A Time and Space Efficient Algorithm for Contextual Linear Bandits

José Bento[1], Stratis Ioannidis[2], S. Muthukrishnan[3], and Jinyun Yan[3]

[1] Stanford University, jbento@stanford.edu
[2] Technicolor, stratis.ioannidis@technicolor.com
[3] Rutgers University, {muthu, jinyuny}@cs.rutgers.edu

Abstract. We consider a multi-armed bandit problem where payoffs are a linear function of an observed stochastic contextual variable. In the scenario where there exists a gap between optimal and suboptimal rewards, several algorithms have been proposed that achieve $O(\log T)$ regret after T time steps. However, proposed methods either have a computation complexity per iteration that scales linearly with T or achieve regrets that grow linearly with the number of contexts $|\mathcal{X}|$. We propose an ϵ-greedy type of algorithm that solves both limitations. In particular, when contexts are variables in $\mathbb{R}^d$, we prove that our algorithm has a constant computation complexity per iteration of $O(poly(d))$ and can achieve a regret of $O(poly(d) \log T)$ even when $|\mathcal{X}| = \Omega(2^d)$. In addition, unlike previous algorithms, its space complexity scales like $O(Kd^2)$ and does not grow with T.

Keywords: Contextual Linear Bandits, Space and Time Efficiency

1 Introduction

The contextual multi-armed bandit problem is a sequential learning problem [17,13]. At each time step, a learner has to chose among a set of possible actions/arms $\mathcal{A}$. Prior to making its decision, the learner observes some additional side information $x \in \mathcal{X}$ over which he has no influence. This is commonly referred to as the *context*. In general, the reward of a particular arm $a \in \mathcal{A}$ under context $x \in \mathcal{X}$ follows some unknown distribution. The goal of the learner is to select arms so that it minimizes its expected *regret*, *i.e.*, the expected difference between its cumulative reward and the reward accrued by an optimal policy, that knows the reward distributions.

Langford and Zhang [17] propose an algorithm called *epoch-Greedy* for general contextual bandits. Their algorithm achieves an $O(\log T)$ regret in the number of timesteps T in the *stochastic* setting, in which contexts are sampled from an unknown distribution in an i.i.d. fashion. Unfortunately, the proposed algorithm and subsequent improvements [13] have high computational complexity. Selecting an arm at time step t requires making a number of calls to a so-called *optimization oracle* that grows polynomially in T. In addition, the cost of an implementation of this optimization oracle can grow linearly in $|\mathcal{X}|$ in the worst case; this is prohibitive in many interesting cases, including the case where $|\mathcal{X}|$ is exponential in the dimension of the context. In addition, both algorithms proposed in [17] and [13] require keeping a history of observed contexts and arms chosen at every time instant. Hence, their space complexity grows linearly in T.

In this paper, we show that the challenges above can be addressed when rewards are linear. In the above contextual bandit set up, this means that $\mathcal{X}$ is a subset of $\mathbb{R}^d$, and the expected reward of an arm $a \in \mathcal{A}$ is an unknown linear function of the context x, *i.e.*, it has the form $x^\dagger \theta_a$, for some unknown vector θ_a. This is a case of great interest, arising naturally when, conditioned on x, rewards from different arms are uncorrelated:

Example 1. *(Processor Scheduling)* A simple example is assigning incoming jobs to a set of processors $\mathcal{A}$, whose processing capabilities are not known *a priori*. This could be the case if, *e.g.*, the processors are machines in the cloud or, alternatively, humans offering their services through, *e.g.*, Mechanical Turk. Each arriving job is described by a set of attributes $x \in \mathbb{R}^d$, each capturing the work load of different types of sub-tasks this job entails, *e.g.*, computation, I/O, network communication, *etc.* Each processor's unknown feature vector θ_a describes its processing capacity, *i.e.*, the time to complete a sub-task unit, in expectation. The expected time to complete a task x is given by $x^\dagger \theta_a$; the goal of minimizing the delay (or, equivalently, maximizing its negation) brings us in the contextual bandit setting with linear rewards. □

Example 2. *(Display Ad Placement)* In the online ad placement problem, online users are visiting a website, which must decide which ad to show them selected from a set $\mathcal{A}$. Each online user visiting the website is described by a set of attributes $x \in \mathbb{R}^d$ capturing, *e.g.*, its geo-location, its previous viewing history, or any information available through a tracking service like BlueKai. Each ad $a \in \mathcal{A}$ has a probability of being clicked that is of the form $x^\dagger \theta_a$, where $\theta_a \in \mathbb{R}^d$ an unknown vector describing each ad. The system objective is to maximize the number of clicks, falling again under the above contextual bandit setting. □

Example 3. *(Group Activity Selection)* Another motivating example is maximizing group satisfaction, observed as the outcome of a secret ballot election. In this setup, a subset of d users congregate to perform a joint activity, such as, *e.g.*, dining, rock climbing, watching a movie, *etc.* The group is dynamic and, at each time step, the vector $x \in \{0,1\}^d$, is an indicator of present participants. An arm (*i.e.*, a joint activity) is selected; at the end of the activity, each user votes whether they liked the activity or not in a secret ballot, and the final tally is disclosed. In this scenario, the unknown vectors $\theta_a \in \mathbb{R}^d$ indicate the probability a given participant will enjoy activity a, and the goal is to select activities that maximize the aggregate satisfaction among participants present at the given time step. □

Our contributions are as follows.

- We isolate and focus on linear payoff case of stochastic multi-armed bandit problems, and design a simple arm selection policy which does not recourse to sophisticated oracles inherent in prior work.
- We prove that our policy achieves an $O(\log T)$ regret after T steps in the stochastic setting, when the expected rewards of each arm are well separated. This meets the regret bound of best known algorithms for contextual multi-armed bandit problems. In addition, for many natural scenarios, it scales as $O(poly(d) \log T)$, which we believe we are the first to prove under arm separation and for an efficient algorithm.
- We show that our algorithm has $O(|\mathcal{A}|d^3)$ computational complexity per step and its expected space complexity scales like $O(|\mathcal{A}|d^2)$. For algorithms that achieve

similar regrets, this is a significant improvement over known contextual multi-armed bandit problems, as well as for bandits specialized for linear payoffs.

Our algorithm is inspired by the work of [3] on the ϵ-greedy algorithm and the use of linear regression to estimate the parameters θ_a. The main technical innovation is the use of matrix concentration bounds to control the error of the estimates of θ_a in the stochastic setting. We believe that this is a powerful realization and may ultimately help us analyze richer classes of payoff functions.

The remainder of this paper is organized as follows: in Section 2 we compare our results with existing literature. In Section 3 we describe the set up of our problem in more detail. In Section 4 we state our main results and prove them in Section 5. Section 6 is devoted to exemplifying the performance and limitations of our algorithm by means of simple numerical simulations. We discuss challenges in dealing with an adversarial setting in Section 7 and draw our conclusions in Section 8.

2 Related Work

The original paper by Langford and Zhang [17] assumes that the context $x \in \mathcal{X}$ is sampled from a probability distribution $p(x)$ and that, given an arm $a \in \mathcal{A}$, and conditioned on the context x, rewards r are sampled from a probability distribution $p_a(r \mid x)$. As is common in bandit problems, there is a tradeoff between *exploration*, *i.e.*, selecting arms to sample rewards from the distributions $\{p_a(r \mid x)\}_{a \in \mathcal{A}}$ and learn about them, and *exploitation*, whereby knowledge of these distributions based on the samples is used to select an arm that yields a high payoff.

