A fully adaptive algorithm for pure exploration in linear bandits

Liyuan Xu[†‡] **Junya Honda**[†‡] **Masashi Sugiyama**[‡†]
†:The University of Tokyo ‡:RIKEN

Abstract

We propose the first fully-adaptive algorithm for pure exploration in linear bandits—the task to find the arm with the largest expected reward, which depends on an unknown parameter linearly. While existing methods partially or entirely fix sequences of arm selections before observing rewards, our method adaptively changes the arm selection strategy based on past observations at each round. We show our sample complexity matches the achievable lower bound up to a constant factor in an extreme case. Furthermore, we evaluate the performance of the methods by simulations based on both synthetic setting and real-world data, in which our method shows vast improvement over existing ones.

1 Introduction

The *multi-armed bandit* (MAB) problem (Robbins, 1985) is a sequential decision-making problem, where the agent sequentially chooses one arm out of K arms and receives a stochastic reward drawn from a fixed, unknown distribution related with the arm chosen. While most of the literature on the MAB focused on the maximization of the cumulative rewards, we consider a pure-exploration setting called the best arm identification problem (Bubeck et al., 2009). Here, the goal of the agent is to identify the arm with the maximum expected reward.

The best arm identification problem has recently gained increasing attention, and a considerable amount of work covers many variants of it. For example, Audibert and Bubeck (2010) considered the fixed budget setting, where the agent tries to minimize the misspecification probability in a fixed number of trials, and Even-Dar et al. (2006) introduced the fixed confidence setting, where the agent tries to minimize the number of trials until the probability of misspecification becomes smaller than a fixed threshold.

An important extension of the MAB is the *linear bandit* (LB) problem (Abe and Long, 1999; Auer, 2002). In the LB problem, each arm has its own feature $x \in \mathbb{R}^d$, and the expected reward can be written as $x^\top \theta$, where $\theta \in \mathbb{R}^d$ is an unknown parameter and $x^\top$ is the transpose of x. Although there are a number of studies in the LB (Abbasi-Yadkori et al., 2011; Li et al., 2010), most of them aim for maximization of the cumulative rewards, and only a few consider the pure-exploration setting.

In spite of the scarce literature, the best arm identification problem in the LB has a wide range of applications. For example, Hoffman et al. (2014) applied the pure exploration in the LB to the optimization of a traffic sensor network and automatic hyper-parameter tuning in machine learning.

The first work that addressed the best arm identification problem for the LB was by Hoffman et al. (2014). They studied the best arm identification in the fixed-budget setting with correlated reward distributions and devised an algorithm called BayesGap, which is a Bayesian version of the gap based exploration algorithm (Gabillon et al., 2012).

Although BayesGap outperformed algorithms that ignore the linear correlation of rewards, there is a drawback that it never pulls arms turned out to be suboptimal. As studied in Soare et al. (2014) and Lattimore and Szepesvari (2017), ignoring sub-optimal arms can significantly harm the performance in the LB. For example, consider the case where there are three arms and their features are $x_1 = (1, 0)^\top$, $x_2 = (1, 0.01)^\top$, and $x_3 = (0, 1)^\top$, respectively. Now, if $\theta = (\theta_1, \theta_2)^\top = (2, 0.01)^\top$, then the expected reward of arms 1 and 2 are close to each other, hence it is hard to figure out the best arm just by observing the samples from them. On the other hand, pulling arm 3 greatly reduces the number of samples required, since

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

it enhances the accuracy of estimation of θ_2. As illustrated in this example, pulling a sub-optimal arm can give valuable insight for comparing near-optimal arms in the LB.

Soare et al. (2014) constructed the first algorithm that pulls sub-optimal arms for exploration in the best arm identification. They studied the fixed-confidence setting and derived an algorithm based on transductive experimental design (Yu et al., 2006), called $\mathcal{XY}$-static allocation. The algorithm employs a static arm selection strategy, in the sense that it fixes all arm selections before observing any reward. Therefore, it is not able to focus on estimating near-optimal arms, thus the algorithm can only be the worst-case optimal.

In order to develop more efficient algorithms, it is necessary to pull arms adaptively based on past observations so that most samples are allocated for comparison of near-optimal arms. The difficulty in constructing an adaptive strategy is that a confidence bound for statically selected arms is not always applicable when arms are adaptively selected. In particular, a confidence bound for an adaptive strategy introduced by Abbasi-Yadkori et al. (2011) is looser than a bound for a static strategy derived from Azuma's inequality (Azuma, 1967) by a factor of $\sqrt{d}$ in some cases, where d is the dimension of the feature. Soare et al. (2014) tried to mitigate this problem by introducing a semi-adaptive algorithm called $\mathcal{XY}$-adaptive allocation, which divides rounds into multiple phases and uses different static allocation strategies in different phases. Although this theoretically improves the sample complexity, the algorithm has to discard all samples collected in the previous phases to make the confidence bound for static strategies applicable, which drops the empirical performance significantly.

