Robust Pure Exploration in Linear Bandits with Limited Budget

Ayya Alieva 1  Ashok Cutkosky 2  Abhimanyu Das 3

Abstract

We consider the pure exploration problem in the fixed-budget linear bandit setting. We provide a new algorithm that identifies the best arm with high probability while being robust to unknown levels of observation noise as well as to moderate levels of misspecification in the linear model. Our technique combines prior approaches to pure exploration in the multi-armed bandit problem with optimal experimental design algorithms to obtain both problem dependent and problem independent bounds. Our success probability is never worse than that of an algorithm that ignores the linear structure, but seamlessly takes advantage of such structure when possible. Furthermore, we only need the number of samples to scale with the dimension of the problem rather than the number of arms. We complement our theoretical results with empirical validation.

1. Introduction

A variety of problems across disciplines involve making decisions based on noisy observations. For example, online ad companies need to decide which ads to show to users based on noisy estimates of click rates, and machine learning practitioners need to tune hyperparameters based on performance on evaluation sets. All of these situations call for an exploration period, after which a decision is made. In the ads setting, the exploration is a testing phase prior to deployment. In machine learning, it is a model development or tuning phase. Further, in all these cases, the exploration phase is limited by a fixed budget: there is only a limited amount of clicks ad companies can use for estimating click rates for ads, or a hyperparameter tuning job will only have a certain amount of resources. This limitation means that the exploration phase should be somehow efficient - we wish to make the best use of our limited budget in order to maximize the chance that our final decision is the best one.

We model this problem as a pure exploration bandit problem (Even-Dar et al., 2002; Bubeck et al., 2009). We consider a finite set of n possible actions 1, which we call “arms”, such that each action x has an associated noisy reward y ∈ R. We are allowed to take an action (or “pull an arm”) and obtain an unbiased estimate of the corresponding reward. We will pull arms for a given exploratory period, observe the resulting reward estimates, and then output our best guess for the arm with the highest reward. The goal is to maximize the probability that our output is indeed the arm with the highest reward.

Often, we have some knowledge about relationships between different arms which we would like to take advantage of during the exploration phase. We model this prior knowledge by embedding our arms as x ∈ Rd in such a way that arms that seem qualitatively similar have similar embeddings. Note that this embedding is known - we either choose or are given the features for each arm. We then assume a linear model for the rewards y = θ · x for some fixed (but unknown) θ ∈ Rd. However, in deference to the reality that such a model is almost certainly not perfect, we allow for a certain amount of misspecification: y = θ · x + γ, where we assume γ is unknown but not too large. This setting can be described as the misspecified linear bandit. We provide an algorithm that, after exploring T arms, outputs a suboptimal arm with probability at most:

$$\log_2(n) \exp \left( - \frac{T}{\sigma^2 \log_2(n) \bar{H}_2} \right)$$

(1)

where $\bar{H}_2$ is an instance-dependent quantity (defined precisely in Section 2) that is bounded by

$$\frac{d}{\max(\Delta_2 - \sqrt{\Delta_2}, 0)^2},$$

where $\Delta_2$ indicates the gap in reward between the second-best arm and the best arm, and $\sigma^2$ is the variance of the observational noise. We also provide a lower bound that suggests that the above expression is tight up to logarithmic factors.

1In Section 6 we provide a covering argument that extends to infinite actions
2The above expressions supress some additional constants that appear in our formal results.
The pure exploration bandit problem has been studied by several previous authors (Audibert & Bubeck, 2010a; Karnin et al., 2013; Gabillon et al., 2012; Jamieson et al., 2014). Classically, the literature distinguishes between two cases, one in which the goal is to minimize the expected number of samples of an exploration phase required to guarantee a certain success probability of identifying the best arm, and one in which one must maximize the probability of finding the best arm given a certain exploration budget. These are called the fixed confidence and fixed budget settings respectively. The former case has been studied in the linear bandit setting in a number of prior works (Karnin, 2016; Tao et al., 2018; Soare et al., 2014; Xu et al., 2018; Degenne et al., 2020). Notice that sample complexity bounds for fixed confidence algorithms do not, in general, translate to bounds for the fixed budget setting, and furthermore, the sampling strategies for many fixed confidence algorithms require knowledge of a target success probability, which is not available for fixed budget problems.