In this setup, a significant challenge is that, though contexts x are sampled independently, they are not independent conditioned on the arm played: an arm will tend to be selected more often in contexts in which it performs well. Hence, learning the distributions $\{p_a(r \mid x)\}_{a \in \mathcal{A}}$ from such samples is difficult. The epoch-Greedy algorithm [17] deals with this by separating the exploration and exploitation phase, effectively selecting an arm uniformly at random at certain time slots (the exploration "epochs"), and using samples collected only during these epochs to estimate the payoff of each arm in the remaining time slots (for exploitation). Our algorithm uses the same separation in "epochs". Langford and Zhang [17] establish an $O(T^{2/3}(\ln |\mathcal{X}|)^{1/3})$ bound on the regret for epoch-Greedy in their stochastic setting. They further improve this to $O(\log T)$ when a lower bound on the gap between optimal and suboptimal arms in each context exists, *i.e.*, under *arm separation*.

Unfortunately, the price of the generality of the framework in [17] is the high computational complexity when selecting an arm during an exploitation phase. In a recent improvement [13], this computation requires a $poly(T)$ number of calls to an optimization oracle. Most importantly, even in the linear case we study here, there is no clear way to implement this oracle in sub-exponential time in d, the dimension of the context. As Dudik *et al.* [13] point out, the optimization oracle solves a so-called cost-sensitive classification problem. In the particular case of linear bandits, the oracle thus reduces to finding the "least-costly" linear classifier. This is hard, even in the case of only two arms: finding the linear classifier with the minimal number of errors is NP-hard [15], and remains NP hard even if an approximate solution is required [7]. As such, a different approach is warranted under linear rewards.

Contextual bandits with linear rewards is a special case of the classic linear bandit setup [4,9,18,19]. In this setup, the arms themselves are represented as vectors, *i.e.*, $\mathcal{A} \subset \mathbb{R}^d$, and, in addition, the set $\mathcal{A}$ can change from one time slot to the next. The expected payoff of an arm a with vector x_a is given by $x_a^\dagger \theta$, for some unknown vector $\theta \in \mathbb{R}^d$, *common among all arms*.

There are several different variants of the above linear model. Auer [4], Li *et al.* [18], and Chu *et al.* [9], and Li a study this problem in the adversarial setting, assuming a finite number of arms $|A|$. In the adversarial setting, contexts are not sampled i.i.d. from a distribution but can be an arbitrary sequence, for example, chosen by an adversary that has knowledge of the algorithm and its state variables.Both algorithms studied, LinRel and LinUCB, are similar to ours in that they use an upper confidence bound and both estimate the unknown parameters for the linear model using a least-square-error type method. In addition, both methods apply some sort of regularization. LinRel does it by truncating the eigenvalues of a certain matrix and LinUCB by using ridge regression. In the adversarial setting, and with no arm separation, the regret bounds obtained of the form $O(\sqrt{T} polylog(T))$.

Dani *et al.* [12], Rusmevichientong and Tsitsiklis [19], and Abbasi-Yadkori *et al.* [1] study contextual linear bandits in the stochastic setting, in the case where $\mathcal{A}$ is a fixed but possibly uncountable bounded subset of $\mathbb{R}^d$. Dani *et al.* [12] obtain regret bounds of $O(\sqrt{T})$ for an infinite number of arms; under arm separation, by introducing a gap constant Δ, their bound is $O(d^2 (\log T)^3)$. Rusmevichientong and Tsitsiklis [19] also study the regret under arm separation and obtain a $O(\log(T))$ bound that depends exponentially on d. Finally, Abbasi-Yadkori *et al.* [1] obtain a $O(poly(d) \log^2(T))$ bound under arm separation.

Our problem can be expressed as a special case of the linear bandits setup by taking $\theta = [\theta_1; \ldots; \theta_K] \in \mathbb{R}^{Kd}$, where $K = |\mathcal{A}|$, and, given context x, associating the i-th arm with an appropriate vector of the form $x_{a_i} = [0 \ldots x \ldots 0]$. As such, all of the bounds described above [4,18,9,12,19,1] can be applied to our setup. However, in our setting, arms are uncorrelated; the above algorithms do not exploit this fact. Our algorithm indeed exploits this to obtain a *logarithmic* regret, while also scaling well in terms of the dimension d.

Several papers study contextual linear bandits under different notions of regret. For example, Dani *et al.* [11] define regret based on the worst sequence of loss vectors. In our setup, this corresponds to the rewards coming from an arbitrary temporal sequence and not from adding noise to $x^\dagger \theta_a$, resembling the 'worst-case' regret definition of [5]. Abernethy *et al.* [2] assume a notion of regret with respect to a best choice fixed in time that the player can make from a fixed set of choices. However, in our case, the best choice changes with time t via the current context. This different setup yields worse bounds than the ones we seek: for both stochastic and adversarial setting the regret is $O(\sqrt{T} polylog(T))$.

Recent studies on multi-class prediction using bandits [16,14,10] have some connections to our work. In this setting, every context x has an associated label y that a learner tries to predict using a linear classifier of the type $\hat{y} = \arg\max_a \theta_a^\dagger x$. Among algorithms proposed, the closest to ours is by Crammer and Gentile [10], which uses an estimator for $\{\theta_a\}$ that is related to LinUCB, LinRel and our algorithm. However, the multi-class prediction problem differs in many ways from our setting. To learn the vectors θ_a, the learner receives a one-bit feedback indicating whether the label predicted is correct (*i.e.*, the arm was maximal) or not. In contrast, in our setting, the learner directly observes $\theta_a^\dagger x$, possibly perturbed by noise, without learning if it is maximal.

Finally, bandit algorithms relying on experts such as EXP4 [6] and EXP4.P [8] can also be applied to our setting. These algorithms require a set of policies (experts) against which the regret is measured. Regret bounds grow as $\log^C N$, where N is the number of experts and C a constant. The trivial reduction of our problem to EXP4(.P) assigns an expert to each possible context-to-arm mapping. The 2^d contexts in our case lead to K^{2^d} experts, an undesirable exponential growth of regret in d; a better choice of experts is a new problem in itself.

3 Model

In this section, we give a precise definition of our linear contextual bandit problem.

Contexts. At every time instant $t \in \{1, 2, ...\}$, a context $x_t \in \mathcal{X} \subset \mathbb{R}^d$, is observed by the learner. We assume that $\|x\|_2 \leq 1$; as the expected reward is linear in x, this assumption is without loss of generality (w.l.o.g.). We prove our main result (Theorem 2) in the stochastic setting where x_t are drawn i.i.d. from an unknown multivariate probability distribution $\mathcal{D}$. In addition, we require that the set of contexts is finite *i.e.*, $|\mathcal{X}| < \infty$. We define $\Sigma_{\min} > 0$ to be the smallest non-zero eigenvalue of the covariance matrix $\Sigma \equiv \mathbb{E}\{x_1 x_1^\dagger\}$.

Arms and Actions. At time t, after observing the context x_t, the learner decides to play an arm $a \in \mathcal{A}$, where $K \equiv |\mathcal{A}|$ is finite. We denote the arm played at this time by a_t. We study *adaptive* arm selection policies, whereby the selection of a_t depends only on the current context x_t, and on all past contexts, actions and rewards. In other words, $a_t = a_t\left(x_t, \{x_\tau, a_\tau, r_\tau\}_{\tau=1}^{t-1}\right)$.