To discuss tightness of the sample complexity of $\mathcal{XY}$-adaptive allocation, Soare et al. (2014) introduced the $\mathcal{XY}$-oracle allocation algorithm, which assumes access to the true parameter θ for selecting arms to pull. They discussed that the sample complexity of this algorithm can be used as a lower bound on the sample complexity for this problem and claimed that the upper bound on the sample complexity of $\mathcal{XY}$-adaptive allocation is close to this lower bound. However, the derived upper bound is not given in an explicit form and contains a complicated term coming from $\mathcal{XY}$-static allocation used as a subroutine. In fact, the sample complexity of $\mathcal{XY}$-adaptive allocation is much worse than that of $\mathcal{XY}$-oracle allocation, as we will see numerically in Section 7.1.

Our contribution is to develop a novel fully-adaptive algorithm, which changes arm selection strategies based on all of the past observations at every round. Although this prohibits us from using a tighter bound for static strategies, we show that the factor $\sqrt{d}$ can be avoided by the careful construction of the confidence bound, and the sample complexity almost matches that of $\mathcal{XY}$-oracle allocation. We conduct experiments to evaluate the performance of the proposed algorithm, showing that it requires ten times less samples than existing methods to achieve the same level of accuracy.

2 Problem formulation

We consider the LB problem, where there are K arms with features $x_1, \dots, x_K \in \mathbb{R}^d$. We denote the set of the features as $\mathcal{X} = \{x_1, \dots, x_K\}$ and the largest l_2-norm of the features as $L = \max_{i \in \{1,\dots,K\}} \|x_i\|_2$. At every round t, the agent pulls an arm $a_t \in [K] = \{1, \dots, K\}$, and observes immediate reward r_t, which is characterized by

$$r_t = x_{a_t}^\top \theta + \varepsilon_t.$$

Here, $\theta \in \mathbb{R}^d$ is an unknown parameter, and ε_t represents a noise variable, whose expectation equals zero. We assume that the l_2-norm of θ is less than S and the noise distribution is conditionally R-sub-Gaussian, which means that noise variable ε_t satisfies

$$\mathbb{E}\left[\mathrm{e}^{\lambda \varepsilon_t} | x_{a_1}, \dots, x_{a_{t-1}}, \varepsilon_1, \dots, \varepsilon_{t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right)$$

for all $\lambda \in \mathbb{R}$. This condition requires the noise distribution to have zero expectation and R^2 or less variance (Abbasi-Yadkori et al., 2011). As prior work (Abbasi-Yadkori et al., 2011; Soare et al., 2014), we assume that parameters R and S are known to the agent.

We focus on the (ε, δ)-best arm identification problem. Let $a^* = \arg\max_i x_i^\top \theta$ be the best arm, and x^* be the feature of arm a^*. The problem is to design an algorithm to find arm $\hat{a}^*$ which satisfies

$$\mathbb{P}[(x^* - x_{\hat{a}^*})^\top \theta \geq \varepsilon] \leq \delta, \tag{1}$$

as fast as possible.

3 Confidence Bounds

In order to solve the best arm identification in the LB setting, the agent sequentially estimates θ from past observations and bounds the estimation error. However, if arms are pulled adaptively based on past observations, the estimation becomes much more complicated compared to the case where pulled arms are fixed in advance. In this section, we discuss this difference and how we can construct a tight bound for an algorithm with an adaptive selection strategy.

Given the sequence of arm selections $\mathbf{x}_n = (x_{a_1}, \dots, x_{a_n})$, one of the most standard estimators for θ is the least-squares estimator given by

$$\hat{\theta}_n = A_{\mathbf{x}_n}^{-1} b_{\mathbf{x}_n},$$

where $A_{\mathbf{x}_n}$ and $b_{\mathbf{x}_n}$ are defined as

$$A_{\mathbf{x}_n} = \sum_{t=1}^{n} x_{a_t} x_{a_t}^\top, \quad b_{\mathbf{x}_n} = \sum_{t=1}^{n} x_{a_t} r_t.$$

Soare et al. (2014) used the ordinary least-squares estimator $\hat{\theta}_n$ combined with the following proposition on the confidence ellipsoid for $\hat{\theta}_n$, which is derived from Azuma's inequality (Azuma, 1967).

Proposition 1 (Soare et al., 2014, Prop. 1)**.** *Let noise variable ε_t be bounded as $\varepsilon \in [-\sigma, \sigma]$ for $\sigma > 0$, then, for any fixed sequence $\mathbf{x}_n$, statement*

$$|x^\top \theta - x^\top \hat{\theta}_n| \le 2\sigma \|x\|_{A_{\mathbf{x}_n}^{-1}} \sqrt{2 \log\left(\frac{6n^2 K}{\delta \pi^2}\right)} \qquad (2)$$

holds for all $n \in \{1, 2, \dots\}$ and $x \in \mathcal{X}$ with probability at least $1 - \delta$ for $\|x\|_A = \sqrt{x^\top A x}$.

The assumption that $\mathbf{x}_n$ is fixed is essential in Prop. 1. In fact, if $\mathbf{x}_n$ is adaptively determined depending on past observations, then the estimator $\hat{\theta}_n$ is no more unbiased and it becomes essential to consider the regularized least-squares estimator $\hat{\theta}_n^\lambda$ given by

$$\hat{\theta}_n^\lambda = (A_{\mathbf{x}_n}^\lambda)^{-1} b_{\mathbf{x}_n},$$

where $A_{\mathbf{x}_n}^\lambda$ is defined by

$$A_{\mathbf{x}_n}^\lambda = \lambda I + \sum_{t=1}^{n} x_{a_t} x_{a_t}^\top,$$

for $\lambda > 0$ and the identity matrix I. For this estimator, we can use another confidence bound which is valid even if an adaptive strategy is used.