Indeed, surprisingly little work (e.g. (Hoffman et al., 2014; Katz-Samuels et al., 2020)) has focused on the fixed-budget setting in linear bandits, which is the focus of this paper. Furthermore, there appears to be a dearth of work on the problem of misspecified linear bandits in the pure exploration setting. Prior work on linear bandits (Karnin, 2016; Hoffman et al., 2014; Katz-Samuels et al., 2020) seems to rely heavily on the assumption that the linear model is correct, and it is unclear to what extent their results will degrade when this assumption is violated. Since, in reality, no linear model is likely to be completely correct, this limits the theoretical guarantees of these algorithms in practical situations. Nevertheless, it is intuitively the case that even a linear model with modest misspecification should provide some advantage over simply ignoring the feature vectors.

Our algorithm, in addition to obtaining the bound (1), guarantees a mistake probability that gracefully degrades with the level of misspecification. Although our mistake probability without misspecification does not quite match the recent bound in (Katz-Samuels et al., 2020) for non-misspecified settings, note that our algorithm does not need to know what the level of misspecification is in advance (i.e. it is adaptive), and moreover we provide an explicit polynomial-time algorithm, while (Katz-Samuels et al., 2020) involves running mirror descent on a subproblem that is only provably convex in particular scenarios. Beyond misspecification, we automatically adapt to the variance of the random observations, again without requiring this parameter as input. Additionally, our results can also be extended to provide guarantees for the case of problem independent bounds (independent of the \( \Delta_i \)).

A closely related problem to pure exploration in bandits is the problem of minimizing regret. In this setting, the algorithm attempts to minimize the total loss obtained over the set of pulled arms, thus mixing the “exploration” phase with an “exploitation” phase. This setting has been extensively studied in both the general (Auer & Ortner, 2010; Kaufmann et al., 2012; Audibert & Bubeck, 2010b) and linear bandit (Abbasi-Yadkori et al., 2011; Srinivas et al., 2009; Agrawal & Goyal, 2013) setting. Some work has also been done on the problem on misspecified linear models in the cumulative regret setting. (Ghosh et al., 2017; Gopalan et al., 2016) provide algorithms that consider misspecified linear bandits, and show how to obtain small regret when the vector of misspecifications \( \gamma \in \mathbb{R}^n \) is bounded in 2-norm. This is a strong restriction on misspecification, but in fact (Ghosh et al., 2017) shows that for the case of minimizing regret, more moderate levels of misspecification may completely destroy any chance for improved performance. In contrast, in our pure exploration setting, we are able to handle \( \gamma \) bounded in \( \infty \)-norm, which is much less restrictive.

We organize this paper as follows: in Section 2 we formally describe our setting and notation. In Sections 3 and 4 we describe our algorithm and give its analysis. In Section 5 we provide our lower bound, and in Section 6 we extend our results to obtain a problem independent bound for the mistake probability. In Section 7 we provide a few experimental examples, and in Section 8 we provide some concluding remarks and open problems.

2. Problem Statement

In the pure exploration linear bandit setting, a player is given a set of \( n \) arms \( \mathcal{A} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d \) with \( \|x_i\| \leq 1 \) for all \( i \) and \( n \geq d \). Each \( x_i \) is associated with an expected reward \( y_i \). Each \( y_i \) takes the form \( y_i = \langle \theta, x_i \rangle + \gamma_i \), where \( \theta \) is some vector in \( \mathbb{R}^d \), and \( \gamma_i \) is the deviation from the linear model. Neither \( \theta \) nor \( \gamma \) is known to the player. We define \( \gamma_{\max} = \max_i |\gamma_i| \). At each step of the game, the player chooses one arm \( x_i \) of their choice and observes an independent sample \( \hat{y}_i = y_i + \zeta \) where \( \zeta \) is mean-zero \( \sigma \)-subgaussian random variable that is independent of the past history of the game, where \( \sigma \) is also not known to the player. The goal of the game is to query at most \( T \) arms \( x_{i_1}, \ldots, x_{i_T} \) and then output the arm with the highest expected reward.

For ease of notation, we assume that the arms are enumerated in order of the decreasing expected reward, i.e. \( y_1 \geq y_2 \geq \cdots \geq y_n \). We also assume that the set of arms \( \{x_1, \ldots, x_n\} \) spans \( \mathbb{R}^d \). This is without loss of generality, as we may always change coordinates to a subspace spanned by the arms. We define the gap \( \Delta_i = y_i - y_1 \) for \( i > 1 \), i.e. the difference between the expected rewards of the best arm and the \( i \)th best arm. We also define \( N = \frac{T}{\log_2 n} \). Further, to simplify our presentation we assume that \( n, d \) and \( \frac{T}{\log_2 (n)} \)
are powers of 2, and there is a unique best armootnote{These simplifying assumptions allow us to avoid $\lceil x \rceil$ operators in several places.}.