Payoff. After observing a context x_t and selecting an arm a_t, the learner receives a payoff r_{a_t,x_t} which is drawn from a distribution p_{a_t,x_t} independently of all past contexts, actions or payoffs. We assume that the expected payoff is a linear function of the context. In other words,

$$r_{a_t,x_t} = x_t^\dagger \theta_a + \epsilon_{a,t} \qquad (1)$$

where $\{\epsilon_{a,t}\}_{a\in\mathcal{A},t\geq 1}$ are a set of independent random variables with zero mean and $\{\theta_a\}_{a\in\mathcal{A}}$ are unknown parameters in $\mathbb{R}^d$. Note that, w.l.o.g, we can assume that $Q = \max_{a\in\mathcal{A}} \|\theta_a\|_2 \leq 1$. This is because if $Q > 1$, as payoffs are linear, we can divide all payoffs by Q; the resulting payoff is still a linear model, and our results stated below apply. Recall that Z is a sub-gaussian random variable with constant L if $\mathbb{E}\{e^{\gamma Z}\} \leq e^{\gamma^2 L^2}$. In particular, sub-gaussianity implies $\mathbb{E}\{Z\} = 0$. We make the following technical assumption.

Assumption 1 *The random variables $\{\epsilon_{a,t}\}_{a\in\mathcal{A},t\geq 1}$ are sub-gaussian random variables with constant $L > 0$.*

Regret. Given a context x, the optimal arm is $a^*_x = \arg\max_{a\in\mathcal{A}} x^\dagger\theta_a$. The expected cumulative regret the learner experiences over T steps is defined by

$$R(T) = \mathbb{E}\Big\{\sum_{t=1}^{T} x_t^\dagger(\theta_{a^*_{x_t}} - \theta_{a_t})\Big\}. \tag{2}$$

The expectation above is taken over the contexts x_t. The objective of the learner is to design a policy $a_t = a_t\left(x_t, \{x_\tau, a_\tau, r_\tau\}_{\tau=1}^{t-1}\right)$ that achieves as low expected cumulative regret as possible. In this paper we are also interested in arm selection policies having a low computational complexity. We define $\varDelta_{\max} \equiv \max_{a,b\in\mathcal{A}} ||\theta_a - \theta_b||_2$, and $\varDelta_{\min} \equiv \inf_{x\in\mathcal{X},a:x^\dagger\theta_a<x^\dagger\theta_{a^*_x}} x^\dagger(\theta_{a^*_x} - \theta_a) > 0$. Observe that, by the finiteness of $\mathcal{X}$ and $\mathcal{A}$, the defined infimum is attained (*i.e.*, it is a minimum) and is indeed positive.

4 Main Results

We now present a simple and *efficient* on-line algorithm that, under the above assumptions, has expected *logarithmic* regret. Specifically, its computational complexity, at each time instant, is $O(Kd^3)$ and the expected memory requirement scales like $O(Kd^2)$. As far as we know, our analysis is the first to show that a simple and *efficient* algorithm for the problem of linearly parametrized bandits can, under reward separation and i.i.d. contexts, achieve logarithmic expected cumulative regret that simultaneously can scale like $polylog(|\mathcal{X}|)$ for natural scenarios.

Before we present our algorithm in full detail, let us give some intuition about it. Part of the job of the learner is to estimate the unknown parameters θ_a based on past actions, contexts and rewards. We denote the estimate of θ_a at time t by $\hat{\theta}_a$. If $\theta_a \approx \hat{\theta}_a$ then, given an observed context, the learner will more accurately know which arm to play to incur in small regret. The estimates $\hat{\theta}_a$ can be constructed based on a history of past rewards, contexts and arms played. Since observing a reward r for arm a under context x does not give information about the magnitude of θ_a along directions orthogonal to x, it is important that, for each arm, rewards are observed and recorded for a rich class of contexts. This gives rise to the following challenge: If the learner tries to build this history while trying to minimize the regret, the distribution of contexts observed when playing a certain arm a will be biased and potentially not rich enough. In particular, when trying to achieve a small regret, conditioned on $a_t = a$, it is more likely that x_t is a context for which a is optimal.

We address this challenge using the following idea, also appearing in the epoch-Greedy algorithm of [17]. We partition time slots into *exploration* and *exploitation epochs*. In exploration epochs, the learner plays arms uniformly at random, independently of the context, and records the observed rewards. This guarantees that in the history of past events, each arm has been played along with a sufficiently rich set of contexts. In exploitation epochs, the learner makes use of the history of events stored during exploration to estimate the parameters θ_a and determine which arm to play given a current observed context. The rewards observed during exploitation are not recorded.

Algorithm 1 Contextual ϵ -greedy

```
For all $a \in A$, set $A_a \leftarrow 0_{d\times d}$ ;$n_a \leftarrow 0$; $b_a \leftarrow 0_d$
for $t = 1$ to $p$ do
  $a \leftarrow 1 + (t \mod K)$; Play arm $a$
  $n_a \leftarrow n_a + 1$; $b_a \leftarrow b_a + r_t x_t$; $A_a \leftarrow A_a + x_t x_t^\dagger$
end for
for $t = p + 1$ to $T$ do
  $e \leftarrow$ Bernoulli$(p/t)$
  if $e = 1$ then
    $a \leftarrow$ Uniform$(1/K)$ ; Play arm $a$
    $n_a \leftarrow n_a + 1$; $b_a \leftarrow b_a + r_t x_t$; $A_a \leftarrow A_a + x_t x_t^\dagger$
  else
    for $a \in \mathcal{A}$ do
      Get $\hat\theta_a$ as the solution to the linear system: $\left(\lambda_{n_a} I + \frac{1}{n_a} A_a\right)\hat\theta_a = \frac{1}{n_a} b_a$
    end for
    Play arm $a_t = \arg\max_{a\in\mathcal{A}} x_t^\dagger \hat\theta_a$
  end if
end for
```

More specifically, when exploiting, the learner performs two operations. In the first operation, for each arm $a \in \mathcal{A}$, an estimate $\hat\theta_a$ of θ_a is constructed from a simple ℓ_2-regularized regression, as in in [4] and [9]. In the second operation, the learner plays the arm a that maximizes $x_t^\dagger \hat\theta_a$. Crucially, in the first operation, only information collected during exploration epochs is used. In particular, let $\mathcal{T}_{a,t-1}$ be the set of exploration epochs up to and including time $t-1$ (*i.e.*, the times that the learner played an arm a uniformly at random (u.a.r.)). Moreover, for any $\mathcal{T} \subset \mathbb{N}$, denote by $r_\mathcal{T} \in \mathbb{R}^n$ the vector of observed rewards for all time instances $t \in \mathcal{T}$, and $X_\mathcal{T} \in \mathbb{R}^{n\times d}$ is a matrix of $\mathcal{T}$ rows, each containing one of the observed contexts at time $t \in \mathcal{T}$. Then, at time t the estimator $\hat\theta_a$ is the solution of the following convex optimization problem.

$$\min_{\theta\in\mathbb{R}^d} \frac{1}{2n}\|r_\mathcal{T} - X_\mathcal{T}\theta\|_2^2 + \frac{\lambda_n}{2}\|\theta\|_2^2. \tag{3}$$

where $\mathcal{T} = \mathcal{T}_{a,t-1}$, $n = |\mathcal{T}_{a,t-1}|$, $\lambda_n = 1/\sqrt{n}$. In other words, the estimator $\hat\theta_a$ is a (regularized) estimate of θ_a, based only on observations made during exploration epochs. Note that the solution to (3) is given by $\hat\theta_a = \left(\lambda_n I + \frac{1}{n} X_\mathcal{T}^\dagger X_\mathcal{T}\right)^{-1} \frac{1}{n} X_\mathcal{T}^\dagger r_\mathcal{T}$.

