Regret Bounds for Batched Bandits

Hossein Esfandiari,[1] **Amin Karbasi,**[2] **Abbas Mehrabian,**[3] **Vahab Mirrokni**[1]
[1]Google Research, New York City, New York, USA
[2]School of Engineering and Applied Science, Yale University, New Haven, Connecticut, USA
[3]McGill University, Montréal, Quebec, Canada
esfandiari@google.com, amin.karbasi@yale.edu, abbas.mehrabian@gmail.com, mirrokni@google.com

Abstract

We present simple algorithms for batched stochastic multi-armed bandit and batched stochastic linear bandit problems. We prove bounds for their expected regrets that improve and extend the best known regret bounds of Gao, Han, Ren, and Zhou (NeurIPS 2019), for any number of batches. In particular, our algorithms in both settings achieve the optimal expected regrets by using only a logarithmic number of batches. We also study the batched adversarial multi-armed bandit problem for the first time and provide the optimal regret, up to logarithmic factors, of any algorithm with predetermined batch sizes.

1 Introduction

A central challenge in optimizing many complex systems, such as experimental design (Robbins 1952), clinical trials (Perchet et al. 2016), hyperparameter tuning (Snoek et al. 2015), and product marketing (Bertsimas and Mersereau 2007), is to simultaneously explore the unknown parameter space while at the same time exploit the acquired knowledge for maximizing the utility. In theory, the most convenient way is to explore one parameter at a time. However, in practice, it is often possible/desirable, and sometimes the only way, to explore several parameters in parallel. A notable example is designing clinical trials, where it is impractical to wait to observe the effect of a drug on a single patient before deciding about the next trial. Instead, groups of patients with multiple treatments are studied in parallel. Similarly, in marketing and advertising, the efficacy of a strategy is not tested on individual subjects one at a time; instead, multiple strategies are run simultaneously in order to gather information in a timely fashion. Similar issues arise in crowdsourcing platforms, where multiple tasks are distributed among users (Kittur, Chi, and Suh 2008), and time-consuming numerical simulations, which are prevalent in reinforcement learning (Le, Voloshin, and Yue 2019; Lange, Gabel, and Riedmiller 2012).

Parallelizing the exploration of the parameter space has a clear advantage: more information can be gathered at a shorter period of time. It also has a clear disadvantage: information cannot be immediately shared across different parallel paths, thus future decisions cannot benefit from the intermediate results. Note that a fully sequential policy, in which exploration is done in a fully sequential manner, and a fully parallel policy, in which exploration is completely specified a priori without any information exchange, are two extremes of the policy spectrum. In fact, carefully calibrating between the information parallelization phase (how many experiments should be run in parallel) and the information exchange phase (how often information should be shared) is crucial for applications in which running experiments is costly or time-consuming. *Batch policies*, which are the focus of this paper, aim to find the sweet spot between these two phases. The main challenge is to carefully design batches of experiments, out of a combinatorially large set of possibilities, which can be run in parallel and explore the parameter space efficiently while being able to exploit the parameter region with the highest utility.

In this paper, we study the problem of batch policies in the context of multi-armed and linear bandits with the goal of minimizing *regret*, the standard benchmark for comparing performance of bandit policies. We advance the theoretical understanding of these problems by designing algorithms along with hardness results. In particular, we prove bounds for batched stochastic multi-armed bandits that improve and extend the best known regret bounds of Gao et al. (2019), for any number of batches.

2 Bandits, Regret, and Batch Policies

A *bandit problem* is a game between a player/learner and an environment. The game is played over T rounds, called the *time horizon*. In each round, first the player chooses an action from a set of actions and then the environment presents a reward. For instance, in clinical trials, the actions correspond to the available treatments and the rewards correspond to whether the treatment cured the patient or not. As the player does not know the future, she follows a *policy*, a mapping from histories to actions. Similarly, the environment can also be formulated as a mapping from actions to rewards. Note that both the player and the environment may randomize their decisions. The standard performance measure for a bandit policy is its *regret*, defined as the difference between the total expected reward collected by the policy and the total expected reward collected by an optimal policy. Many bandit models exist depending on the type of the environment; next we define those which we will use here.

Multi-Armed Bandits

Traditionally, actions are referred to as 'arms' and 'taking an action' is referred to as 'pulling an arm.' A *multi-armed bandit* is a one-player game in which the number of arms is a finite number K. Let $[K] \coloneqq \{1, 2, \dots, K\}$ denote the set of arms. In each round $t = 1, 2, \dots, T$, the player pulls an arm $a_t \in [K]$ and receives a corresponding reward r_t.

We consider two possible generation models for the rewards: the *stochastic* setting and the *adversarial* setting. In the former, the rewards of each arm are sampled in each round independently from some fixed distribution supported on $[0, 1]$. In other words, each arm has a potentially different reward distribution, while the distribution of each arm does not change over time. Suppose the player pulls the arms $a_1, a_2, \dots, a_T$, and suppose μ_i denotes the mean of arm $i \in [K]$. Then, in the stochastic setting, the *expected regret* is defined as

$$\mathbf{E}[\text{Regret}] \coloneqq T \max_{i \in [K]} \mu_i - \mathbf{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right].$$

In contrast to the stochastic setting, a multi-armed adversarial bandit is specified by an *arbitrary* sequence of rewards $(r_{i,t})_{i \in [K], t \in [T]} \in [0, 1]$. In each round t, the player chooses a distribution P_t; an arm $a_t \in [K]$ is sampled from P_t and the player receives reward $r_{a_t,t}$. In this sense, a policy can be viewed as a function mapping history sequences to distributions over arms. The expected regret is defined as the difference between the expected reward collected by the policy and the best fixed action in hindsight, i.e.,

$$\mathbf{E}[\text{Regret}] \coloneqq \max_{i \in [K]} \sum_{t=1}^{T} r_{i,t} - \mathbf{E}\left[\sum_{t=1}^{T} r_{a_t,t}\right].$$

Note that the only source of randomness in the regret stems from the randomized policy used by the player. Randomization is indeed crucial for the player and is the only way for her to avoid a regret of $\Omega(T)$.