We will also use the following non-standard notation: given a symmetric positive semidefinite matrix $M$, and a vector $x$, we use $M^T x$ to indicate either $\infty$ if $x$ is not in the range of $M$, or $\arg\min_{y, My = x} \|y\|_2$ otherwise. This notation is well-defined because the kernel and range of a symmetric positive semi-definition matrix are orthogonal subspaces. Intuitively, $M^T$ is similar to a pseudo-inverse, but instead of sending kernel elements to 0, we send them to $\infty$. We will also refer to the matrix norm of $x$ with respect to $M$ as $\|x\|_M := \sqrt{x^T M x}$.

We will refer to the classical bandit problem in which the arms also decreases.

This quantity frequently appears in error probabilities of algorithms. For example, (Karnin et al., 2013) obtains error

$$H_{\min(3, H_2^T)} \leq \frac{T}{4 \log 2(n)}$$

In our linear bandit setting, we define a similar quantity:

$$\hat{d} = \inf_v \inf_{\|x\|_2 \leq \|x\|_2} \sup_{\pi \in [0, N]} \|x - v\|_2^2$$

$$h_i = \begin{cases} \min\{\hat{d}, i\} & \text{if } i > \frac{T}{4 \log 2(n)} \\ \min\{\hat{d}, i\} & \text{if } i \leq \frac{\|x\|_2(n)}{4 \log 2(n)} \end{cases}$$

$$\tilde{H}_2 = \max_{1 \leq i \leq n} \max(\Delta_i - (2 + 2\sqrt{2\tilde{H}_2}) \gamma_{\max}, 0)^2$$

We will show that $\hat{d} \leq d$, so that $\hat{d}$ cannot be very large even in the worst-case. In the next section, we will also draw a connection between $\tilde{H}_2$ and the notion of “characteristic time” (Degene et al., 2020) used in lower bounds for the fixed confidence problem setting.

In the absence of misspecification, $\tilde{H}_2$ can be bounded by $\min\{3\tilde{H}_2, \frac{\tilde{H}_2}{2\Delta^2}\}$. Our bounds will depend on $\tilde{H}_2$ in a way analogous to how bounds for the multi-armed bandit setting depend on $H_2$. Larger values of $\tilde{H}_2$ correspond to more difficult problems. As $\Delta_i - (2 + \sqrt{\tilde{H}_2}) \gamma_{\max}$ decreases, the arms become harder to distinguish, and as $\tilde{d}$ increases, our ability to utilize linear structure to gain information about the arms also decreases.

3. Algorithm

Our approach is based on the sequential halving algorithm of (Karnin et al., 2013). We construct a sequence of “candidate sets” $S_0, \ldots, S_M$ for each of $M = \log_2(n)$ rounds such that $S_0 = \mathcal{A}_0$ and $|S_M| = 1$. To get $S_{m+1}$ from $S_m$, we pick $N = \log_2(n)$ points $x_{m,1}, \ldots, x_{m,N}$ in $\mathcal{A}_m$ and receive rewards $y_{m,1}, \ldots, y_{m,N}$ where $y_{m,i} = \langle (\theta, x_{m,i}) + \gamma_{x_{m,i}}, \mathcal{A}_m \rangle$. Then, we perform linear regression to get an estimate $\hat{\theta}_m$. Next, for each element $x \in S_m$, we compute $\hat{y}_m = \langle (\hat{\theta}_m, x) \rangle$. Finally, we order the elements of $S_m$ according to the values of $\hat{y}_i$ and remove the bottom half of these elements to obtain $S_{m+1}$, breaking ties arbitrarily. The final output of our algorithm is the sole element of $S_M$.

A key step in this procedure is the choice of the $N$ arms we pull in every round. We choose these arms using an optimal experimental design algorithm. Specifically, we deploy the method described by (Allen-Zhu et al., 2017), which we refer to as OptDesign. This algorithm provides a way to choose a discrete subset of arms whose covariance matrix approximately optimizes a certain objective, which arises organically in the analysis of the failure probability of our approach. In order to employ this result however, we will require $T \geq 45d \log_2(n)$ due to technical limitations on the algorithm of (Allen-Zhu et al., 2017). This requirement is not very restrictive, since $T$ must be at least $\Omega(d)$ for us to be able to even sample all dimensions.

We provide the pseudocode for our pure exploration algorithm in Algorithms 1 and 2.

Algorithm 1 LinearExploration

Input: total budget $T$, set of arms $\mathcal{A} = \{x_1, \ldots, x_n\}$.