An important design choice is the above process selection of the time slots at which the algorithm explores, rather than exploits. Following the ideas of [20], we select the exploration epochs so that they occur approximately $\Theta(\log t)$ times after t slots. This guarantees that, at each time step, there is enough information in our history of past events to determine the parameters accurately while only incurring in a regret of $O(\log t)$. There are several ways of achieving this; our algorithm explores at each time step with probability $\Theta(t^{-1})$.

The above steps are summarized in pseudocode by Algorithm 1. Note that the algorithm contains a scaling parameter p, which is specified below, in Theorem 2. Because there are K arms and for each arm $(x_t, r_{a,t}) \in \mathbb{R}^{d+1}$, the expected memory required by the algorithm scales like $O(Kd^2)$. In addition, both the matrix $X^{\dagger}_{\mathcal{T}}X_{\mathcal{T}}$ and the vector $X^{\dagger}_{\mathcal{T}}r_{\mathcal{T}}$ can be computed in an online fashion in $O(d^2)$ time: $X^{\dagger}_{\mathcal{T}}X_{\mathcal{T}} \leftarrow X^{\dagger}_{\mathcal{T}}X_{\mathcal{T}} + x_t x_t^{\dagger}$ and $X^{\dagger}_{\mathcal{T}}r_{\mathcal{T}} \leftarrow X^{\dagger}_{\mathcal{T}}r_{\mathcal{T}} + r_t x_t$. Finally, the estimate of $\hat{\theta}_a$ requires solving a linear system (see Algorithm 1), which can be done in $O(d^3)$ time. The above is summarized in the following theorem.

Theorem 1. *Algorithm 1 has computational complexity of $O(Kd^3)$ per iteration and its expected space complexity scales like $O(Kd^2)$.*

We now state our main theorem that shows that Algorithm 1 achieves $R(T) = O(\log T)$.

Theorem 2. *Under Assumptions 1, the expected cumulative regret of algorithm 1 satisfies,*

$$R(T) \leq p\Delta_{\max}\sqrt{d} + 14\Delta_{\max}\sqrt{d}Ke^{Q/4} + p\Delta_{\max}\sqrt{d}\log T.$$

for any

$$p \geq \frac{CKL'^2}{(\Delta'_{\min})^2(\Sigma'_{\min})^2}. \tag{4}$$

Above, C is a universal constant, $\Delta'_{\min} = \min\{1, \Delta_{\min}\}$, $\Sigma'_{\min} = \min\{1, \Sigma_{\min}\}$ and $L' = \max\{1, L\}$.

Algorithm 1 requires the specification of the constant p. In Section 4.2, we give two examples of how to efficiently choose a p that satisfies (4). In Theorem 2, the bound on the regret depends on p - small p is preferred - and hence it is important to understand how the right hand side (r.h.s.) of (4) might scale when K and d grow. In Section 4.1, we show that, for a concrete distribution of contexts and choice of expected rewards θ_a, and assuming (4) holds, $p = O(K^3d^5)$ [1] . There is nothing special about the concrete details of how contexts and θ_a's are chosen and, although not included in this paper, for many other distributions, one also obtains $p = O(poly(d))$. We can certainly construct pathological cases where, for example, p grows exponentially with d. However, we do not find these intuitive. Specially when interpreting these having in mind real applications as the ones introduced in Examples 1- 3.

4.1 Example of Scaling of p with d and K

Assume that contexts are obtained by normalizing a d-dimensional vector with i.i.d. entries as Bernoulli random variables with parameter w. Assume in addition that every θ_a is obtained i.i.d. from the following prior distribution: every entry of θ_a is drawn i.i.d. from a uniform distribution and then θ_a is normalized. Finally, assume that the payoffs are given by $r_{a,t} = x_t^{\dagger}\Theta_a$, where $\Theta_a \in \mathbb{R}^d$ are random variables that fluctuate around $\theta_a = \mathbb{E}\{\Theta_a\}$ with each entry fluctuating by at most F.

Under these assumptions the following is true:

- $\Sigma_{\min} = \Omega(d^{-1})$. In fact, the same result holds asymptotically independently of $w = w(d)$ if, for example, we assume that on average groups are roughly of the same size, M, with $w = M/d$;

[1] This bound holds with probability converging to 1 as K and d get large

- $L = O(\sqrt{d})$. This holds because $\epsilon_{a,t} = r_{a,t} - \mathbb{E}\{r_{a,t}\} = x_t^\dagger(\Theta_a - \theta_a)$ are bounded random variables with zero mean and $\|x_t^\dagger(\Theta_a - \theta_a)\}\|_\infty = O(\sqrt{d})$.
- $\Delta_{\min} = \Omega(1/(Kd\sqrt{w})$ with high-probability (for large K and d). This can be seen as follows, if $\Delta_{\min} = x^\dagger(\theta_a - \theta_b)$ for some x, a and b, then it must be true that θ_a and θ_b differ in a component for which x is non-zero. The minimum difference between components among all pairs of θ_a and θ_b is lower bounded by $\Omega(1/(K\sqrt{d}))$ with high probability (for large K and d). Taking into account that each entry of x is $O(1/\sqrt{dw})$ with high-probability, the bound on $\Delta_{\min}$ follows.

If we want to apply Theorem 2 then (4) must hold and hence putting all the above calculations together we conclude that $p = O(K^3 d^5)$ with high probability for large K and p.

4.2 Computing *p* in Practice

If we have knowledge of an a priori distribution for the contexts, for the expected payoffs and for the variance of the rewards then we can quickly compute the value of $\Sigma_{\min}$, L and a typical value for $\Delta_{\min}$. An example of this was done above (Section 4.1). There, the values were presented only in order notation but exact values are not hard to obtain for that and other distributions. Since a suitable p only needs to be larger then the r.h.s. of (4), by introducing an appropriate multiplicative constant, we can produce a p that satisfied (4) with high probability.

If we have no knowledge of any model for the contexts or expected payoffs, it is still possible to find p by estimating $\Delta_{\min}$, $\Sigma_{\min}$ and L from data gathered while running Algorithm 1. Notice again that, since all that is required for our theorem to hold is that p is greater then a certain function of these quantities, an exact estimation is not necessary. This is important because, for example, accurately estimating $\Sigma_{\min}$ is hard when matrix $\mathbb{E}\{x_1 x_1^\dagger\}$ has a large condition number.

Not being too concerned about accuracy, $\Sigma_{\min}$ can be estimated from $\mathbb{E}\{x_1 x_1^\dagger\}$, which can be estimated from the sequence of observed x_t. $\Delta_{\min}$ can be estimated from Algorithm 1 by keeping track of the smallest difference observed until time t between $\max_b x^\dagger \hat{\theta}_b$ and the second largest value of the function being maximized. Finally, the constant L can be estimated from the variance of the observed rewards for the same (or similar) contexts. Together, these estimations do not incur in any significant loss in computational performance of our algorithm.

5 Proof of Theorem 2

The general structure of the proof of our main result follows that of [3]. The main technical innovation is the realization that, in the setting when the contexts are drawn i.i.d. from some distribution, a standard matrix concentration bound allows us to treat $\lambda_n I + n^{-1}(X_\mathcal{T}^\dagger X_\mathcal{T})$ in Algorithm 1 as a deterministic positive-definite symmetric matrix, even as $\lambda_n \to 0$.

