretically justified version of) LinTS is also very large due to the unnecessary variance factor required by its theory.

6 DISCUSSION

We presented a new optimisation-based algorithm for linear contextual bandits that is asymptotically optimal and adapts to both the action sets and unknown parameter. The new algorithm enjoys sub-logarithmic regret when the collection of optimal actions spans $\mathbb{R}^d$, a property that we also prove for optimism-based approaches. There are many open questions. A natural starting point is to prove near-minimax optimality of the new algorithm, possibly with minor modifications. Our work also highlights the dangers of focusing too intensely on asymptotics, which for contextual bandits hide completely the dependence on the context distribution. This motivates the intriguing challenge to understand the finite-time instance-dependent regret. Another open direction is to consider the asymptotics when the context space is continuous, which has not seen any attention.

Acknowledgements

Csaba Szepesvári gratefully acknowledges funding from the Canada CIFAR AI Chairs Program, Amii and NSERC.

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, D., Chen, B.-C., Elango, P., Motgi, N., Park, S.-T., Ramakrishnan, R., Roy, S., and Zachariah, J. (2009). Online models for content optimization. In *Advances in Neural Information Processing Systems*, pages 17–24.

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

Awerbuch, B. and Kleinberg, R. (2008). Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114.

Bastani, H., Bayati, M., and Khosravi, K. (2017). Mostly exploration-free algorithms for contextual bandits. *arXiv preprint arXiv:1704.09011*.

Chan, H. P. and Lai, T. L. (2006). Sequential generalized likelihood ratios and adaptive treatment allocation for optimal sequential selection. *Sequential Analysis*, 25:179–201.

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.

Combes, R., Magureanu, S., and Proutiere, A. (2017). Minimal exploration in structured stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 1763–1771.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In Servedio, R. A. and Zhang, T., editors, *21st Annual Conference on Learning Theory - COLT*, pages 355–366.

Degenne, R., Koolen, W. M., and Ménard, P. (2019). Non-asymptotic pure exploration by solving games. *arXiv preprint arXiv:1906.10431*.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Garivier, A. and Kaufmann, E. (2016). Optimal best arm identification with fixed confidence. In Feldman, V., Rakhlin, A., and Shamir, O., editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA. PMLR.

Kannan, S., Morgenstern, J. H., Roth, A., Waggoner, B., and Wu, Z. S. (2018). A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. In *Advances in Neural Information Processing Systems*, pages 2227–2236.

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

Langford, J. and Zhang, T. (2007). The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 817–824.

Lattimore, T. and Szepesvári, C. (2017). The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Artificial Intelligence and Statistics*, pages 728–737.

Lattimore, T. and Szepesvári, C. (2019). Bandit algorithms. *preprint*.

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*, WWW '10, pages 661–670.

Ok, J., Proutiere, A., and Tranos, D. (2018). Exploration in structured reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8874–8882.

Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

Tsybakov, A. B. (2008). *Introduction to Nonparametric Estimation.* Springer, 1st edition.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.