Initialize: $S_0 = \{x_1, \ldots, x_n\} = \mathcal{A}$, $m \leftarrow 0$, $N \leftarrow \lceil \frac{T}{\log_2(n)} \rceil$.

while $|S_m| > 1$ do

Pick $Z_m \leftarrow \{x_{m,1}, \ldots, x_{m,N}\}$ ← GetArms($S_m$, $N$, $\mathcal{A}$).

Sample each arm $x_{m,i} \in Z_m$ to obtain a reward estimate $\hat{y}_{m,i}$.

Compute minimum OLS estimate

$$\hat{\theta}_m = \arg\min_{\theta \in \text{span}(Z_m)} \sum_{i=1}^{N} (\langle \theta, x_{m,i} \rangle - \hat{y}_{m,i})^2$$

Set $S_{m+1}$ be the set of $|S_m|/2$ arms in $S_m$ with largest values of $\hat{y}_m = \langle (\hat{\theta}_m, x) \rangle$.

$m \leftarrow m + 1$

end while

Return The unique element of $S_m$.

Theorem 1. Let $T > 45d \log_2(n)$ and $2 + 2\sqrt{2\tilde{H}_2} \gamma_{\max} \leq \Delta_i$ for all $i$. Then LinearExploration makes at most $T$ arm pulls, and fails to return the optimal arm $x_1$ with probability at most:

$$3 \log_2(n) \exp\left(-\frac{T}{16 \log_2(n) \tilde{H}_2 \sigma^2}\right)$$
We also have the following important bounds on $\tilde{d}$, which is an immediate consequence of Lemma 11, using the Kiefer-Wolfowitz theorem (Kiefer & Wolfowitz, 1960).

**Proposition 2.** $\tilde{d} \leq d$

It is helpful to compare Theorem 1 to the result for ordinary multi-armed bandit exploration of (Karnin et al., 2013), which obtains an error probability of:

$$3 \log_2(n) \exp\left(-\frac{T}{16 \log_2(n) \sigma^2 \max_{i \leq n} i \Delta_i^2}\right)$$

Notice that in case of a strictly linear model (for which $\gamma_i = 0$ for all $i$), our definition of $\tilde{H}_2$ simplifies to:

$$\tilde{H}_2 = \max_{1 \leq i \leq n} \frac{h_i}{\Delta_i^2} \leq \min\left(3H_2, \frac{7d}{4 \Delta_2^2}\right)$$

Hence, our dependence on $\tilde{H}_2$ improves on the ordinary multi-armed bandit bound (2). Moreover, our algorithm’s performance can be directly bounded in terms of the dimension of the space $d$ rather than any other geometric properties of the arrangement of the arms. Even if the linear structure is essentially useless (i.e. $d \geq n$), we decay gracefully to near-optimal bounds for the pure-exploration multi-armed bandit problem:

For a tighter characterization of our algorithm’s performance in terms of the geometric structure of the arms, it is also instructive to consider the following bound on $\tilde{H}_2$

$$\tilde{H}_2 \leq \frac{7}{4 \Delta_2^2} \sup_{\|\pi\|_1 \leq N} \sup_{x_i, \pi(a) a a^T} (x_i - x_1)^T (\sum_{a \in A} \pi(a) a a^T) (x_i - x_1)$$

This expression simplifies to:

$$\frac{7}{4 \Delta_2^2} \inf_{\|\pi\|_1 \leq N} \sup_{x_i, \pi(a) a a^T} (x_i - x_1)^T (\sum_{a \in A} \pi(a) a a^T) (x_i - x_1)$$

Although it is not clear how to convert this result into a lower bound for our fixed-budget setting, this at least provides intuitive evidence that our error bound is measuring the difficulty of the problem using the “right” geometric quantities.

Further, note that our algorithm does not need to sample all of the possible arms: it is possible for $T < |A|$. This property highlights how our method is using the linear structure: by performing regression, we learn about arms that we have never pulled, and are able to infer whether they are worth considering or not.

Finally, comparing the output of OptDesign to a uniform sampling allows us to always at least match the error bound of successive halving. If we were to run OptDesign by itself, the variance of the estimates might be up to 3 times higher than the uniform sampling approach (due to error in optimization and rounding from fractional to integer solutions). In practice, this would rarely be an issue since it is rare for the best experiment design to be uniform sampling. Yet, for the sake of exposition we include the comparison in our algorithm.

We prove Theorem 1 by examining the probability of discarding the correct arm in each round of the loops in the algorithm. Then, we use union bound to bound the overall error probability, in a manner roughly analogous to the strategy followed by (Karnin et al., 2013).