Let $\mathcal{E}_T$ denote the time instances for $t > p$ and until time T in which the algorithm took an exploitation decision. Recall that, by Cauchy-Schwarz inequality, $x_t^\dagger(\theta_{a^*_{x_t}} - \theta_a) \leq \|x_t\|_1 \|(\theta_{a^*_{x_t}} - \theta_a)\|_\infty \leq \sqrt{d}\|x_t\|_2 \|(\theta_{a^*_{x_t}} - \theta_a)\|_\infty \leq \sqrt{d}\Delta_{\max}$. In addition, recall that $\sum_{t=2}^{T} 1/t \leq \log T$. For $R(T)$ the cumulative regret until time T, we can write

$$\begin{aligned}R(T) = \mathbb{E}\{\sum_{t=1}^{T} x_t^\dagger(\theta_{a^*_{x_t}} - \theta_a)\} &\leq p\Delta_{\max}\sqrt{d} + \Delta_{\max}\sqrt{d}\mathbb{E}\{\sum_{t=p+1}^{T} \mathbb{1}\{x_t^\dagger\theta_a < x_t^\dagger\theta_{a^*_{x_t}}\}\} \\ &\leq p\Delta_{\max}\sqrt{d} + \Delta_{\max}\sqrt{d}\mathbb{E}\{|\mathcal{E}_T|\} + \Delta_{\max}\sqrt{d}\mathbb{E}\{\sum_{t\in\mathcal{E}_T} \mathbb{1}\{x_t^\dagger\theta_a < x_t^\dagger\theta_{a^*_{x_t}}\}\} \\ &\leq p\Delta_{\max}\sqrt{d} + p\Delta_{\max}\sqrt{d}\log T + \Delta_{\max}\sqrt{d}\mathbb{E}\{\sum_{t\in\mathcal{E}_T} \mathbb{1}\{x_t^\dagger\theta_a < x_t^\dagger\theta_{a^*_{x_t}}\}\} \\ &\leq p\Delta_{\max}\sqrt{d} + p\Delta_{\max}\sqrt{d}\log T + \Delta_{\max}\sqrt{d}\mathbb{E}\{\sum_{t\in\mathcal{E}_T}\sum_{a\in\mathcal{A}} \mathbb{1}\{x_t^\dagger\hat{\theta}_a > x_t^\dagger\hat{\theta}_{a^*_{x_t}}\}\}.\end{aligned}$$

In the last line we used the fact that when exploiting, if we do not exploit the optimal arm $a^*_{x_t}$, then it must be the case that the estimated reward for some arm a, $x_t^\dagger\hat{\theta}_a$, must exceed that of the optimal arm, $x_t^\dagger\hat{\theta}_{a^*_{x_t}}$, for the current context x_t.

We can continue the chain of inequalities and write,

$$R(T) \leq p\Delta_{\max}\sqrt{d} + p\Delta_{\max}\sqrt{d}\log T + \Delta_{\max}\sqrt{d}K\sum_{t=1}^{T}\mathbb{P}\{x_t^\dagger\hat{\theta}_a > x_t^\dagger\hat{\theta}_{a^*_{x_t}}\}.$$

The above expression depends on the value of the estimators for time instances that might or might not be exploitation times. For each arm, these are computed just like in Algorithm 1, using the most recent history available. The above probability depends on the randomness of x_t and on the randomness of recorded history for each arm.

Since $x_t^\dagger(\theta_{a^*_{x_t}} - \theta_a) \geq \Delta_{\min}$ we can write

$$\mathbb{P}\{x_t^\dagger\hat{\theta}_a > x_t^\dagger\hat{\theta}_{a^*_{x_t}}\} \leq \mathbb{P}\Big\{x_t^\dagger\hat{\theta}_a \geq x_t^\dagger\theta_a + \frac{\Delta_{\min}}{2}\Big\} + \mathbb{P}\Big\{x_t^\dagger\hat{\theta}_{a^*_{x_t}} \leq x_t^\dagger\theta_{a^*_{x_t}} - \frac{\Delta_{\min}}{2}\Big\}.$$

We now bound each of these probabilities separately. Since their bound is the same, we focus only on the first probability.

Substituting the definition of $r_a(t) = x_t^\dagger\theta_a + \epsilon_{a,t}$ into the expression for $\hat{\theta}_a$ one readily obtains,

$$(\hat{\theta}_a - \theta_a) = \left(\lambda_n I + \frac{1}{n}X_{\mathcal{T}}^\dagger X_{\mathcal{T}}\right)^{-1}\left(\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_\tau\epsilon_{a,\tau} - \lambda_n\theta_a\right).$$

We are using again the notation $\mathcal{T} = \mathcal{T}_{a,t-1}$ and $n = |\mathcal{T}|$. From this expression, an application of Cauchy-Schwarz's inequality and the triangular inequality leads to,

$$\begin{aligned}|x_t^\dagger(\hat{\theta}_a - \theta_a)| &= \Big|x_t^\dagger\left(\lambda_n I + \frac{1}{n}X_{\mathcal{T}}^\dagger X_{\mathcal{T}}\right)^{-1}\left(\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_\tau\epsilon_{a,\tau} - \lambda_n\theta_a\right)\Big| \\ &\leq \sqrt{x_t^\dagger\left(\lambda_n I + \frac{1}{n}X_{\mathcal{T}}^\dagger X_{\mathcal{T}}\right)^{-2} x_t}\left(\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^\dagger x_\tau\epsilon_{a,\tau}\Big| + \lambda_n|x_t^\dagger\theta_a|\right).\end{aligned}$$

We introduce the following notation

$$c_{a,t} \equiv \sqrt{x_t^\dagger \left(\lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}}\right)^{-2} x_t}. \tag{5}$$

Note that, given a and t both n and $\mathcal{T}$ are well specified.

We can now write,

$$\mathbb{P}\Big\{x_t^\dagger \hat{\theta}_a \geq x_t^\dagger \theta_a + \frac{\varDelta_{\min}}{2}\Big\} \leq \mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau}\Big| \geq \frac{\varDelta_{\min}}{2c_{a,t}} - \lambda_n |x_t^\dagger \theta_a|\Big\}$$
$$\leq \mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau}\Big| \geq \frac{\varDelta_{\min}}{2c_{a,t}} - \lambda_n Q\Big\}.$$

Since $\epsilon_{a,\tau}$ are sub-gaussian random variables with sub-gaussian constant upper bounded by L and since $|x_t^\dagger x_\tau| \leq 1$, conditioned on x_t, $\mathcal{T}$ and $\{x_\tau\}_{\tau\in\mathcal{T}}$, each $x_t^\dagger x_\tau \epsilon_{a,\tau}$ is a sub-gaussian random variable and together they form a set of i.i.d. sub-gaussian random variables. One can thus apply standard concentration inequality and obtain,

$$\mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau}\Big| \geq \frac{\varDelta_{\min}}{2c_{a,t}} - \lambda_n Q\Big\} \leq \mathbb{E}\Big\{2e^{-\frac{n}{2L^2}\left(\frac{\varDelta_{\min}}{2c_{a,t}} - \lambda_n Q\right)^{+2}}\Big\}. \tag{6}$$

where both n and $c_{a,t}$ are random quantities and $z^+ = z$ if $z \geq 0$ and zero otherwise.