Proposition 2 (Abbasi-Yadkori et al., 2011, Thm. 2)**.** *In the LB with conditionally R-sub-Gaussian noise, if the l_2-norm of parameter θ is less than S, then statement*

$$|x^\top (\hat{\theta}_n^\lambda - \theta)| \le \|x\|_{(A_{\mathbf{x}_n}^\lambda)^{-1}} C_n$$

holds for given $x \in \mathbb{R}^d$ and all $n \in \{1, 2, \dots\}$ with probability at least $1 - \delta$, where C_n is defined as

$$C_n = R\sqrt{2 \log \frac{\det(A_{\mathbf{x}_n}^\lambda)^{\frac{1}{2}}}{\lambda^{\frac{d}{2}} \delta}} + \lambda^{\frac{1}{2}} S. \qquad (3)$$

Moreover, if $\|x_{a_t}\| \le L$ holds for all $t > 0$, then

$$C_n \le R\sqrt{d \log \frac{1 + nL^2/\lambda}{\delta}} + \lambda^{\frac{1}{2}} S. \qquad (4)$$

Although the bound in (4) holds regardless of whether the arm selection strategy is static or adaptive, the bound is looser than Prop. 1 by an extra factor $\sqrt{d}$ when a static strategy is considered.

In the following sections, we use the bound in (3) to construct an algorithm that adaptively selects arms based on past data. We reveal that the extra factor $\sqrt{d}$ arises from looseness of (4) and the sample complexity can be bounded without this factor by an appropriate evaluation of (3).

4 Arm Selection Strategies

In order to minimize the number of samples, the agent has to select arms that reduce the interval of the confidence bound as fast as possible. In this section, we discuss such an arm selection strategy, and in particular, we consider the strategy to reduce the matrix norm $\|x_i - x_j\|_{A_{\mathbf{x}_n}^{-1}}$, which represents the uncertainty in the estimation of the gap of expected rewards between arms i and j.

Soare et al. (2014) introduced the strategy called $\mathcal{XY}$-static allocation, which makes the sequence of selection $\mathbf{x}_n$ to be

$$\arg\min_{\mathbf{x}_n} \max_{x, x' \in \mathcal{X}} \|x - x'\|_{A_{\mathbf{x}_n}^{-1}}. \qquad (5)$$

The problem is to minimize the confidence bound of the direction hardest to estimate, which is known as transductive experimental design (Yu et al., 2006). Note that this problem does not depend on the past reward, which satisfies the prerequisite of Prop. 1.

A drawback of this strategy is that it minimizes the largest matrix norm $\max_{x,x' \in \mathcal{X}} \|x - x'\|_{A_{\mathbf{x}_n}}$ for all feature pairs $x, x' \in \mathcal{X}$. However, considering that our goal is to find the best arm a^*, we are not interested in estimating the gaps between all arms but the gaps between the best arm and the rest. Therefore, we should spare more samples for estimating gaps of arms with relatively high rewards. This cannot be achieved in the static strategy, since we need to change arm selections based on past rewards.

In order to overcome this weakness while using Prop. 1, Soare et al. (2014) proposed a semi-adaptive strategy called the $\mathcal{XY}$-adaptive strategy. This strategy partitions rounds into multiple phases and arms to select are static within a phase but changes between phases. At the beginning of phase j, it constructs a set of potentially optimal arms $\hat{\mathcal{X}}_j$ based on the samples collected during the previous phase $j-1$. Then, it selects the sequence $\mathbf{x}_n$ in phase j as

$$\arg\min_{\mathbf{x}_n} \max_{x, x' \in \hat{\mathcal{X}}_j} \|x - x'\|_{A_{\mathbf{x}_n}^{-1}}, . \qquad (6)$$

As it goes through the phases, the size of $\hat{\mathcal{X}}_j$ decreases so that the algorithm can focus on discriminating a small number of arms.

Although the $\mathcal{XY}$-adaptive strategy can avoid the extra factor $\sqrt{d}$ in (4), the agent has to reset the design matrix $A_{\mathbf{x}_n}$ at the beginning of each phase in order to make Prop. 1 applicable. As experimentally shown in Section 7, we observe that this empirically degrades the performance considerably.

In contrast, our approach is fully adaptive and pulls arms based on all of the past observations at every round. More specifically, at every round t, our algorithm chooses (but not pulls) a pair of arms, i_t and j_t, the gap of which needs to be estimated. Then, it pulls an arm so that the sequence of selected arms becomes close to

$$\mathbf{x}_n^*(i_t, j_t) = \arg\min_{\mathbf{x}_n} \|x_{i_t} - x_{j_t}\|_{(A^{\lambda}_{\mathbf{x}_n})^{-1}}. \tag{7}$$

Although Prop. 1 is no longer applicable to our strategy, it can focus on the estimation of the gaps between the best arm and near-optimal arms.

5 LinGapE Algorithm

In this section, we present a novel algorithm for (ε, δ)-best arm identification in LB. We name the algorithm *LinGapE (Linear Gap-based Exploration)*, as it is inspired by *Unified Gap-based Exploration* (UGapE, Gabillon et al., 2012).