Remark. The assumption that the rewards are bounded in $[0, 1]$ is a standard normalization assumption in online learning, but it is immediate to generalize our algorithms and analyses to reward distributions bounded in any known interval, or (in the stochastic case) to Gaussian or subgaussian distributions whose mean lie in a known interval. The only change in the analysis is that instead of using Hoeffding's inequality which requires a bounded distribution, one has to use a concentration inequality for sums of subgaussian distributions, see, e.g., Wainwright (2019, Proposition 2.5).

Stochastic Linear Bandits

In a stochastic linear bandit, each arm is a vector $a \in \mathbf{R}^d$ belonging to some action set $\mathcal{A} \subseteq \mathbf{R}^d$, and there is a parameter $\theta^\star \in \mathbf{R}^d$ unknown to the player. In round t, the player chooses some action $a_t \in \mathcal{A}$ and receives reward $r_t = \langle a_t, \theta^\star \rangle + \mu_t$, where μ_t is a zero-mean 1-subgaussian noise; that is, μ_t is independent of other random variables, has $\mathbf{E}\mu_t = 0$ and satisfies, for all real λ, $\mathbf{E}\left[e^{\lambda \mu_t}\right] \leq \exp(\lambda^2/2)$. Note that any zero-mean Gaussian distribution with variance at most 1 and any zero-mean distribution supported on an interval of length at most 2 satisfies the above inequality. For normalization purposes, we assume that $\|\theta^\star\|_2 \leq 1$ and $\|a\|_2 \leq 1$ for all arms $a \in \mathcal{A}$.

Denoting the pulled arms by $a_1, \dots, a_T$, the expected regret of a stochastic linear bandit algorithm is

$$\mathbf{E}[\text{Regret}] \coloneqq T \sup_{a \in \mathcal{A}} \langle a, \theta^\star \rangle - \mathbf{E}\left[\sum_{t=1}^{T} \langle a_t, \theta^\star \rangle\right].$$

Batch Policies

As opposed to the bandit problems described above, in the *batch mode*, the player commits to a sequence of actions (a *batch* of actions) and observes the rewards *after all actions in that sequence are played*. More formally, at the beginning of each batch $i = 1, 2, \dots$, the player announces a list of arms/actions to be pulled/played. Afterwards, she receives a list of pairs consisting of arm indices and rewards, corresponding to the rewards generated from these pulls. Then the player decides about the next batch.

The batch sizes could be chosen non-adaptively or adaptively. In a *non-adaptive* policy, the player fixes the batch *sizes* before starting the game, while in an *adaptive* policy, the batch sizes may depend on the observations of the player. Obviously, an adaptive policy is more powerful and may achieve a smaller regret. In both cases, the player is subject to using at most a given number of batches, B. Moreover, the total number of actions played by the player must sum to the *horizon* T. We assume that the player knows the values of B and T. Notice that the case $B = T$ corresponds to original bandit problems where actions are committed fully sequentially and has been studied extensively, see, e.g., Lattimore and Szepesvári (2020). Thus, we refer to the case $B = T$ as the *original* or the *sequential* setting.

Our algorithms for stochastic bandits are adaptive, while in the adversarial setting we focus mostly on non-adaptive algorithms.

3 Contributions and Paper Outline

We provide analytic regret bounds for the batched version of three bandit problems: stochastic multi-armed bandits, stochastic linear bandits, and adversarial multi-armed bandits.

Recall that K denotes the number of arms, T the time horizon and B the number of batches. The case $B = T$ corresponds to the sequential setting which has been studied extensively, while if $B = 1$ then no learning can happen, thus we are mostly interested in the regime $1 < B < T$.

Stochastic Multi-Armed Bandits

Let $\Delta_i \coloneqq \max_{a \in [K]} \mu_a - \mu_i \geq 0$ denote the gap of arm i. For stochastic multi-armed bandits, the optimum regret achievable in the easier sequential setting is $O\Big(\log(T) \sum_{i:\Delta_i > 0} {\Delta_i}^{-1}\Big)$; this is achieved, e.g., by the well-known upper confidence bound (UCB) algorithm of Auer, Cesa-Bianchi, and Fischer (2002).

Our first contribution is a simple and efficient algorithm (Algorithm 1) whose regret scales as

$O\left(T^{1/B}\log(T)\sum_{i:\Delta_i>0}{\Delta_i}^{-1}\right)$ (Theorem 1). In particular, as soon as $B \geq C\log T$ for some absolute constant C, it matches the optimum regret achievable in the fully sequential setting. In other words, increasing the number of batches from $C\log T$ to T does not reduce the expected regret by more than a constant multiplicative factor. Gao et al. (2019, Corollary 2) show that $B = \Omega(\log T/\log\log T)$ batches are *necessary* to achieve the optimal regret $O\Big(\log(T)\sum_{i:\Delta_i>0}\Delta_i^{-1}\Big)$. This lower bound implies that our algorithm uses almost the minimum number of batches needed (i.e., $O(\log T)$ versus $\Omega(\log T/\log\log T)$) to achieve the optimal regret.

Our result improves the state-of-the-art result of Gao et al. (2019, Theorem 4), who provide a non-adaptive algorithm with

$$\mathbf{E}[\text{Regret}] \leq O\left(\frac{K\log(K)\log(T)T^{1/B}}{\min_{i:\Delta_i>0}\Delta_i}\right).$$

Our Theorem 1 shaves off a factor of $\log(K)$ and also improves the factor $\frac{K}{\min\Delta_i}$ to $\sum_i\frac{1}{\Delta_i}$. This latter improvement may be as large as a multiplicative factor of K in some instances, e.g., if $\mu_1 = 1, \mu_2 = 1-1/K, \mu_3 = \cdots = \mu_K = 0$. The algorithm of Gao et al. (2019) does not achieve this improved regret because their batch sizes are predetermined. We achieve these improvements by adapting the batch sizes based on the previous outcomes as opposed to predetermined batch sizes.

Stochastic Linear Bandits

In the sequential setting of stochastic linear bandits, the best-known upper bound for regret is $O(d\sqrt{T\log(T)})$ (Lattimore and Szepesvári 2020, Note 2 in Section 21.2), while the best known lower bound is $\Omega(d\sqrt{T})$ (Lattimore and Szepesvári 2020, Theorem 24.2). For a finite number of arms K, the best known upper bound is $O(\sqrt{dT\log K})$ (Bubeck, Cesa-Bianchi, and Kakade 2012).