We now upper bound $c_{a,t}$ using the following fact about the eigenvalues of any two real-symmetric matrices M_1 and M_2: $\lambda_{\max}(M_1^{-1}) = 1/\lambda_{\min}(M_1)$ and $\lambda_{\min}(M_1 + M_2) \geq \lambda_{\min}(M_1) - \lambda_{\max}(M_2) = \lambda_{\min}(M_1) - \|M_2\|$.

$$c_{a,t} \leq \left(\lambda_n + \lambda_{\min}^+(\mathbb{E}\{x_1^\dagger x_1\}) - \Big\|\frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} - \mathbb{E}\{x_1^\dagger x_1\}\Big\|^+\right)^{-1}.$$

Both the eigenvalue and the norm above only need to be computed over the subspace spanned by the vectors x_t that occur with non-zero probability. We use the symbol $^+$ to denote the restriction to this subspace. Now notice that $\|.\|^+ \leq \|.\|$ and, since we defined $\varSigma_{\min} \equiv \min_{i:\lambda_i>0} \lambda_i(\mathbb{E}\{X_1 X_1^\dagger\})$, we have that $\lambda_{\min}^+(\mathbb{E}\{X_1 X_1^\dagger\}) \geq \varSigma_{\min}$. Using the following definition, $\varDelta\varSigma_n \equiv n^{-1} X_{\mathcal{T}}^\dagger X_{\mathcal{T}}^\dagger - \mathbb{E}\{X_1 X_1^\dagger\}$, this leads to, $c_{a,t} \leq (\lambda_n + \varSigma_{\min} - \|\varDelta\varSigma_n\|)^{-1} \leq (\varSigma_{\min} - \|\varDelta\varSigma_n\|)^{-1}$.

We now need the following Lemma.

Lemma 1. *Let $\{X_i\}_{i=1}^n$ be a sequence of i.i.d. random vectors of 2-norm bounded by 1. Define $\hat{\varSigma} = \frac{1}{n}\sum_{i=1}^n X_i X_i^\dagger$ and $\varSigma = \mathbb{E}\{X_1 X_1^\dagger\}$. If $\epsilon \in (0,1)$ then,*

$$\mathbb{P}(|\,\hat{\varSigma} - \varSigma\| > \epsilon\|\varSigma\|) \leq 2e^{-C\epsilon^2 n},$$

where $C < 1$ is an absolute constant.

For a proof see [21] (Corollary 50).

We want to apply this lemma to produce a useful bound on the r.h.s. of (6). First notice that, conditioning on n, the expression inside the expectation in (6) depends through $c_{a,t}$ on n i.i.d. contexts that are distributed according to the original distribution. Because of this, we can write,

$$\begin{aligned}&\mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^\dagger x_\tau \epsilon_{a,\tau}\Big| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q\Big\} \leq \mathbb{E}\Big\{2e^{-\frac{n}{2L^2}\left(\frac{\Delta_{\min}}{2c_{a,t}}-\lambda_n Q\right)^{+2}}\Big\}\\ &\leq \sum_{n=1}^{t}\Big(\mathbb{P}\{|\mathcal{T}_{a,t-1}|=n\}\times\mathbb{E}\Big\{2e^{-\frac{n}{2L^2}\left(\frac{\Delta_{\min}}{2c_{a,t}}-\lambda_n Q\right)^{+2}}\Big||\mathcal{T}_{a,t-1}|=n\Big\}\Big).\end{aligned}$$

Using the following algebraic relation: if $z, w > 0$ then $(z-w)^{+2} \geq z^2 - 2zw$, we can now write,

$$\begin{aligned}&\mathbb{E}\Big\{e^{-\frac{n}{2L^2}\left(\frac{\Delta_{\min}}{2c_{a,t}}-\lambda_n Q\right)^{+2}}\Big||\mathcal{T}_{a,t-1}|=n\Big\}\\ &\leq \mathbb{P}\{|\Delta\Sigma_n| > \Sigma_{\min}/2|\ |\mathcal{T}_{a,t-1}|=n\} + e^{-\frac{n}{2L^2}\left(\frac{\Sigma_{\min}\Delta_{\min}}{4}-\lambda_n Q\right)^{+2}}\\ &\leq \mathbb{P}\{|\Delta\Sigma_n| > \Sigma_{\min}/2|\ |\mathcal{T}_{a,t-1}|=n\} + e^{\frac{Q\Delta_{\min}\Sigma_{\min}}{4L^2}}e^{-\frac{n(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}}\end{aligned}$$

Using Lemma 1 we can continue the chain of inequalities,

$$\mathbb{E}\Big\{e^{-\frac{n}{2L^2}\left(\frac{\Delta_{\min}}{2c_{a,t}}-\lambda_n Q\right)^{+2}}\Big||\mathcal{T}_{a,t-1}|=n\Big\} \leq 2e^{-C(\Sigma_{\min})^2 n/4} + e^{\frac{Q\Delta_{\min}\Sigma_{\min}}{4L^2}}e^{-\frac{n(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}}.$$

Note that $||\Sigma|| \leq 1$ follows from our non-restrictive assumption that $\|x_t\|_2 \leq 1$ for all x_t. Before we proceed we need the following lemma:

Lemma 2. *If $n_c = \frac{p}{2k}\log t$, then $\mathbb{P}\{|\mathcal{T}_{a,t-1}| < n_c\} \leq t^{-\frac{p}{16K}}$.*

Proof. First notice that $|\mathcal{T}_{a,t-1}| = \sum_{i=1}^{t-1} z_i$ where $\{z_i\}_{i=1}^{t-1}$ are independent Bernoulli random variables with parameter $p/(Ki)$. Remember that we can assume that $i > p$ since in the beginning of Algorithm 1 we play each arm p/K times.

Note that $\mathbb{P}(X > c) \leq \mathbb{P}(X + q > c)$ is always true for any r.v. X, c and $q > 0$. Now write,

$$\begin{aligned}\mathbb{P}(|\mathcal{T}_{a,t-1}| < n_c) &= \mathbb{P}\left(\sum_{i=1}^{t-1} z_i < n_c\right) = \mathbb{P}\left(\sum_{i=1}^{t-1}(z_i - p/(Ki)) < n_c - (p/K)\sum_{i=1}^{t-1} 1/i\right)\\ &\leq \mathbb{P}\left(\sum_{i=1}^{t-1}(-z_i + p/i) > -n_c + (p/K)\sum_{i=1}^{t-1} 1/i\right)\\ &\leq \mathbb{P}\left(\sum_{i=1}^{t-1}(-z_i + p/i) > (p/K)\log t - n_c\right). \qquad (7)\end{aligned}$$

Since $\sum_{i=1}^{t-1}\mathbb{E}\{(z_i - p/(Ki))^2\} = \sum_{i=p+1}^{t-1}(1-p/(Ki))(p/(Ki)) \leq \frac{p}{K}\log t$, we have that $\{-z_i + p/i\}_{i=1}^{t-1}$ are i.i.d. random variables with zero mean and sum of variances upper bounded by $(p/K)\log t$. Replacing $n_c = (p/2K)\log t$ in (7) and applying Bernstein inequality we get, $\mathbb{P}(|\mathcal{T}_{a,t-1}| < n_c) \leq e^{-\frac{\frac{1}{2}(p/(2K))^2\log^2 t}{\frac{p}{K}\log t + \frac{1}{3}(p/(2K))\log t}} \leq t^{-\frac{p}{16K}}$. □