The entire algorithm is shown in Algorithm 1. At each round, LinGapE first chooses (but does not pull) two arms, the arm with the largest estimated reward i_t and the most ambiguous arm j_t. Then, it pulls the most informative arm to estimate the gap of expected rewards $(x_{i_t} - x_{j_t})^\top \theta$ by Line 9 in Algorithm 1.

The algorithm for choosing arms i_t and j_t is presented in Algorithm 2, where we denote the estimated gap by $\hat{\Delta}_t(i,j) = (x_i - x_j)^\top \hat{\theta}_t^{\lambda}$ and the confidence interval of the estimation by $\beta_t(i,j)$ defined as

$$\beta_t(i,j) = \|x_i - x_j\|_{A_t^{-1}} C_t, \tag{8}$$

for C_t given in (3).

5.1 Arm Selection Strategy

After choosing arms i_t and j_t, the algorithm has to select arm a_t, which most decreases the confidence bound $\beta_t(i_t, j_t)$, or equivalently, $\|x_{i_t} - x_{j_t}\|_{A_t^{-1}}$. As in Soare et al. (2014), we propose two procedures for this.

Algorithm 1: LinGapE

Input: accuracy ε, confidence level δ, noise level R, norm S of unknown parameter θ, regularization parameter λ

Output: the arm $\hat{a}^*$ which satisfies stopping condition (1)

1 Set $A_0 \leftarrow \lambda I$, $b_0 \leftarrow \mathbf{0}$, $t \leftarrow 0$;
`// Initialize by pulling each arm once`
2 **for** $i \in [K]$ **do**
3 $t \leftarrow t+1$;
4 Observe $r_t \leftarrow x_i^\top \theta + \varepsilon_t$, and update A_t and b_t;
5 **Loop**
  `// Select which gap to examine`
6 $(i_t, j_t, B(t)) \leftarrow$ Select-direction(t);
7 **if** $B(t) \le \varepsilon$ **then**
8 **return** i_t as the best arm $\hat{a}^*$;
  `// Pull the arm based on the gap`
9 Pull the arm a_{t+1} based on (9) or (12) ;
10 $t \leftarrow t+1$;
11 Observe $r_t \leftarrow x_{a_t}^\top \theta + \varepsilon_t$, and update A_t and b_t;

Algorithm 2: Select-direction

1 **Procedure** `Select-direction(`t`):`
2 $\hat{\theta}_t^{\lambda} \leftarrow A_t^{-1} b_t$;
3 $i_t \leftarrow \arg\max_{i \in [K]} (x_i^\top \hat{\theta}_t^{\lambda})$;
4 $j_t \leftarrow \arg\max_{j \in [K]} (\hat{\Delta}_t(j, i_t) + \beta_t(j, i_t))$;
5 $B(t) \leftarrow \max_{j \in [K]} (\hat{\Delta}_t(j, i_t) + \beta_t(j, i_t))$;
6 **return** $(i_t,\, j_t,\, B(t))$;

One is to select arms greedily, which is

$$a_{t+1} = \arg\min_{a \in [K]} \|x_{i_t} - x_{j_t}\|_{(A_t + x_a x_a^\top)^{-1}}. \tag{9}$$

We were not able to gain a theoretical guarantee of the performance for this greedy strategy, though our experiment shows that it performs well.

The other is to consider the optimal selection ratio of each arm for decreasing $\|x_{i_t} - x_{j_t}\|_{A_t^{-1}}$. Let $p_k^*(i_t, j_t)$ be the ratio of arm k appearing in the sequence $\mathbf{x}_n^*(i_t, j_t)$ in (7) when $n \to \infty$. By the discussion given in Appendix C, we have

$$p_k^*(i_t, j_t) = \frac{|w_k^*(i_t, j_t)|}{\sum_{k=1}^K |w_k^*(i_t, j_t)|}, \tag{10}$$

where $w_k^*(i_t, j_t)$ is the k-th element of $\mathbf{w}^*(i_t, j_t)$ defined

as follows.

$$\mathbf{w}^*(i_t, j_t) = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}\|_1 \quad \text{s.t. } x_{i_t} - x_{j_t} = \sum_{k=1}^{K} w_k x_k, \tag{11}$$

where $\|\mathbf{w}\|_1$ is the l_1-norm of $\mathbf{w}$. The optimization is easier compared with Soare et al. (2014), which solved (5) and (6) via nonlinear convex optimization.

We pull the arm that makes the ratio of arm selections close to ratio $p_k^*(i_t, j_t)$. To be more precise, a_{t+1} is decided by

$$a_{t+1} = \arg\min_{a\in[K]:\, p_a^*(i_t,j_t)>0} T_a(t)/p_a^*(i_t, j_t), \tag{12}$$

where $T_a(t)$ is the number of times that arm a is pulled until the t-th round. This strategy is a little more complicated than the greedy strategy in (9) but enjoys a simple theoretical characteristic, based on which we conduct analysis.

LinGapE is capable of solving (ε, δ)-best arm identification, regardless of which strategy is employed, as stated in the following theorem.

Theorem 1. *Whichever the strategy in* (9) *or* (12) *is employed, arm $\hat{a}^*$ returned by LinGapE satisfies condition* (1)*.*

The proof can be found in Appendix D.