Our second contribution is the first algorithm for batched stochastic linear bandits (Algorithm 2): an efficient algorithm for the case of finitely many arms with regret $O\left(T^{1/B}\sqrt{dT\log(KT)}\right)$ (Theorem 4), matching the $O(\sqrt{dT\log(K)})$ upper bound for the sequential setting as soon as the number of batches is $\Omega(\log T)$. If there are infinitely many arms, we achieve a regret upper bound of $O\left(T^{1/B}\cdot d\sqrt{T\log(T)}\right)$, which matches the best-known upper bound of $O(d\sqrt{T\log T})$ for the sequential setting as soon as the number of batches is $\Omega(\log T)$.

Our algorithm for batched stochastic linear bandits is based on arm eliminations. However, it would blow up the regret if we were to pull each arm in each batch the same number of times; instead we use the geometry of the action set to carefully choose a sequence of arms in each batch, based on an approximately optimal G-design, that would give us the information needed to gradually eliminate the suboptimal arms. The extension to the case of infinitely many arms is achieved via a discretization argument.

For the case of finitely many arms, the algorithm's running time is polynomial in the number of arms and the time horizon. When there are infinitely many arms, the running time is exponential in the dimension.

Adversarial Multi-Armed Bandits

The optimal regret for adversarial multi-armed bandits in the sequential setting is $\Theta(\sqrt{KT})$, see Audibert and Bubeck (2009). Our third contribution is to prove that the best achievable regret of any non-adaptive algorithm for batched adversarial multi-armed bandits is $\widetilde{\Theta}\left(\sqrt{T\left(K+\frac{T}{B}\right)}\right)$, where the $\widetilde{\Theta}$ allows for logarithmic factors (Theorem 7). That is, we prove an optimal (minimax) regret bound, up to logarithmic factors, for non-adaptive algorithms for batched adversarial multi-armed bandits.

Finally, we prove a lower bound of $\Omega(T/B)$ for the regret of *any* algorithm, adaptive or non-adaptive, for batched adversarial multi-armed bandits (Theorem 8). This shows a large contrast with the stochastic version, since there is a polynomial relation between the number of batches and the regret. In particular, one needs at least $\Omega(\sqrt{T/K})$ batches to achieve the optimum regret of $O(\sqrt{TK})$.

The upper bound for batched adversarial multi-armed bandits is proved via a reduction to the setting of multi-armed bandits with delays, while the lower bounds are proved by carefully designing hard reward sequences.

Paper Outline In the next section we review prior work on the batched bandits model. Stochastic multi-armed bandits, stochastic linear bandits, and adversarial multi-armed bandits are studied in Sections 5, 6, and 7 respectively. We conclude with a discussion and directions for further research in Section 8.

4 Related Work

Sequential bandit problems, in particular multi-armed bandits, have been studied for almost a century. While we cannot do justice to all the work that has been done, let us highlight a few excellent monographs (Bubeck and Cesa-Bianchi 2012; Slivkins 2019; Lattimore and Szepesvári 2020). We now review the results in the batched setting.

Auer and Ortner (2010) present an algorithm for stochastic multi-armed bandits based on arm elimination. Even though their algorithm is presented for the sequential setting, it can be turned into an algorithm for the batched setting for $\Omega(\log T)$ batches. More precisely, they prove that the optimal problem-dependent regret of $O\Big(\log(T)\sum_{i:\Delta_i>0}{\Delta_i}^{-1}\Big)$ is achievable as soon as the number of batches is $B = \Omega(\log T)$. Our results, however, are more general and hold for *any* number of batches $B \in \{1,\ldots,T\}$. We emphasize that it is crucial to have algorithms for small number of batches; for designing clinical trials in time-sensitive situations, for instance for COVID-19, it is likely that one needs to design a trial with much

fewer than $\log(T)$ batches, as it takes a couple of weeks to receive the feedback for each batch.

Similarly, Cesa-Bianchi, Dekel, and Shamir (2013, Theorem 6) provide an algorithm using $B = O(\log \log T)$ batches that achieves problem-independent regret of $O(\sqrt{KT \log K})$ for stochastic multi-armed bandits. (Their algorithm is also for the sequential version but it can be batched easily.) This rate is minimax optimal for the sequential version up to a $\sqrt{\log K}$ factor. However, for stochastic multi-armed bandits, our focus is on problem-dependent bounds with logarithmic dependence on T.

A special case of batched multi-armed bandits was studied by Perchet et al. (2016); they consider only two arms and provide upper and lower bounds on the regret. In particular, denoting the gap between the arms by Δ, Perchet et al. (2016, Theorem 2) provide a regret bound of

$$\mathbf{E}[\text{Regret}] \leq O\left(\left(\frac{T}{\log T}\right)^{1/B} \frac{\log T}{\Delta}\right)$$

when $K = 2$. In contrast, we consider the more general setting of $K \geq 2$ and we give an algorithm for *any* number of arms K satisfying

$$\mathbf{E}[\text{Regret}] \leq O\left(T^{1/B} \sum_{i:\Delta_i > 0} \frac{\log(T)}{\Delta_i}\right).$$

The other bandit problems we study, namely, stochastic linear bandits and adversarial multi-armed bandits, have not been studied in the batched setting prior to our work.

Optimization in batch mode has also been studied in other machine learning settings where information-parallelization is very effective. Examples include best arm identification (Jun et al. 2016; Agarwal et al. 2017), bandit Gaussian processes (Desautels, Krause, and Burdick 2014; Kathuria, Deshpande, and Kohli 2016; Contal et al. 2013), submodular maximization (Balkanski and Singer 2018; Fahrbach, Mirrokni, and Zadimoghaddam 2019; Chen, Feldman, and Karbasi 2019), stochastic sequential optimization (Esfandiari, Karbasi, and Mirrokni 2019; Agarwal, Assadi, and Khanna 2019), active learning (Hoi et al. 2006; Chen and Krause 2013), and reinforcement learning (Ernst, Geurts, and Wehenkel 2005), to name a few.