We can now write, by splitting the sum in $n < n_c$ and $n \geq n_c$

$$\begin{aligned}
&\mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{\tau\in\mathcal{T}} x_t^{\dagger} x_\tau \epsilon_{a,\tau}\Big| \geq \frac{\Delta_{\min}}{2c_{a,t}} - \lambda_n Q\Big\} \\
&\leq \sum_{n=1}^{t} \mathbb{P}\{|\mathcal{T}_{a,t-1}| = n\}\mathbb{E}\Big\{2e^{-\frac{n}{2L^2}\left(\frac{\Delta_{\min}}{2c_{a,t}}-\lambda_n Q\right)^{+2}}\Big||\mathcal{T}_{a,t-1}| = n\Big\} \\
&\leq \mathbb{P}\{|\mathcal{T}_{a,t-1}| < n_c\} + 4e^{-C(\Sigma_{\min})^2 n_c/4} + 2e^{\frac{Q\Delta_{\min}\Sigma_{\min}}{4L^2}} e^{-\frac{n_c(\Delta_{\min})^2(\Sigma_{\min})^2}{32L^2}} \\
&\leq t^{-\frac{p}{16K}} + 4t^{-\frac{Cp(\Sigma_{\min})^2}{8K}} + 2e^{\frac{Q\Delta_{\min}\Sigma_{\min}}{4L^2}} t^{-\frac{p(\Delta_{\min})^2(\Sigma_{\min})^2}{64KL^2}}.
\end{aligned}$$

We want this quantity to be summable over t. Hence we require that,

$$p \geq \frac{128KL^2}{(\Delta_{\min})^2(\Sigma_{\min})^2}, p \geq \frac{16K}{C(\Sigma_{\min})^2}, p \geq 32K. \tag{8}$$

It is immediate to see that our proof also follows if $\Delta_{\min}$, $\Sigma_{\min}$ and L are replaced by $\Delta'_{\min} = \min\{1, \Delta_{\min}\}$, $\Sigma'_{\min} = \min\{1, \Sigma_{\min}\}$ and $L' = \max\{1, L\}$ respectively. If this is done, it is easy to see that conditions (8) are all satisfied by the p stated in Theorem 2. Since $\sum_{t=1}^{\infty} 1/t^2 \leq 2$, gathering all terms together we have,

$$\begin{aligned}
&R(T) \leq p\Delta_{\max}\sqrt{d} + p\Delta_{\max}\sqrt{d}\log T + \Delta_{\max}\sqrt{d}K\left(4e^{\frac{Q\Delta'_{\min}\Sigma'_{\min}}{4L'^2}} + 10\right) \\
&\leq p\Delta_{\max}\sqrt{d} + 14\Delta_{\max}\sqrt{d}Ke^{Q/4} + p\Delta_{\max}\sqrt{d}\log T. \qquad \square
\end{aligned}$$

6 Numerical Results

In Theorem 2, we showed that, in the stochastic setting, Algorithm 1 has an expected regret of $O(\log T)$. We now illustrate this point by numerical simulations and, most importantly, exemplify how violating the stochastic assumption might degrade its performance. Figure 1 (a) shows the average cumulative regret (in semi-log scale) over 10 independent runs of Algorithm 1 for $T = 10^5$ and for the following setup. The context variables $x \in \mathbb{R}^3$ and at each time step $\{x_t\}_{t\geq 1}$ are drawn i.i.d. in the following way: (a) set each entry of x to 1 or 0 independently with probability 1/2; (b) normalize x. We consider $K = 6$ arms with corresponding parameters θ_a generated independently from a standard multivariate gaussian distribution. Given a context x and an arm a, rewards were random and independently generated from a uniform distribution $U([0, 2x^{\dagger}\theta_a])$. As expected, the regret is logarithmic. Figure 1 (a) shows a straight line at the end.

To understand the effect of the stochasticity of x on the regret, we consider the following scenario: with every other parameter unchanged, let $\mathcal{X} = \{x, x'\}$. At every time step $x = [1, 1, 1]$ appears with probability $1/I$, and $x' = [1, 0, 1]$ appears with probability $1 - (1/I)$. Figure 1 (b) shows the dependency of the expected regret on the context distribution for I = 5, 10 and 100. One can see that an increase of I causes a proportional increase in the regret.

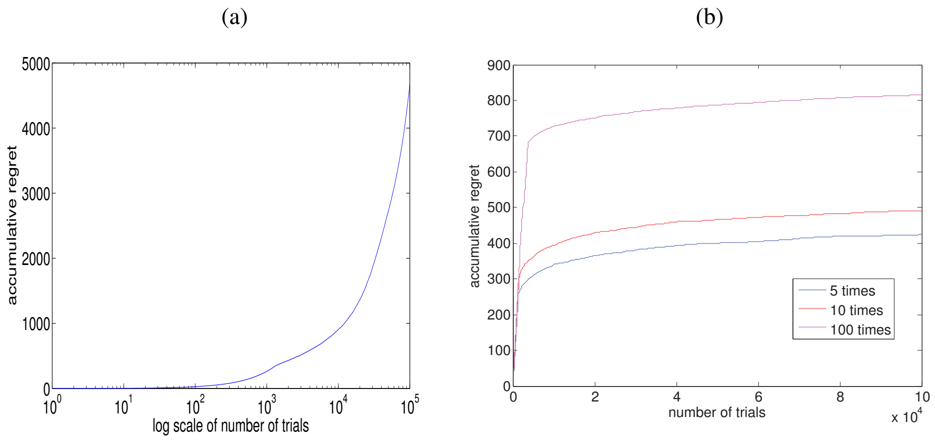

Fig. 1. (a) Regret over T when x_t is from i.i.d. (b) Regret over T when x_t is not from i.i.d.

7 Adversarial Setting

In the stochastic setting, the richness of the subset of $\mathbb{R}^d$ spanned by the observed contexts is related to the skewness of the distribution $\mathcal{D}$. The fact that the bound in Theorem 2 depends on $\Sigma_{\min}$ and that the regret increases as this value becomes smaller indicates that our approach does not yield a $O(\log T)$ regret for the adversarial setting, where an adversary choses the contexts and can, for example, generate $\{x_t\}$ from a sequence of stochastic processes with decreasing $\Sigma_{\min}(t)$.

In particular, the main difficulty in using a linear regression, and the reason why our result depends on $\Sigma_{\min}$, is related to the dependency of our estimation of $x_t^\dagger \theta_a$ on $\frac{1}{|\mathcal{T}_{a,t-1}|} X^\dagger_{\mathcal{T}_{a,t-1}} X_{\mathcal{T}_{a,t-1}}$. It is not hard to show that the error in approximating $x_t^\dagger \theta_a$ with $x_t^\dagger \hat{\theta}_a$ is proportional to

$$\sqrt{x_t^\dagger \left(\lambda_n I + \frac{1}{n} X^\dagger_{\mathcal{T}} X_{\mathcal{T}} \right)^{-2} x_t}. \tag{9}$$

This implies that, even if a given context has been observed relatively often in the past, the algorithm can "forget" it because of the *mean* over contexts that is being used to produce estimates of $x_t^\dagger \theta_a$ (the mean shows up in (9) as $\frac{1}{n} X^\dagger_{\mathcal{T}} X_{\mathcal{T}}$).