5.2 Comparison of Confidence Bounds

A distinctive character of LinGapE is that it considers an upper confidence bound of the *gap of rewards*, while UGapE and other algorithms for the LB, such as *Optimism in the Face of Uncertainty Linear bandit algorithm* (OFUL, Abbasi-Yadkori et al., 2011), consider an upper confidence bound of the *reward of each arm*. This approach is, however, not suited for the pure exploration in the LB, where the gap plays an essential role.

The following example illustrates the importance of considering such quantities. Consider that there are three arms, features of which are $x_1 = (-10, 10)^\top$, $x_2 = (-9, 10)^\top$, and $x_3 = (-1, 0)^\top$. Assuming that we have $\hat{\theta}_t^\lambda = (\hat{\theta}_{t,(1)}^\lambda, \hat{\theta}_{t,(2)}^\lambda)^\top = (-1, 0)^\top$, thus the estimated best arm is $i_t = 1$. Now, let us consider the case where we have already been confident that $\hat{\theta}_{t,(1)}^\lambda \approx -1$ but still unsure of $\hat{\theta}_{t,(2)}^\lambda \approx 0$. In such a case, algorithms considering an upper confidence bound of the rewards of each arm, such as UGapE, choose arm 2 as j_t, since it has a larger estimated expected reward and a wider confidence interval than arm 3. However, it is not efficient, since arm 2 cannot have a larger expected reward than arm 1 when $\theta_1 = -1$. On the other hand, LinGapE can avoid this problem, since the confidence interval for $(x_1 - x_3)^\top \hat{\theta}_t^\lambda$ is wider than $(x_1 - x_2)^\top \hat{\theta}_t^\lambda$.

6 Sample Complexity

In this section, we give an upper bound of the sample complexity of LinGapE and compare it with existing methods.

6.1 Sample Complexity

Here, we bound the sample complexity of LinGapE when arms to pull are selected by (12). Let the problem complexity H_ε be defined as

$$H_\varepsilon = \sum_{k=1}^{K} \max_{i,j\in[K]} \frac{p_k^*(i,j)\rho(i,j)}{\max\left(\varepsilon, \frac{\varepsilon+\Delta_i}{3}, \frac{\varepsilon+\Delta_j}{3}\right)^2}, \tag{13}$$

where Δ_i is defined as

$$\Delta_i = \begin{cases} (x_{a^*} - x_i)^\top \theta & (i \neq a^*), \\ \arg\min_{j\in[K]} (x_{a^*} - x_j)^\top \theta & (i = a^*), \end{cases} \tag{14}$$

and $\rho(i, j)$ is the optimal value of problem (11), denoted as

$$\rho(i,j) = \sum_{k=1}^{K} |w_k^*(i,j)| = \|\mathbf{w}^*(i,j)\|_1. \tag{15}$$

Now, the sample complexity can be bounded depending on the value of λ as follows.

Theorem 2. *Assume that a_t is determined by* (12)*. If $\lambda \leq \frac{2R^2}{S^2} \log \frac{K^2}{\delta}$, then the stopping time τ of LinGapE satisfies*

$$\mathbb{P}\left[\tau \leq 8H_\varepsilon R^2 \log\frac{K^2}{\delta} + C(H_\varepsilon, \delta) + K\right] \geq 1 - \delta, \tag{16}$$

where $C(H_\varepsilon, \delta)$ is

$$C(H_\varepsilon, \delta) = 4H_\varepsilon R^2 d \log\left(1 + \frac{4ML^2}{\lambda d}\right)$$

for $M = \frac{16H_\varepsilon^2 R^4 d L^2}{\lambda} + \left(8H_\varepsilon R^2 \log\frac{K^2}{\delta} + K\right)^2$.

Theorem 3. *If $\lambda > 4H_\varepsilon R^2 L^2$ and a_t is determined by* (12)*, then*

$$\mathbb{P}\left[\tau \leq \left(8H_\varepsilon R^2 \log\frac{K^2}{\delta} + 4H_\varepsilon \lambda S^2 + 2K\right)\right] \geq 1 - \delta. \tag{17}$$

The proofs can be found in Appendix D. These theorems state that there are two types of sample complexity. The first bound (16) is practically more applicable, since the condition $\lambda \leq \frac{2R^2}{S^2} \log \frac{K^2}{\delta}$ can be checked by known parameters. On the other hand, we cannot ensure whether the condition $\lambda > 4H_\varepsilon R^2 L^2$ is satisfied, since we cannot know H_ε in advance. However, the second bound in (17) can be tighter than the first one in (16) when $H_\varepsilon \ll d$.

6.2 Discussion of Problem Complexity

The problem complexity (13) has an interesting relation with that of the $\mathcal{XY}$-oracle allocation algorithm introduced by Soare et al. (2014). They considered the case where the agent knows the true parameter θ when selecting an arm to pull, and tries to *confirm arm a^* is actually the best arm.* Then, an efficient strategy is to let the sequence of arm selections $\mathbf{x}_n$ be

$$\arg\min_{\mathbf{x}_n} \max_{i \in [K] \setminus \{a^*\}} \frac{\|x_{a^*} - x_i\|_{A_{\mathbf{x}_n}^{-1}}}{\Delta_i}. \tag{18}$$

An upper bound of the sample complexity of $\mathcal{XY}$-oracle allocation was proved to be $\mathcal{O}(H_{\mathrm{oracle}} \log(1/\delta))$ in Soare et al. (2014), where problem complexity H_{oracle} is defined as

$$H_{\mathrm{oracle}} = \lim_{n \to \infty} \min_{\mathbf{x}_n} \max_{i \in [K] \setminus \{a^*\}} \frac{n \|x_{a^*} - x_i\|^2_{A_{\mathbf{x}_n}^{-1}}}{\Delta_i^2}.$$

This is expected to be close to the achievable lower bound of the problem complexity (Soare et al., 2014). Here, we prove a theorem that points out the relation between H_{oracle} and our problem complexity H_ε.