5 Batched Stochastic Multi-Armed Bandits

Our algorithm works by gradually eliminating suboptimal arms. Let $\delta := 1/(2KTB)$ and $q := T^{1/B}$, and define $c_i := \lfloor q^1 \rfloor + \cdots + \lfloor q^i \rfloor$. Note that $c_B \geq T$. Initially, all arms are 'active.' In each batch $i = 1, 2, \ldots$, except for the last batch, each active arm is pulled $\lfloor q^i \rfloor$ times. Then, after the rewards of this batch are observed, the mean of each active arm is estimated as the average reward received from its pulls. An arm is then eliminated if its estimated mean is smaller, by at least $\sqrt{2\ln(1/\delta)/c_i}$, than the estimated mean of another active arm. The last batch is special: if we have used $i - 1$ batches so far and the number of active arms times $\lfloor q^i \rfloor$ exceeds the number of remaining rounds, the size of the next batch equals the number of remaining rounds, and in this last batch we pull the active arm with the largest empirical mean.

Algorithm 1 Batched arm elimination for stochastic multi-armed bandits

Input: number of arms K, time horizon T, number of batches B
$q \longleftarrow T^{1/B}$
$\mathcal{A} \longleftarrow [K]$ {active arms}
$\widehat{\mu}_a \longleftarrow 0$ for all $a \in \mathcal{A}$ {estimated means}
for $i = 1$ **to** $B - 1$ **do**
  if $\lfloor q^i \rfloor \times |\mathcal{A}| >$ remaining rounds **then**
    break
  end if
  In the ith batch, play each arm $a \in \mathcal{A}$ for $\lfloor q^i \rfloor$ times
  Update $\widehat{\mu}_a$ for all $a \in \mathcal{A}$
  $c_i \longleftarrow \sum_{j=1}^{i} \lfloor q^j \rfloor$
  for $a \in \mathcal{A}$ **do**
    if $\widehat{\mu}_a < \max_{\alpha \in \mathcal{A}} \widehat{\mu}_\alpha - \sqrt{2\ln(2KTB)/c_i}$ **then**
      Remove a from $\mathcal{A}$
    end if
  end for
end for
In the last batch, play $\operatorname{argmax}_{a \in \mathcal{A}} \widehat{\mu}_a$

See Algorithm 1 for the pseudocode. We bound its regret as follows.

Theorem 1. *The expected regret of Algorithm 1 for batched stochastic multi-armed bandits is bounded by* $\mathbf{E}[\text{Regret}] \leq 9T^{1/B} \ln(2KTB) \sum_{j:\Delta_j > 0} \frac{1}{\Delta_j}$.

For proving this theorem, we will use Hoeffding's inequality.

Theorem 2 (Hoeffding's inequality, see Theorem 2 in Hoeffding (1963))**.** *Suppose $X_1, \ldots, X_n$ are independent, identically distributed random variables supported on $[0, 1]$. Then, for any $t \geq 0$,*

$$\mathbf{Pr}\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbf{E}X_1 > t\right|\right] < 2\exp(-2nt^2).$$

Proof of Theorem 1. For an active arm at the end of some batch i, we say its estimation is 'correct' if the estimation of its mean is within $\sqrt{\ln(1/\delta)/2c_i}$ of its actual mean. Since each active arm is pulled c_i times by the end of batch i, by Hoeffding's inequality, the estimation of any active arm at the end of any batch (except possibly the last batch) is correct with probability at least $1 - 2\delta$. Note that there are K arms and at most B batches (since $c_B \geq T$). Hence, by the union bound, the probability that the estimation is incorrect for some arm at the end of some batch is bounded by $KB \times 2\delta = 1/T$. If some estimation is incorrect, we upper bound the regret by T. Let $\mathcal{E}$ denote the event that all estimations are correct. Hence, the expected regret can be bounded as $\mathbf{E}[\text{Regret}] \leq \frac{1}{T} \times T + \mathbf{E}[\text{Regret}|\mathcal{E}]$.

So, from now on, we assume that $\mathcal{E}$ happens, which implies that each *gap* is correctly estimated within an additive factor of $\sqrt{2\ln(1/\delta)/c_i}$. Thus, the best arm is never eliminated. We can then write the expected regret as $\mathbf{E}[\text{Regret}|\mathcal{E}] = \sum_{j:\Delta_j > 0} \Delta_j \mathbf{E}[T_j|\mathcal{E}]$, where T_j denotes the

total number of pulls of arm j. Let us now fix some (suboptimal) arm j with $\Delta_j > 0$, and upper bound T_j. Suppose batch $i+1$ is the last batch that arm j was active. Since this arm is not eliminated at the end of batch i, and the estimations are correct, we have $\Delta_j \le 2\sqrt{2\ln(1/\delta)/c_i}$ deterministically, which means $c_i \le 8\ln(1/\delta)\Delta_j^{-2}$. The total pulls of this arm is thus $T_j \le c_{i+1} = q + qc_i \le q + \frac{8q\ln(1/\delta)}{\Delta_j^2}$, whence,

$$\begin{aligned}\mathbf{E}[\text{Regret}|\mathcal{E}] &\le \sum_{j:\Delta_j>0}\left\{q\Delta_j + \frac{8q\ln(1/\delta)}{\Delta_j}\right\} \\ &\le q(K-1) + 8q\ln(1/\delta)\sum_{j:\Delta_j>0}\frac{1}{\Delta_j}.\end{aligned}$$

Substituting the values of q and δ completes the proof. □

6 Batched Stochastic Linear Bandits

Our algorithm is based on arm elimination, as in the multi-armed case. Here is the key lemma, which follows from the results in Lattimore and Szepesvári (2020, Chapter 21). Note that we may assume without loss of generality that the action set spans $\mathbf{R}^d$, otherwise we may work in the space spanned by the action set.