To this end, we first need the following Lemma (whose proof is deferred to the appendix) that bounds the probability of a single arm being misordered when deciding which arms to remove in a round:

**Lemma 3.** Assume that the best arm was not eliminated prior to round $m$. Let $|x|_+ = \max(x, 0)$. Then for any arm
We also require the following Lemma:

**Lemma 4.** Assume that the best arm was not eliminated prior to round \( m \), and let \( |x|_+ = \max(0, x) \). Then the probability that the best arm is eliminated on round \( m \) is at most

\[
P[\langle \hat{\theta}_m, x \rangle > \langle \hat{\theta}_m, x_i \rangle] \leq \exp\left(-\frac{\left[\Delta_1 \left(2 + \sqrt{2h_{|S_m|/4}\gamma_{\max}}\right)\right]^2 T}{16 \log_2(n)h_{|S_m|/4}\sigma^2}\right)
\]

Proof. If the best arm is thrown out at round \( m \), there are at least \( \frac{1}{2}|S_m| \) arms in \( S_m \) whose \( \hat{y} \) estimates are higher than that of the best arm. Let \( S_m' \subseteq S_m \) be the set of arms that excludes the \( \frac{1}{2}|S_m| \) arms with the largest true means in \( S_m \). If the best arm is thrown out, then at least \( \frac{1}{2} \) of arms in \( S_m' \) must have higher \( \hat{y} \) estimates than that of the best arm. Let \( N_m \) be the number of such arms. Define \( D = \max\left\{\left(\Delta_1 |S_m'| - (2 + 2\sqrt{2h_{|S_m|/4}\gamma_{\max}})^2, 0\right)\right\} \). Then using Lemma 3, the expected number of such arms is at most

\[
E[N_m] = \sum_{x_i \in S_m'} P[\langle \hat{\theta}, x_i \rangle \geq \langle \hat{\theta}, x_i \rangle] \\
\leq |S_m'| \exp\left(-\frac{DT}{16 \log_2(n)h_{|S_m|/4}\sigma^2}\right)
\]

Then, by Markov inequality, the probability of the best arm being thrown out at round \( m \) is at most

\[
P\left[N_m > \frac{1}{3}|S_m'|\right] \leq \frac{E[N_m]}{\frac{1}{3}|S_m'|} \\
\leq 3 \exp\left(-\frac{DT}{16 \log_2(n)h_{|S_m|/4}\sigma^2}\right)
\]

4. Analysis of GetArms

In this section, we provide an analysis of the GetArms procedure that is used to choose which arms to pull in each round of LinearExploration. The heavy-lifting here is performed by the experimental design algorithm proposed by (Allen-Zhu et al., 2017), which we refer to as OptDesign. Given \( S \subseteq A \subseteq \mathbb{R}^d \), a number \( N \) and the objective \( f_S : S^d \rightarrow \mathbb{R} \):

\[
f_S(X) = \sup_{x_i, x_j \in S} (x_j - x_i)^T X^\dagger (x_i - x_j)
\]

OptDesign returns a set \( \hat{Z} = \{z_1, \ldots, z_n\} \subset A \) such that \( \hat{X} = \sum_{z \in \hat{Z}} z z^\top \) nearly minimizes \( f_S(X) \). We verify technical conditions required of \( f_S \) in order to employ OptDesign in Appendix C.

We use OptDesign to obtain a near-optimal set of arms to pull. GetArms then potentially improves this near-optimal set by comparing to a few special-case candidate sets of arms. By comparing to these special cases, we mitigate the risk that the approximation factor in the discrete optimization algorithm will cause worse performance than an algorithm that ignores the linear structure.

Our main result is the following, proved in Appendix B:

**Theorem 5.** On inputs \( S, N \) and \( A \), GetArms runs in time polynomial in \( |S|, N, |A| \) and \( d \) and produces a set \( \hat{Z} \subset A \) such that \( \|\hat{Z}\| \leq N \) that satisfies:

\[
\sup_{x_i, x_j \in S} (x_j - x_i)^T \left(\sum_{z \in \hat{Z}} z z^T\right) (x_j - x_i) \leq 2 \frac{h_{|S|/4}}{N}
\]

Note that this theorem does not use any geometric properties of the arms (e.g. no condition numbers). The quantity \( h_{|S|/4} \) depends only on the dimension and the relative gaps between the arms.