Theorem 4. *Let H_0 be the problem complexity of LinGapE* (13) *when ε is set as $\varepsilon = 0$. Then, we have*

$$H_0 \leq 72 H'_{\mathrm{oracle}} \leq 72 K H_{\mathrm{oracle}},$$

where H'_{oracle} is defined as

$$H'_{\mathrm{oracle}} = \sum_{i \in [K] \setminus \{a^*\}} \frac{\rho(a^*, i)}{\Delta_i^2}.$$

The proof of the theorem can be found in Appendix D.3. This result shows that our problem complexity matches the lower bound up to a factor of K, the number of arms. Furthermore, if Δ_i for some i is very small compared with $\{\Delta_{i'}\}_{i' \neq i}$, that is, if there is only one near-optimal arm, then H_{oracle} becomes close to H'_{oracle}, and hence our problem complexity H_0 achieves the lower bound up to a constant factor.

Soare et al. (2014) claimed that $\mathcal{XY}$-adaptive allocation achieves this lower bound as well. To be precise, they discussed that the sample complexity of $\mathcal{XY}$-adaptive allocation is $\mathcal{O}(\max(M^*, N^*))$, where N^* is the sample complexity of $\mathcal{XY}$-oracle allocation. Nevertheless, they did not give an explicit bound of M^*, which stems from the static strategy employed in each phase. Our experiments in Section 7 show that M^* can be as large as the sample complexity of $\mathcal{XY}$-static allocation, the problem complexity of which is proved to be $\Omega(4d/\Delta_{a^*}^2)$ and can be arbitrarily larger than H_{oracle} in the case of $d \to \infty$ (Soare et al., 2014). Therefore, LinGapE is the first algorithm that always achieves the lower bound up to a factor of K.

We point out another interpretation of our problem complexity. If set of features $\mathcal{X}$ equals the set of canonical bases $(e_1, e_2, \ldots, e_d)$, then the LB problem is reduced to the ordinary multi-armed bandit problem. In such a case, $p_k^*(i,j)$ and $\rho(i,j)$ are computed as

$$\rho(i,j) = 4,\ p_k^*(i,j) = \begin{cases} \frac{1}{2} & (k = i \text{ or } k = j), \\ 0 & (\text{otherwise}), \end{cases}$$

Therefore, if the noise variable is bounded in the interval $[-1, 1]$, which is known as 1-sub-Gaussian, the problem complexity becomes

$$H_\varepsilon = \sum_{k=1}^{K} \frac{2}{\max\left(\varepsilon, \frac{\varepsilon + \Delta_i}{3}\right)^2} \leq \frac{9}{8} H_\varepsilon^{\mathrm{UGapE}},$$

where $H_\varepsilon^{\mathrm{UGapE}}$ is the problem complexity of UGapE (Gabillon et al., 2012). This fact suggests that LinGapE incorporates the linear structure into UGapE from the perspective of the problem complexity.

Lastly, we mention one drawback of our algorithm, which is that the sample complexity is $\mathcal{O}(K \log \frac{1}{\delta})$, since $H_\varepsilon = \mathcal{O}(K)$. This is problematic when $K \gg d$, though, with a slight modification, we can derive another algorithm whose sample complexity is bounded by $O(d\sqrt{d} \log \frac{1}{\delta})$ (see Appendix B for the details).

7 Experiments

In this section, we compare the performance of LinGapE with the algorithms proposed by Soare et al. (2014) through experiments in two synthetic settings and simulations based on real data. All codes are available online[1].

7.1 Experiment on Synthetic Data

We conduct experiments in two synthetic settings. One is the setting where an adaptive strategy is suitable, and the other is where pulling all arms uniformly becomes the optimal strategy. We set the noise distribution as $\varepsilon_t \sim \mathcal{N}(0, 1)$ and run LinGapE with parameters $\varepsilon = 0$ and $\delta = 0.05$ in both cases. We tried various

[1] `https://github.com/liyuan9988/LinGapE`

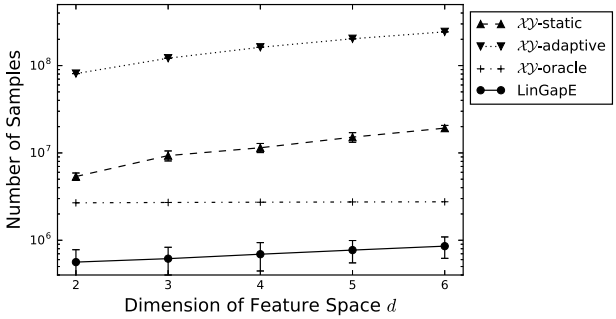

Figure 1: The number of samples required to estimate the best arm in the synthetic setting introduced by Soare et al. (2014).

values of regularization parameter λ and different arm selection strategies in (9) and (12), but they had very little impact on the performance. Hence, we plot the results only for the greedy strategy (9) and $\lambda = 1$. We repeated experiments ten times for each setting, the average of which is reported.