Lemma 3. *For any finite action set $\mathcal{A}$ that spans $\mathbf{R}^d$ and any $\delta, \varepsilon > 0$, we can find, in time polynomial in $|\mathcal{A}|$, a* multi-set *of $\Theta(d\log(1/\delta)/\varepsilon^2)$ actions (possibly with repetitions) such that they span $\mathbf{R}^d$ and if we perform them in a batched stochastic linear bandits setting and let $\widehat{\theta}$ be the least-squares estimate for $\theta^\star$, then, for any $a \in \mathcal{A}$, with probability at least $1-\delta$ we have $\left|\left\langle a, \widehat{\theta} - \theta^\star\right\rangle\right| \le \varepsilon$.*

Remark. If the performed actions are $a_1, \ldots, a_n$ and the received rewards are $r_1, \ldots, r_n$, then the least squares estimate for $\theta^\star$ is $\widehat{\theta} := \left(\sum_{i=1}^n a_i a_i^\mathsf{T}\right)^{-1}\left(\sum_{i=1}^n r_i a_i\right)$.

Remark. It is known that a multi-set of actions with the guarantee of Lemma 3 exists and has size at most $\frac{d^2+d}{2} + \frac{d\ln(2/\delta)}{\varepsilon^2}$; it can be defined based on a so-called G-optimal design for $\mathcal{A}$, see Lattimore and Szepesvári (2020, equation (21.3)), or geometrically, the minimum volume ellipsoid containing $\mathcal{A}$. An algorithm, with running time polynomial in $|\mathcal{A}|$, for finding an approximate G-optimal design of size $\Theta(d\log(1/\delta)/\varepsilon^2)$ is given in Lattimore and Szepesvári (2020, Note 3 in Section 21.2).

Let B denote the number of batches, and let c and C be the constants hidden in the Θ notation in Lemma 3; namely, the size of the multi-set is in $[cd\log(1/\delta)/\varepsilon^2, Cd\log(1/\delta)/\varepsilon^2]$. Define $q := (T/c)^{1/B}$ and $\varepsilon_i := \sqrt{d\log(KT^2)/q^i}$.

We now describe the algorithm. Initially, all arms are active. In each batch $i = 1, 2, \ldots$, except for the last batch, we compute the multi-set given by Lemma 3, with $\mathcal{A}$ being the set of active arms, $\delta := 1/KT^2$ and $\varepsilon = \varepsilon_i$; we then perform the actions given by the lemma, compute $\widehat{\theta}_i$, and eliminate any arm a with

$$\left\langle a, \widehat{\theta}_i\right\rangle < \max_{\text{active } a}\left\langle a, \widehat{\theta}_i\right\rangle - 2\varepsilon_i. \tag{1}$$

Algorithm 2 Batched arm elimination for stochastic linear bandits

Input: action set $\mathcal{A} \subseteq \mathbf{R}^d$, time horizon T, number of batches B
$q \longleftarrow (T/c)^{1/B}$
for $i = 1$ **to** $B-1$ **do**
  $\varepsilon_i \longleftarrow \sqrt{d\log(KT^2)/q^i}$
  $a_1, \ldots, a_n \longleftarrow$ multi-set given by Lemma 3 with parameters $\delta = 1/KT^2$ and $\varepsilon = \varepsilon_i$
  if $n >$ remaining rounds **then**
    break
  end if
  In the ith batch, play the arms $a_1, \ldots, a_n$ and receive rewards $r_1, \ldots, r_n$
  $\widehat{\theta} \longleftarrow \left(\sum_{i=1}^n a_i a_i^\mathsf{T}\right)^{-1}\left(\sum_{i=1}^n r_i a_i\right)$
  for $a \in \mathcal{A}$ **do**
    if $\left\langle a, \widehat{\theta}\right\rangle < \max_{\alpha\in\mathcal{A}}\left\langle \alpha, \widehat{\theta}\right\rangle - 2\varepsilon_i$ **then**
      Remove a from $\mathcal{A}$
    end if
  end for
end for
In the last batch, play $\operatorname{argmax}_{a\in\mathcal{A}}\left\langle a, \widehat{\theta}\right\rangle$

In the last batch, we pull the active arm with the largest dot product with the last estimated $\widehat{\theta}$. The pseudocode can be found in Algorithm 2, and the regret is bounded by the following theorem.

Theorem 4. *The regret of Algorithm 2 is at most $O\left(T^{1/B}\sqrt{dT\log(KT)}\right)$ for the batched stochastic linear bandit problem with K arms, and its running time is polynomial.*

Proof. Let n_i denote the size of batch i. Then, by Lemma 3, we have $n_i \in [cq^i, Cq^i]$. Since $q = (T/c)^{1/B}$, the number of batches is not more than B.

We define the following *good event*: "for any arm a that is active at the beginning of batch i, at the end of this batch we have $\left|\left\langle a, \widehat{\theta}_i - \theta^\star\right\rangle\right| \le \varepsilon_i$." Since there are K arms and at most T batches and $\delta = 1/KT^2$, by Lemma 3 and the union bound the good event happens with probability at least $1-1/T$. We assume it happens in the following, for if it does not happen, we can upper bound the regret by T, adding just 1 to the final regret bound, as in the proof of Theorem 1.

Since the good event happens, and because of our elimination rule (1), the triangle inequality shows the optimal arm will not be eliminated: let $a^\star$ denote the optimal arm; for any suboptimal arm a,

$$\left\langle a, \widehat{\theta}_i\right\rangle - \left\langle a^\star, \widehat{\theta}_i\right\rangle \le (\langle a, \theta^\star\rangle + \varepsilon_i) - (\langle a^\star, \theta^\star\rangle - \varepsilon_i) < 2\varepsilon_i.$$

Next, fix a suboptimal arm a, and let $\Delta := \langle a^\star - a, \theta^\star\rangle$ denote its gap. Let i be the smallest positive integer such that $\varepsilon_i < \Delta/4$. Then, since the good event happens, and because of our elimination rule (1), the triangle inequality shows this

arm will be eliminated by the end of batch i:

$$\left\langle a^\star, \widehat{\theta}_i \right\rangle - \left\langle a, \widehat{\theta}_i \right\rangle \geq (\langle a^\star, \theta^\star \rangle - \varepsilon_i) - (\langle a, \theta^\star \rangle + \varepsilon_i)$$
$$= \Delta - 2\varepsilon_i > 2\varepsilon_i.$$