5. Lower Bound

We next provide a lower bound for the pure exploration linear bandit problem without misspecification, suggesting...
that $H_2$ is a complex measure for this problem. 
Our lower bound is a reduction to the ordinary multi-armed 
bayesian lower bound (Audibert & Bubeck, 2010a). This result states 
that for any $p \in (0, 1/2)$, for any algorithm, there 
exists a $d$-armed bandit with true rewards in $[p, 1 - p]$ and 
whose observed values are Bernoulli random variables such 
that the probability of successfully identifying the best 
arm is at most $1 - \exp \left( \frac{-(5 + o(1))T}{p(1-p)H_2^{1/2}} \right)$, where $H_2 = \max_i \Delta_i^{-2}$. 
Using this, we have the following: 

**Theorem 6.** Given a $d$-dimensional linear bandit pure-
exploration algorithm, any $p \in (0, 1/2)$, and any $n \geq d$, 
there exists an algorithm instance on which the probability of 
identifying the best arm is at most $1 - \exp \left( \frac{-(15 + o(1))T}{p(1-p)H_2^{1/2}} \right)$, 
where the $o(1)$ depends on $n$, $T$ and $p$ and goes to zero as $T \to \infty$.

Note that this is not an instance dependent lower bound. We also provide a lower bound that depends on the geometric 
structure of the problem. We first define some notation: for any vector $w \in \mathbb{R}^d$ and real number $\gamma$, we define the 
design matrix $V_{w, \gamma} := \sum_{i=1}^d w_i x_i x_i^T$, and $V_{w, \gamma}$ to be $V_{w, \gamma} = V_w + \gamma I_d$, where $I$ is the identity matrix in $\mathbb{R}^{d \times d}$. We define $D_{\gamma}$ as the probability simplex of dimension $d$. Our lower bound for any problem instance with dimension $d$, arms $x_i \in A$ and parameter vector $\theta \in \mathbb{R}^d$ is in terms of the quantity $H_{LB} = \min_{w \in D_{\gamma}} \max_{x \neq x_1} \min_{\gamma \in \mathbb{R}} \frac{\|x_1 - x\|^2}{\max(\theta^T V_{w, \gamma} V_{w, \gamma}(x_1 - x), 0)^2}$.

**Theorem 7.** Given a linear bandit pure-exploration algorithm, 
there exists an algorithm instance on which the probability of 
identifying the best arm is at most $1 - \exp \left( -T \cdot (1/\sqrt{H_{LB}} + 2 \sin \frac{\pi}{2})^2 \right)$.

The $H_{LB}$ term in the above geometry-dependent lower bound 
whose proof is deferred to the appendix) plays a similar role as the $H_2$ quantity in our upper bound. We defer closing the gap between these two terms to future work.

### 6. Problem Independent Bound

Now we provide an analysis of our algorithm that obtains a problem independent bound. This means that the mistake 
probability will not depend on $\Delta_i$. We do this by redefining 
the definition of a mistake: instead of bounding the 
probability of returning the best arm, we bound the probability of 
returning an arm with reward close to highest reward.

To facilitate our discussion, we define the value $x_{1,m}$ to 
be the best arm remaining in $S_m$, and $y_{1,m}$ to be the 
extected reward of this arm. Thus the final output of the 
algorithm has expected reward $y_{1,\log_2(n)}$. We will show that 
the probability that $y_{1,m+1} < y_{1,m} - \epsilon$ is small for any given $\epsilon$. Then, by union bound, we will obtain with high 
probability $y_{1,\log_2(n)} \geq y_1 - \log_2(n)\epsilon$. Specifically, we have the following lemma:

**Lemma 8.** The probability that $y_{1,m+1} < y_{1,m} - \epsilon$ is at most:

$$3 \exp \left( -\frac{\max_i \left( \epsilon - (2 + \sqrt{2h_i |S_m|/4} \gamma_{max})^2 \right) T}{\log_2(n) h_i |S_m|/4} \right)$$

**Proof.** Define $\Delta_i \subseteq \Delta_{i,m}$ to be the gap between the ith 
best arm and the best arm remaining in $S_m$. Then notice that 
the proof of Lemma 3 still holds if we replace $x_1$ with the best arm remaining in $S_m$ and $\Delta_i$ with $\Delta_{i,m}$

Let $S'_m$ be the set of arms in $S_m$ with $y$ value less than $y_{1,m} - \epsilon$. Notice that in order for $y_{1,m+1}$ to be less than $y_{1,m} - \epsilon$, we must have $|S_m|/2$ elements of $|S_m'|$ to have $\bar{y}$ values larger than those of the best arm left in $S_m$. Let $S''_m$ be the set of arms that excludes the $1/2|S_m'|$ arms with highest true mean from $S'_m$. Then if $y_{1,m+1} < y_{1,m}$, we 
then have $1/2$ of the arms in $S''_m$ have higher $\bar{y}$ estimates than 
the best arm in $S_m$. Let $N_m$ be the number of such arms. Define $D = \max\{\epsilon - (2 + \sqrt{h_i |S_m|/4} \gamma_{max})^2, 0\}$.