7.1.1 Setting Where the Adaptive Strategy is Suitable

The first experiment is conducted in the setting where the adaptive strategy is favored, which is introduced by Soare et al. (2014). We set up the LB problem with $d + 1$ arms, where features consist of canonical bases $x_1 = e_1, \ldots, x_d = e_d$ and an additional feature $x_{d+1} = (\cos(0.01), \sin(0.01), 0, \ldots, 0)^\top$. The true parameter is set as $\theta = (2, 0, \ldots, 0)^\top$ so that the expected reward of arm $d + 1$ is very close to that of the best arm $a^* = 1$ compared with other arms. Hence, the performance heavily depends on how much the agent can focus on comparing arms 1 and $d + 1$.

Figure 1 is a semi-log plot of the average stopping time of LinGapE, in comparison with the $\mathcal{XY}$-static allocation, $\mathcal{XY}$-adaptive allocation and $\mathcal{XY}$-oracle allocation algorithms, all of which are introduced by Soare et al. (2014). Their arm selection strategies are given in (5), (6) and (18), respectively. The result indicates the superiority of LinGapE to the existing algorithms.

The difference is due to the adaptive nature of LinGapE. To illustrate it, we present the number of times that each arm is pulled when $d = 5$ in Table 1. From the table, we can see that $\mathcal{XY}$-static allocation pulls all arms almost equally, while LinGapE and $\mathcal{XY}$-oracle allocation pull arm 2 more frequently. In fact, this is an efficient strategy, since pulling arm 2 significantly reduces the norm $\|x_1 - x_{d+1}\|_{A_{\mathbf{x}_n}^{-1}}$. From this result, we can infer that LinGapE figured out two potentially best arms, 1 and $d+1$, and changed the arm selection for focusing on comparing these arms.

Table 1: An example of arm selection when $d = 5$.

	$\mathcal{XY}$-static	LinGapE	$\mathcal{XY}$-oracle
Arm 1	1298590	2133	13646
Arm 2	2546604	428889	2728606
Arm 3	2546666	19	68
Arm 4	2546666	34	68
Arm 5	2546666	33	68
Arm 6	1273742	11	1

Although $\mathcal{XY}$-adaptive allocation has adaptive nature as well, it performs much worse than $\mathcal{XY}$-static allocation in this setting. This is due to the limitation that it has to reset the design matrix $A_{\mathbf{x}_n}$ at every phase. In fact, the algorithm succeeds in finding $\hat{\mathcal{X}}_j = \{1, d+1\}$ in the first few phases. However, in the next phase, the agent only pulls arms $1, 2$ and $d+1$ for estimating $(x_1 - x_{d+1})^\top \theta$, thus it cannot discard the sub-optimal arms any longer. Therefore, it handles all arms in the last phase, which requires as many samples as $\mathcal{XY}$-static allocation. We observed that the same happened in the two subsequent experiments and $\mathcal{XY}$-adaptive performed at least five times worse than $\mathcal{XY}$-static allocation. Hence, we omit the result for $\mathcal{XY}$-adaptive allocation in the following for highlighting differences of other methods.

It is somewhat surprising that LinGapE outperforms $\mathcal{XY}$-oracle allocation, given that the latter assumes access to the true parameter θ. The main reason for this is that our confidence bound is tighter than that used in $\mathcal{XY}$-oracle allocation. As discussed in Section 3, our confidence bound $\beta_t(i, j)$ is looser by a factor of $\sqrt{d}$ in the worst case where $\det(A_t) = \mathcal{O}(t^d)$. Nevertheless, $\det(A_t)$ grows almost linearly with t in this setting, where LinGapE mostly pulls the same arm as presented in Table 1. Therefore, the confidence interval is much narrower than the worst-case scenario. This suggests room of improvement in sample complexity (16), given that we only considered the worst-case in the derivation (see Prop. 3 in Appendix D).

7.1.2 Setting Where the Static Strategy is Optimal

We conduct another experiment in a synthetic setting, where $\mathcal{XY}$-static allocation is almost optimal. We consider the LB with $K = d = 5$, where the feature set $\mathcal{X}$ equals the canonical set $(e_1, e_2, \ldots, e_5)$. We set the parameter θ as $\theta = (\Delta, 0, \ldots, 0)^\top$, where $\Delta > 0$, hence arm 1 has a larger expected reward by Δ than all other arms. As $\Delta \to 0$, we need to estimate all arms equally accurately, therefore the optimal strategy is to pull all arms uniformly, which corresponds to $\mathcal{XY}$-static allocation.

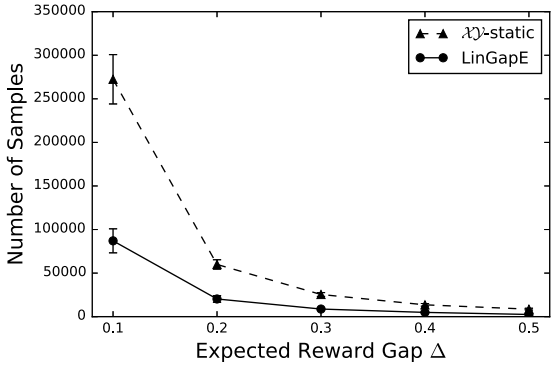

Figure 2: The number of samples required to estimate the best arm in the synthetic setting where the arm selection strategy in $\mathcal{XY}$-static allocation is almost optimal.