Thus, during batch i, any active arm has gap at most $4\varepsilon_{i-1}$, so the instantaneous regret in any round is not more than $4\varepsilon_{i-1}$, whence the expected regret of the algorithm conditional on the good event can be bounded by:

$$\sum_{i=1}^{B} 4n_i\varepsilon_{i-1} \leq 4C \sum_{i=1}^{B} q^i \sqrt{d\log(KT^2)/q^{i-1}}$$
$$\leq 6Cq\sqrt{d\log(KT)} \sum_{i=0}^{B-1} q^{i/2}$$
$$= O\left(q\sqrt{d\log(KT)}q^{B/2}\right)$$
$$= O\left(q\sqrt{dT\log(KT)}\right),$$

completing the proof. □

Infinite Action Sets

Next, we prove a regret bound of $O\left(T^{1/B} \cdot d\sqrt{T\log T}\right)$ for batched stochastic linear bandits even if the action set $\mathcal{A}$ has infinite cardinality. An *ε-net* for $\mathcal{A}$ is a set $\mathcal{A}' \subseteq \mathcal{A}$ such that for any $a \in \mathcal{A}$ there exists some $a' \in \mathcal{A}'$ with $\|a - a'\|_2 \leq \varepsilon$. Since $\mathcal{A}$ is a subset of the unit Euclidean ball in $\mathbf{R}^d$, it has a $\frac{1}{T}$-net $\mathcal{A}'$ of cardinality not more than $(3T)^d$, see, e.g., Vershynin (2018, Corollary 4.2.13). (If $\mathcal{A}'$ does not span $\mathbf{R}^d$, we may just add d additional vectors to it so it spans $\mathbf{R}^d$, and this will not affect the asymptotic regret of the algorithm.)

We execute Algorithm 2 using the finite action set $\mathcal{A}'$. Let $a_1, \ldots, a_T$ denote the algorithms' actions. Then, by Theorem 4, we have

$$T \sup_{a\in\mathcal{A}'} \langle a, \theta^\star \rangle - \mathbf{E}\left[\sum_{t=1}^{T} \langle a_t, \theta^\star \rangle\right]$$
$$= O\left(T^{1/B}\sqrt{dT\log(|\mathcal{A}'|T)}\right)$$
$$= O\left(T^{1/B} \cdot d\sqrt{T\log T}\right).$$

On the other hand, since $\mathcal{A}'$ is a $\frac{1}{T}$-net for $\mathcal{A}$, for any $a \in \mathcal{A}$ there exists some $a' \in \mathcal{A}'$ with $\|a - a'\|_2 \leq \frac{1}{T}$, which implies

$$\langle a, \theta^\star \rangle - \langle a', \theta^\star \rangle \leq \|a - a'\|_2 \cdot \|\theta^\star\|_2 \leq \frac{1}{T},$$

and in particular,

$$\sup_{a\in\mathcal{A}} \langle a, \theta^\star \rangle - \sup_{a\in\mathcal{A}'} \langle a, \theta^\star \rangle \leq \frac{1}{T},$$

and thus,

$$\mathbf{E}[\text{Regret}] = T \sup_{a\in\mathcal{A}} \langle a, \theta^\star \rangle - \mathbf{E}\left[\sum_{t=1}^{T} \langle a_t, \theta^\star \rangle\right]$$
$$= \left(T \sup_{a\in\mathcal{A}} \langle a, \theta^\star \rangle - T \sup_{a\in\mathcal{A}'} \langle a, \theta^\star \rangle\right)$$
$$+ \left(T \sup_{a\in\mathcal{A}'} \langle a, \theta^\star \rangle - \mathbf{E}\left[\sum_{t=1}^{T} \langle a_t, \theta^\star \rangle\right]\right)$$
$$\leq 1 + O\left(T^{1/B} \cdot d\sqrt{T\log T}\right),$$

completing the proof. The running time of this algorithm is polynomial in T^d.

7 Batched Adversarial Multi-Armed Bandits

We start by proving a regret upper bound.

Lemma 5. *There is a non-adaptive algorithm for batched adversarial multi-armed bandits with regret bounded by* $\mathbf{E}[\text{Regret}] \leq O\left(\sqrt{TK + T^2\log(K)/B}\right)$.

Proof. The proof is via a reduction to the setting of sequential adversarial multi-armed bandits with delays, in which the reward received in each round is revealed to the player D rounds later. For this problem, Zimmert and Seldin (2020, Theorem 1) gave an algorithm with regret $O\left(\sqrt{KT} + \sqrt{DT\log(K)}\right)$ (see also Cesa-Bianchi, Gentile, and Mansour (2019, Corollary 15) for a slightly weaker result). Now, for the batched adversarial bandit problem, we partition the time horizon into B batches of size T/B. Thus, the reward of each pull is revealed at most $D = T/B$ rounds later, hence the above result gives an algorithm with regret bounded by $O\left(\sqrt{TK + T^2\log(K)/B}\right)$. □

We complement the above result with a nearly tight lower bound.

Lemma 6. *Any non-adaptive algorithm for batched adversarial multi-armed bandits has regret at least* $\Omega\left(\frac{T}{\sqrt{B}}\right)$.

Proof. Suppose the batch sizes are $t_1, \ldots, t_B$, which are fixed before starting the game. Consider the following example with $K = 2$ arms. For each batch, we choose a uniformly random arm and set its reward to 1 throughout the batch, and set the other arm's reward to 0. Since the expected instantaneous reward of any round given information from past is $\frac{1}{2}$, the expected reward of *any* non-adaptive algorithm is $\frac{T}{2}$.