Then, since all arms in $S_m'$ have $\Delta_{i,m} \geq \epsilon$, we have by Lemma 3 that

$$\mathbb{E}[N_m] \leq |S'_m'| \exp \left( -\frac{DT}{\log_2(n) h_i |S_m'|/4} \right)$$

So using Markov inequality in exactly the same way as in 
the proof of Lemma 3, the conclusion follows.

This Lemma allows us to prove an analog of Theorem 1:

**Theorem 9.** For any given $\epsilon > 0$, the probability that 
$\text{LinearExploration}$ returns an arm with true mean 
more than $\epsilon \log_2(n)$ lower than $y_1$ is at most:

$$3 \log_2(n) \exp \left( -\inf_i \frac{\max{\epsilon - (2 + \sqrt{2h_i \gamma_{max}})^2} T}{\log_2(n) h_i} \right)$$

Note that the $h_i$ in the bound do not depend on the problem 
instance. The proof is identical to the proof of Theorem 1, 
using Lemma 8 in the same way as Lemma 4.

In addition to verifying the natural intuition that our algorithm 
performs reasonably even with very small $\Delta_2$, Theorem 9 lets us design an algorithm that allows for an infinite 
number of arms. For example, we might consider the situation in which $A = \{x : \|x\| \leq 1\}$. The procedure is 
computationally inefficient, but straightforward. We require 
only one more assumption, namely that $\|\theta\| \leq 1$. For simplicity of exposition, also assume that our bandit problem 
is well-specified ($\gamma_{max} = 0$), although this is not necessary. Given an error tolerance $\epsilon$, we choose a subset of the arms, $A$, of cardinality at most $\epsilon^{-d}$ such that any arm in $A$ is at
most $O(\varepsilon)$ far away in the 2-norm from an arm in $\tilde{A}$. Such a choice is possible because we assumed every arm in $\mathcal{A}$ has norm at most one. Since $\|\theta\| \leq 1$, this implies that the highest expected reward in $\tilde{A}$ is at most $O(\varepsilon)$ away from the highest expected reward in $\mathcal{A}$.

Our approach is then to run LinearExploration on $\tilde{A}$ rather than $\mathcal{A}$. Since $\log_2(|\tilde{A}|) \leq d \log_2(1/\varepsilon)$, by Theorem 9, with probability at least

$$1 - 3d \log_2(1/\varepsilon) \exp\left(-\frac{\varepsilon^2 T}{d \log_2(1/\varepsilon) h_e}\right)$$

the returned arm’s expected reward is optimal to within $O(d \log_2(1/\varepsilon))$ in $\tilde{A}$, and so within $O(d \log_2(1/\varepsilon))$ of the best expected reward in $\mathcal{A}$.

Clearly, this method is slow computationally. If $\mathcal{A}$ has some structure, it may be possible to reduce the amount of computation. We leave this question for future work.

### 7. Experiments

In this section, we empirically validate our approach via synthetic experiments. We tested our algorithm on both truly linear problems (misspecification of 0) and problems with misspecification of order $O(1/d)$.

In Figure 1, we tested four different distributions of arms: a Uniform distribution where each arm was sampled uniformly from a hypercube $\{x : \|x\|_\infty \leq 1\}$; a Gaussian-Mixture distribution where each arm was sampled from a mixture of three gaussians with constant variance and three different basis vectors chosen as the means; a Clusters distribution where arms are sampled uniformly from one of two hypercubes of size $[-\epsilon, \epsilon]^d$ for a small $\epsilon = 0.05$, one centered at $\theta$ and one at $-\theta$; and Subspace, where arms were sampled uniformly from $[-1, 1]^{d-1} \times [-\epsilon, \epsilon]$, resulting in arms being close to a subspace of $R^{d-1}$. The experiments used $d = 3$, $n = 50$, $T = 800$, and are averaged over 200 trials.