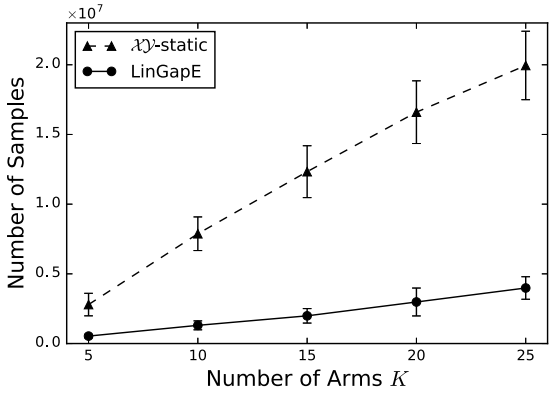

Figure 3: The number of samples required to estimate the best arm on Yahoo! Webscope Dataset R6A.

The result for various value gaps Δ is shown in Figure 2. We observe not only that LinGapE performs better than $\mathcal{XY}$-static allocation but also that the gap of the performance increases as $\Delta \to 0$, where $\mathcal{XY}$-static allocation approaches the optimal strategy. This is because the $\mathcal{XY}$-static allocation considers all arms until the stopping condition is satisfied, while LinGapE only considers arms that have not been turned out to be sub-optimal.

7.2 Simulation Based on Real Data

We conduct another experiment based on a real-world dataset. We use Yahoo! Webscope Dataset R6A[2], which consists of features of 36 dimensions accompanied with binary outcomes. It is originally used as an unbiased evaluation benchmark for the LB aiming for cumulative reward maximization (Li et al., 2010), and we slightly change the situation so that it can be adopted for pure exploration setting. We construct a 36-dimensional feature set $\mathcal{X}$ by random sampling from the dataset, and the reward is generated by

$$r_t = \begin{cases} 1 & \left(\text{w.p. } (1 + x_{a_t}^\top \theta^*)/2\right), \\ -1 & (\text{otherwise}), \end{cases}$$

where θ^* is the regularized least-squares estimator fitted for the original dataset. Although $x_{a_t}^\top \theta^*$ is not necessarily bounded in $[-1, 1]$, we observe that $x^\top \theta^* \in [-1, 1]$ for all features x in the dataset. Therefore, $(1 + x_{a_t}^\top \theta^*)/2$ is always a valid probability in this case. We compare the performance with the $\mathcal{XY}$-static allocation algorithm, where the estimation is given by the regularized least-squares estimator with $\lambda = 0.01$. The detailed procedure can be found in Appendix A.

[2] `https://webscope.sandbox.yahoo.com/`

The average number of samples required in ten simulations is shown in Figure 3, in which LinGapE performs roughly five times better than the $\mathcal{XY}$-static strategy, and the gap of performances increases as we consider more arms. This result shows the superiority of our algorithm.

8 Conclusions

In this paper, we studied pure exploration in an linear bandits. We first reviewed a drawback in an existing work, and then introduced a novel fully-adaptive algorithm, LinGapE. We proved that the sample complexity of LinGapE can match the lower bound and confirmed its superior performance in experiments. Since LinGapE is the first algorithm that achieves the lower bound, we will consider its various extensions and develop computationally efficient algorithms in our future work. In particular, pure exploration in the fixed budget setting is a promising direction of extension, since LinGapE shares many ideas with UGapE, which is known to be applicable in the fixed budget setting as well (Gabillon et al., 2012). Furthermore, as explained in Section 7.1, the derived sample complexity may be improved since evaluation of the determinant in Prop. 3 given in Appendix D is still loose. A bound based on tight evaluation of the determinant is left for future work.

Acknowledgements

We thank the anonymous reviewers for helpful comments. LX utilized the facility provided by Masason Foundation. JH acknowledges support by KAKENHI 16H00881, and MS acknowledges support by KAKENHI 17H00757.

Reference

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320. Curran Associates, Inc., 2011.

N. Abe and P. M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the 16th International Conference on Machine Learning*, pages 3–11, 1999. ISBN 1-55860-612-2.

J.-Y. Audibert and S. Bubeck. Best arm identification in multi-armed bandits. In *Proceedings of the 23th Conference on Learning Theory*, page 13 p., 2010.

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

K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J. (2)*, 19(3):357–367, 1967.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

W. Chu, S.-T. Park, T. Beaupre, N. Motgi, A. Phadke, S. Chakraborty, and J. Zachariah. A case study of behavior-driven conjoint analysis on Yahoo!: front page today module. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1097–1104. ACM, 2009.

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(Jun): 1079–1105, 2006.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems 25*, pages 3212–3220. Curran Associates, Inc., 2012.

M. Hoffman, B. Shahriari, and N. Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 365–374, 2014.

T. Lattimore and C. Szepesvari. The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 728–737, 2017.

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

H. Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.

M. Soare, A. Lazaric, and R. Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems 27*, pages 828–836. Curran Associates, Inc., 2014.

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1081–1088. ACM, 2006.