Next we show that the expected reward of the optimal arm is $\frac{T}{2} + \Omega\left(\frac{T}{\sqrt{B}}\right)$. Let $X \coloneqq t_1R_1 + \cdots + t_BR_B$, where each R_i is -1 or $+1$ independently and uniformly at random. Then the total rewards of the two arms in our instance are distributed as $\frac{T}{2} - \frac{X}{2}$ and $\frac{T}{2} + \frac{X}{2}$. To complete the proof, we need only show that $E[|X|] = \Omega(T/\sqrt{B})$. Hölder's inequality yields

$$\left(\mathbf{E}|X|^4\right)^{1/3} \left(\mathbf{E}|X|\right)^{2/3} \geq \mathbf{E}|X|^2. \tag{2}$$

Thus, to lower bound $\mathbf{E}|X|$ we need to bound $\mathbf{E}|X|^2$ and $\mathbf{E}|X|^4$. For $\mathbf{E}|X|^2 = \mathbf{E}X^2$, observe that

$$\begin{aligned}\mathbf{E}[X^2] &= \mathbf{E}\left[\left(\sum_{i=1}^{B} t_i R_i\right)^2\right] \\ &= \mathbf{E}\left[\sum_{i=1}^{B} t_i^2 R_i^2\right] + \mathbf{E}\left[\sum_{i\neq j} t_i t_j R_i R_j\right] \\ &= \mathbf{E}\sum_{i=1}^{B} t_i^2 R_i^2 = \sum_{i=1}^{B} t_i^2,\end{aligned}$$

since $\mathbf{E}R_i = 0, \mathbf{E}R_i^2 = 1$, and R_i and R_j are independent for $i \neq j$. Similarly, after expanding $X^4 = (\sum t_i R_i)^4$, all terms with odd powers of R_i will have zero expectations, so we get $\mathbf{E}[X^4] = \sum_{i=1}^{B} t_i^4 + 6\sum_{i<j} t_i^2 t_j^2 \leq 3\left(\sum_{i=1}^{B} t_i^2\right)^2$. From (2) we get

$$\mathbf{E}[|X|] \geq \frac{\left(\mathbf{E}[X^2]\right)^{\frac{3}{2}}}{\left(\mathbf{E}[X^4]\right)^{\frac{1}{2}}} \geq \sqrt{\frac{1}{3}\sum_{i=1}^{B} t_i^2} \geq \frac{T}{\sqrt{3B}},$$

where the last inequality follows from the Cauchy-Schwarz inequality, recalling that $\sum_{i=1}^{B} t_i = T$. Hence, the expected regret is at least $\mathbf{E}|X|/2 = \Omega(T/\sqrt{B})$, completing the proof. □

We are now ready to prove the main result of this section, which is a minimax regret characterization of non-adaptive algorithms for batched adversarial multi-armed bandits.

Theorem 7. *The best achievable regret of a non-adaptive algorithm for batched adversarial multi-armed bandits is* $\widetilde{\Theta}\left(\sqrt{TK + T^2/B}\right)$.

Proof. An upper bound of $O\left(\sqrt{TK + T^2\log(K)/B}\right)$ is proved in Lemma 5. A lower bound of $\Omega(T/\sqrt{B})$ is proved in Lemma 6, while a lower bound of $\Omega(\sqrt{KT})$ holds even in the sequential setting when $B = T$ (Auer et al. 2003, Theorem 5.1). □

A Lower Bound for Adaptive Algorithms

Finally, for adaptive algorithms for batched adversarial multi-armed bandits, we show a regret lower bound of $\Omega(T/B)$.

Theorem 8. *Any adaptive algorithm for batched adversarial multi-armed bandits has regret at least* $\Omega(T/B)$.

Proof. We first prove the lower bound for non-adaptive algorithms and then extend it to adaptive algorithms.

Let $K = 2$ and consider the following reward sequences. In the beginning, both arms have 0 rewards. Then, at a round chosen uniformly at random from $\{1, \ldots, T\}$, the reward of one of the arms becomes 1 and stays 1 until the end. Hence, the expected reward of the best arm is $\frac{T}{2}$.

The switching happens inside one of the batches. The expected number of 1s that fall in that batch is half of the size of the batch, and for any strategy chosen by the player in that batch, her expected regret is at least a quarter of the size of the batch.

Denote the batch sizes by $t_1, \ldots, t_B$. The probability that the (random) switching time falls in the ith batch is $\frac{t_i}{T}$. Hence, the expected regret is at least

$$\sum_{i=1}^{B} \frac{t_i}{T}\cdot\frac{t_i}{4} = \frac{1}{4T}\cdot\sum_{i=1}^{B} t_i^2 \geq \frac{1}{4T}\cdot B\cdot\left(\frac{T}{B}\right)^2 = \frac{T}{4B},$$

completing the proof.

To extend the lower bound to adaptive algorithms, note that the defined distribution over reward sequences does not depend on the batch sizes or the algorithms' actions. Hence, the lower bound holds for *any* sequence of batch sizes, deterministic or randomized. □

8 Conclusion

We presented a systematic theoretical study of the batched bandits problem in stochastic and adversarial settings. We have shown a large contrast between the stochastic and adversarial multi-armed bandits: while in the stochastic case a logarithmic number of batches are enough to achieve the optimal regret, the adversarial case needs a polynomial number of batches. This motivates studying batched versions of models in-between stochastic and adversarial; one such model is the non-stationary model, defined next.

Starting from the stochastic model, a *non-stationary* multi-armed bandit problem is one in which the arms reward distributions may change over time, but there is a restriction on the amount of change. A natural assumption is to bound the *number of changes* in the arms' reward distributions. Let S denote the allowed number of changes (or *switches*) of the vector of reward distributions during the T rounds of the game. The case $S = 0$ corresponds to stochastic bandits, while $S = T$ corresponds to adversarial bandits. This problem has been studied in the sequential setting and various algorithms have been devised based on, e.g., UCB (Garivier and Moulines 2011) and EXP3 (Auer et al. 2003). It is natural to study the regret of non-stationary bandits in the batch mode; in particular, the construction of Theorem 8 gives a regret lower bound of $\Omega(T/B)$ for any $S > 0$; proving regret bounds for all S is an interesting avenue for further research.

For batched stochastic multi-armed bandits with two arms, Perchet et al. (2016, Theorem 2) provide a regret bound of $O\left(\left(\frac{T}{\log T}\right)^{1/B}\frac{\log T}{\Delta}\right)$. It is natural to ask whether this bound can be extended to the case $K > 2$: is there an algorithm with regret bounded by $O\left(\left(\frac{T}{\log T}\right)^{1/B}\sum_{i:\Delta_i>0}\frac{\log(T)}{\Delta_i}\right)$?