We compared LinearExploration to two baseline fixed-budget algorithms: the successive halving approach from (Karlin et al., 2013), which ignores the linear structure, and the BayesGap algorithm from (Hoffman et al., 2014). In addition, we also compare against the XY-Allocation fixed confidence algorithm from (Soare et al., 2014), where we ignore the stopping criteria and run for the full time budget. We report results with the static version of the XY-Allocation - the adaptive version (and many other adaptive fixed confidence algorithms) requires knowledge of a target success probability for their arm-selection strategy, which is not available in the fixed budget setting. Finally, we also compare LinearExploration against a non-adaptive variant (denoted as Non-Adaptive), where we just run a single round of GetArms and spend the entire budget in a single experimental design call.

Except for the hard Clusters instance (where all the algorithms perform relatively poorly due to the small gap between arms), LinearExploration performed better than the other baselines in most instances. Note that unlike LinearExploration, BayesGap algorithm (Hoffman et al., 2014) requires hyperparameters $\sigma$, the variance of the noise, and $\eta$, a prior on the variance of $\theta$. In Figure 1, we used the true values for both $\eta$ and $\sigma$. However, these values are unlikely to be known apriori in most applications, which would degrade the performance of BayesGap. All algorithms perform better on simpler arm distributions, but LinearExploration seems to exploit the linear structure significantly better than the baselines. In particular, while XY-Allocation also utilizes linear structure, it is outperformed by LinearExploration, possibly because fixed-confidence algorithms do not explicitly utilize knowledge of the time horizon that is available in the fixed-budget setting. LinearExploration also outperforms...
Robust Pure Exploration in Linear Bandits with Limited Budget

Figure 2. Probabilities of algorithms returning the best arm under varying level of misspecification and number of dimensions. Arms are sampled iid uniformly from a bounded hypercube. Error bars are the standard error, averaged over 200 trials. Budget is $T = 80$, number of arms is $n = 20$. In the figure on the right, misspecification is 0, and dimensions vary from $d = 3$ to $d = 10$. In the figure on the left, arms have dimensions $d = 3$ and the misspecification is sampled uniformly iid for each arm, varying $\|\gamma\|_{\infty}$ from 0 to $\frac{1}{3}$. Its Non-Adaptive counterpart, confirming that our algorithm’s use of multiple adaptive rounds of experiment design are valuable.

Figure 2 compares the performance of our algorithm as a function of varying misspecification and number of dimensions $d$, while fixing $n = 20$ and $T = 80$ for the Uniform distribution instance. For BayesGap, we use the shaded grey area to illustrate its dependence on the hyperparameters $\sigma$ and $\eta$. The shaded part shows the one standard error confidence bound on performance of BayesGap under different reasonable choices of the two parameters. For instance, while the real $\sigma$ and real $\eta$ are both equal to 1.0 in our setting, the grey area was sampled from 0.5 to 3.0 for $\sigma$, and from 0.5 to 2.0 for $\eta$. LinearExploration, Non-Adaptive, Successive Halving, and XY Allocation (Soare et al., 2014) do not require any prior knowledge, so those experiments are plotted as lines with standard error bars.

Since Successive Halving algorithm does not rely on linear structure, its performance does not vary with the number of dimensions or misspecification levels. Similarly, as BayesGap heavily relies on its Bayesian prior, its performance is affected by the specification of $\sigma$ and $\eta$, rather than by misspecification and number of features. While this makes these two baselines seem more robust to a misspecified linear model, this also implies that neither of the two algorithms can successfully leverage the information encoded in the linear structure, as showcased by Figure 2. XY-Allocation and Non-Adaptive do rely on linear structure, but not as effectively as LinearExploration, as seen from the figures. Thus, even though LinearExploration degrades with the level of misspecification and the number of features, the algorithm still outperforms the other baselines as long as the linear structure is able to provide some information.

8. Conclusion

We have introduced a algorithm for pure exploration in the linear bandits setting. Our algorithm is robust in the sense that it does not require any input relating to the distribution of the observation noise, and is even able to handle modest amounts of misspecification. We further demonstrate that our mistake probability is nearly tight in the well-specified case. We also show that we can bound method enjoys a problem independent mistake probability - that is, a mistake probability that does not depend on the gaps in the values of the arms. We leverage this observation to design an algorithm for the case of infinitely many potential arms. We support our theoretical contributions with an empirical study verifying the desirable properties of our algorithm.

While our work represents a step towards robustifying pure exploration problems to misspecification, we hope for further improvements. For example, we require the misspecification to be a factor of $\sqrt{n} \leq \sqrt{d}$ smaller than the gap between the $i$th best and the best arm. It is unclear if this is tight in general. It might be possible to relax this requirement in some settings and do well so long as the misspecification does not actually reorder the linear model.
Robust Pure Exploration in Linear Bandits with Limited Budget

References