The effect of this phenomenon on the performance of Algorithm 1 can be readily seen in the following pathological example. Assume that $\mathcal{X} = \{(1,1),(1,0)\} \subset \mathbb{R}^2$. Assume that the contexts arrive in the following way: $(1,1)$ appears with probability $1/I$ and $(1,0)$ appears with probability $1-1/I$. The correlation matrix for this stochastic process is $\{(1,1/I),(1/I,1/I)\}$ and its minimum eigenvalue scales like $O(1/I)$. Hence, the regret scales as $O(I^2 \log T)$. If I is allowed to slowly grow with t, we expect that our algorithm will not be able to guarantee a logarithmic regret (assuming that our upper bound is tight). In other words, although $(1,1)$ might have appeared a sufficient number of times for us to be able to predict the expected reward for this context, Algorithm 1 performs poorly since the mean (9) will the 'saturated' with the context $(1,0)$ and forget about $(1,1)$.

Algorithm 2 Contextual UCB

```
for $t = 1$ to $p$ do
    $a \leftarrow 1 + (t \mod K)$; Play arm $a$; $\mathcal{T}_{a,t} \leftarrow \mathcal{T}_{a,t-1} \cup \{t\}$
end for
for $t = p + 1$ to $T$ do
    for $a \in \mathcal{A}$ do
        $c_{a,t} \leftarrow \min_{\mathcal{T} \subset \mathcal{T}_{a,t-1}} \frac{\log t}{|\mathcal{T}|} x_t^\dagger \left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right)^{-2} x_t$
        $\mathcal{T}^* \leftarrow$ subset of $\mathcal{T}_{a,t-1}$ that achieves the minimum; $n \leftarrow |\mathcal{T}^*|$
        Get $\hat{\theta}_a$ as the solution to the linear system: $\left( \lambda_n I + \frac{1}{n} X_{\mathcal{T}}^\dagger X_{\mathcal{T}} \right) \hat{\theta}_a = \left( \frac{1}{n} X_{\mathcal{T}}^\dagger r_{\mathcal{T}} \right)$
    end for
    Play arm $a_t = \arg\max_a x_t^\dagger \hat{\theta}_a + \sqrt{c_{a,t}}$; Set $\mathcal{T}_{a,t} \leftarrow \mathcal{T}_{a,t-1} \cup \{t\}$
end for
```

One solution for this problem is to ignore some past contexts when building an estimate for $x_t^\dagger \theta_a$, by including in the mean (9) past contexts that are closer in direction to the current context x_t. Having this in mind, and building on the ideas of [4], we propose the UCB-type Algorithm 2.

It is straightforward to notice that this algorithm cannot be implemented in an efficient way. In particular, the search for $\mathcal{T}^* \subset \mathcal{T}_{a,t-1}$ has a computational complexity exponential in t. The challenge is to find an efficient way of approximating $\mathcal{T}^*$ efficiently. This can be done by either reducing the size of $\mathcal{T}_{a,t-1}$ – the history from which one wants to extract $\mathcal{T}_{a,t-1}$ – by not storing all events in memory (for example, if we can guarantee that $|\mathcal{T}_{a,t}| = O(\log t)$ then the complexity of the above algorithm at time step t is $O(t)$), or by finding an efficient algorithm of approximating the minimization over the $\mathcal{T}_{a,t-1}$ (or both). It remains an open problem to find such an approximation scheme and to prove that it achieves $O(\log T)$ regret for a setting more general than the i.i.d. contexts considered in this paper.

8 Conclusions

We introduced an ϵ-greedy type of algorithm that provably achieves logarithmic regret for the contextual multi-armed bandits problem with linear payoffs in the stochastic setting. Our online algorithm is both fast and uses small space. In addition, our bound on the regret scales nicely with dimension of the contextual variables, $O(poly(d) \log T)$. By means of numerical simulations we illustrate how the stochasticity of the contexts is important for our bound to hold. In particular, we show how to construct a scenario for which our algorithm does not give logarithmic regret. The reason for this amounts to the fact that the mean $n^{-1} X_{\mathcal{T}}^\dagger X_{\mathcal{T}}$ that is used in estimating the parameters θ_a can "forget" previously observed contexts. Because of this, it remains an open problem to show that there are efficient algorithms that achieve $O(poly(d) \log T)$ under reward separation ($\Delta_{\min} > 0$) in the non-stochastic setting. We believe that a possible solution might be constructing a variant of our algorithm where in $n^{-1} X_{\mathcal{T}}^\dagger X_{\mathcal{T}}$ we use a more careful average of past observed contexts give the current observed context. In addition, we leave it open to produce simple and efficient online algorithms for multi-armed bandit problems under rich context models, like the one we have done here for linear payoff.

References

1. Abbasi-Yadkori, Y., Pál, D., Szepesvári, C.: Improved algorithms for linear stochastic bandits. Advances in Neural Information Processing Systems (2011)
2. Abernethy, J., Hazan, E., Rakhlin, A.: Competing in the dark: An efficient algorithm for bandit linear optimization. In: Proceedings of the 21st Annual Conference on Learning Theory (COLT). vol. 3, p. 3 (2008)
3. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning 47(2), 235–256 (2002)
4. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. The Journal of Machine Learning Research 3, 397–422 (2003)
5. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: Gambling in a rigged casino: The adversarial multi-armed bandit problem. In: Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on. pp. 322–331. IEEE (1995)
6. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multiarmed bandit problem. SIAM Journal on Computing 32(1), 48–77 (2002)
7. Bartlett, P., Ben-David, S.: Hardness results for neural network approximation problems. In: Computational Learning Theory. pp. 50–62. Springer (1999)
8. Beygelzimer, A., Langford, J., Li, L., Reyzin, L., Schapire, R.E.: Contextual bandit algorithms with supervised learning guarantees. Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS) (2011)
9. Chu, W., Li, L., Reyzin, L., Schapire, R.E.: Contextual bandits with linear payoff functions. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS) (2011)
10. Crammer, K., Gentile, C.: Multiclass classification with bandit feedback using adaptive regularization. Proceedings of the 28th International Conference on Machine Learning (2011)
11. Dani, V., Hayes, T., Kakade, S.M.: The price of bandit information for online optimization. Advances in Neural Information Processing Systems 20, 345–352 (2008)
12. Dani, V., Hayes, T.P., Kakade, S.M.: Stochastic linear optimization under bandit feedback. In: Proceedings of the 21st Annual Conference on Learning Theory (COLT). pp. 355–366 (2008)
13. Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., Zhang, T.: Efficient optimal learning for contextual bandits. UAI (2011)
14. Hazan, E., Kale, S.: Newtron: an efficient bandit algorithm for online multiclass prediction. Advances in Neural Information Processing Systems (NIPS) (2011)
15. Johnson, D.S., Preparata, F.P.: The densest hemisphere problem. Theoretical Computer Science 6(1), 93–107 (1978)
16. Kakade, S.M., Shalev-Shwartz, S., Tewari, A.: Efficient bandit algorithms for online multiclass prediction. In: Proceedings of the 25th international conference on Machine learning. pp. 440–447. ACM (2008)
17. Langford, J., Zhang, T.: The epoch-greedy algorithm for contextual multi-armed bandits. Advances in Neural Information Processing Systems 20, 1096–1103 (2007)
18. Li, L., Chu, W., Langford, J., Schapire, R.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670. ACM (2010)
19. Rusmevichientong, P., Tsitsiklis, J.: Linearly parameterized bandits. Mathematics of Operations Research 35(2) (2010)
20. Sutton, Barto: Reinforcement learning, and introduction. Cambdrige, MIT Press (1998)
21. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. Chapter 5 of: Compressed Sensing, Theory and Applications (2012)