Acknowledgments

We thank Tor Lattimore for pointing us to Lattimore and Szepesvári (2020, Chapter 21). Abbas Mehrabian is supported by an IVADO-Apogée-CFREF postdoctoral fellowship. Amin Karbasi was partially supported by NSF (IIS-1845032), ONR (N00014-19-1-2406), and AFOSR (FA9550-18-1-0160).

References

Agarwal, A.; Agarwal, S.; Assadi, S.; and Khanna, S. 2017. Learning with Limited Rounds of Adaptivity: Coin Tossing, Multi-Armed Bandits, and Ranking from Pairwise Comparisons. In Kale, S.; and Shamir, O., eds., *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, 39–75. Amsterdam, Netherlands: PMLR. URL http://proceedings.mlr.press/v65/agarwal17c.html.

Agarwal, A.; Assadi, S.; and Khanna, S. 2019. Stochastic submodular cover with limited adaptivity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 323–342. SIAM.

Audibert, J.-Y.; and Bubeck, S. 2009. Minimax policies for adversarial and stochastic bandits. Conference On Learning Theory (COLT 2009).

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2): 235–256.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2003. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32(1): 48–77. ISSN 0097-5397. doi:10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

Auer, P.; and Ortner, R. 2010. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica* 61(1): 55–65.

Balkanski, E.; and Singer, Y. 2018. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 1138–1151.

Bertsimas, D.; and Mersereau, A. J. 2007. A learning approach for interactive marketing to a customer segment. *Operations Research* 55(6): 1120–1135.

Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* 5(1): 1–122.

Bubeck, S.; Cesa-Bianchi, N.; and Kakade, S. M. 2012. Towards Minimax Policies for Online Linear Optimization with Bandit Feedback. In Mannor, S.; Srebro, N.; and Williamson, R. C., eds., *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, 41.1–41.14. Edinburgh, Scotland: PMLR. URL http://proceedings.mlr.press/v23/bubeck12a.html.

Cesa-Bianchi, N.; Dekel, O.; and Shamir, O. 2013. Online Learning with Switching Costs and Other Adaptive Adversaries. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26*, 1160–1168. Curran Associates, Inc.

Cesa-Bianchi, N.; Gentile, C.; and Mansour, Y. 2019. Delay and Cooperation in Nonstochastic Bandits. *J. Mach. Learn. Res.* 20(1): 613–650. ISSN 1532-4435.

Chen, L.; Feldman, M.; and Karbasi, A. 2019. Unconstrained submodular maximization with constant adaptive complexity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 102–113.

Chen, Y.; and Krause, A. 2013. Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization. *ICML (1)* 28(160-168): 8–1.

Contal, E.; Buffoni, D.; Robicquet, A.; and Vayatis, N. 2013. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–240. Springer.

Desautels, T.; Krause, A.; and Burdick, J. W. 2014. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research* 15: 3873–3923.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* 6(Apr): 503–556.

Esfandiari, H.; Karbasi, A.; and Mirrokni, V. 2019. Adaptivity in Adaptive Submodularity. *arXiv preprint arXiv:1911.03620* .

Fahrbach, M.; Mirrokni, V.; and Zadimoghaddam, M. 2019. Submodular Maximization with Nearly Optimal Approximation, Adaptivity and Query Complexity. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms*, 255–273. doi:10.1137/1.9781611975482.17. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975482.17.

Gao, Z.; Han, Y.; Ren, Z.; and Zhou, Z. 2019. Batched Multi-armed Bandits Problem. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 501–511. Curran Associates, Inc. URL http://papers.nips.cc/paper/8341-batched-multi-armed-bandits-problem.pdf.

Garivier, A.; and Moulines, E. 2011. On Upper-Confidence Bound Policies for Switching Bandit Problems. In Kivinen, J.; Szepesvári, C.; Ukkonen, E.; and Zeugmann, T., eds., *Algorithmic Learning Theory*, 174–188. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 58: 13–30.

Hoi, S. C.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, 417–424.

Jun, K.-S.; Jamieson, K.; Nowak, R.; and Zhu, X. 2016. Top Arm Identification in Multi-Armed Bandits with Batch Arm Pulls. In Gretton, A.; and Robert, C. C., eds., *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, 139–148. Cadiz, Spain: PMLR. URL http://proceedings.mlr.press/v51/jun16.html.

Kathuria, T.; Deshpande, A.; and Kohli, P. 2016. Batched Gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems*, 4206–4214.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 453–456.

Lange, S.; Gabel, T.; and Riedmiller, M. 2012. Batch reinforcement learning. In *Reinforcement learning*, 45–73. Springer.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press. Draft available at https://torlattimore.com/downloads/book/book.pdf.

Le, H.; Voloshin, C.; and Yue, Y. 2019. Batch Policy Learning under Constraints. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 3703–3712. Long Beach, California, USA: PMLR. URL http://proceedings.mlr.press/v97/le19a.html.

Perchet, V.; Rigollet, P.; Chassang, S.; and Snowberg, E. 2016. Batched bandit problems. *Ann. Statist.* 44(2): 660–681. doi:10.1214/15-AOS1381. URL https://doi.org/10.1214/15-AOS1381. Extended abstract in COLT 2015.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* 58(5): 527–535.

Slivkins, A. 2019. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning* 12(1-2): 1–286. ISSN 1935-8237. doi:10.1561/2200000068. URL http://dx.doi.org/10.1561/2200000068.

Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; and Adams, R. 2015. Scalable Bayesian optimization using deep neural networks. In *International conference on machine learning*, 2171–2180.

Vershynin, R. 2018. *High-dimensional probability: An introduction, with applications in data science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge. ISBN 978-1-108-41519-4. doi:10.1017/9781108231596. URL https://doi.org/10.1017/9781108231596.

Wainwright, M. J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge. ISBN 978-1-108-49802-9. doi:10.1017/9781108627771. URL https://doi.org/10.1017/9781108627771.

Zimmert, J.; and Seldin, Y. 2020. An Optimal Algorithm for Adversarial Bandits with Arbitrary Delays. In Chiappa, S.; and Calandra, R., eds., *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, 3285–3294. Online: PMLR. URL http://proceedings.mlr.press/v108/zimmert20a.html.