The Impact of Batch Learning in Stochastic Linear Bandits

Danil Provodin1,2, Pratik Gajane1, Mykola Pechenizkiy1,4, Maurits Kaptein2,3

1 Eindhoven University of Technology, Eindhoven, The Netherlands
2 Jheronimus Academy of Data Science, ’s-Hertogenbosch, The Netherlands
3 Tilburg University, Tilburg, The Netherlands
4 University of Jyväskylä, Jyväskylä, Finland

Abstract—We consider a special case of bandit problems, named batched bandits, in which an agent observes batches of responses over a certain time period. Unlike previous work, we consider a more practically relevant batch-centric scenario of batch learning. That is to say, we provide a policy-agnostic regret analysis and demonstrate upper and lower bounds for the regret of a candidate policy. Our main theoretical results show that the impact of batch learning is a multiplicative factor of batch size relative to the regret of online behavior. Primarily, we study two settings of the stochastic linear bandits: bandits with finitely and infinitely many arms. While the regret bounds are the same for both settings, the former setting results hold under milder assumptions. Also, we provide a more robust result for the 2-armed bandit problem as an important insight. Finally, we demonstrate the consistency of theoretical results by conducting empirical experiments and reflect on optimal batch size choice.

Index Terms—batch learning, linear bandits

I. INTRODUCTION

The stochastic bandit problem is one of the central topics of modern literature on sequential decision making, which aims to determine policies that maximize the expected reward. These policies are often learned either online (sequentially) (see, e.g., [1, 2, 3]) or offline (statically) (see, e.g., [4, 5, 6]). In online problems, the agent learns through sequential interaction with the environment, adjusting the behavior for every single response. In offline learning, on the other hand, the agent learns from fixed historical data without the possibility to interact with the environment. Therefore, the agent’s goal is to maximally exploit the static data to determine the best policy. However, in many application domains, batched feedback is an intrinsic characteristic of the problem, and neither setting provides a close approximation of the underlying reality [7, 8, 9]. While the offline setting is not conducive to sequential learning, online learning is often curtailed by the limitations of practical applications. For example, in recommender systems, treating users one at a time can become a formidable computational burden; in online marketing, environments design (campaigns) and the presence of delayed feedback result in treating customers organized into groups. In such cases, an online approach that acts on groups of observations is much more appealing from a practical point of view. Because of the practical restrictions described above, we consider sequential batch learning in bandit problems, as it resolves computational complexity or delayed feedback issues, while retaining the sequential nature of the problem.

Withdrawing the assumptions of online learning that have dominated much of the bandit literature raises fundamental questions as how to benchmark performance of candidate policies, and how one should choose the batch size for a given policy in order to achieve the rate-optimal regret bounds. Yet, a comprehensive understanding of the effects of the batch setting is still missing. As a consequence, it is now frequently a case in practice when the batch size is chosen for the purpose of computational accessibility rather than statistical evidence [8, 9]. Moreover, while the asymptotics of batched policies is known (see, e.g., [10]), the relatively small horizon performance of batch policies is less understood while simultaneously being more practically relevant. Thus, in this work, we make a significant step in these directions by providing a systematic study of the sequential batch learning.

In this work, we focus on the batch learning problem with respect to the batch size $b$. That is to say, we provide upper and lower regret bounds of a batch policy relative to its online behavior as a function of $b$; specifically, we establish that the impact of the batch learning is a multiplicative factor of batch size. The distinctive feature of our work, relative to previous batch learning literature, is that we provide a policy-agnostic analysis, which holds for a certain set of policies.

Contribution. In summary, this paper makes the following contributions: 1) we formulate a more practically relevant batch-centric problem (Section II) and establish upper and lower bounds on the performance for an arbitrary candidate policy (Section IV); 2) we demonstrate the validity of the theoretical results experimentally and reflect on the optimal batch size choice (Section VI); 3) we provide insight and guidance to development of novel batch policies (Section VII).

Related work. Our setting lies in the intersection of batched bandits and bandits with delayed feedback. The origin of the batched bandit constituent can be traced back to [11], which proposes an explicit batched algorithm based on explore-then-commit policy for a two-armed batch bandit problem and explore its upper and lower regret bounds, giving rise to a rich line of work [12, 13, 14, 15]. However, regret bounds in prior batched bandit literature are incomparable to ours because those are with respect to the number of batches $M$, while we consider regret bounds with respect to batch size $b$. 
The problem of batched bandits also relates to learning from delayed feedback (see, e.g., [16, 17, 18]). The delayed setting focuses on similar regret bounds but with respect to the size of delay \( d \) (which is comparable with the batch size \( b \)). Similar to our work, reference [16] provides a policy-agnostic analysis of stochastic bandits. Specifically, they establish that delay increases regret in an additive way in stochastic problems. However, we provide a more general analysis of stochastic linear bandits in our study. References [17, 18] show that in linear bandits, the increase in regret due to rewards being delayed is a multiplicative \( \sqrt{d} \) and \( d \) factor, respectively. From the mathematical point of view, the main difference with our approach is that they propose a specific policy and analyze it, whereas we provide a more general policy-agnostic analysis. In that sense, their problem is easier than ours because policy behavior is completely known; yet, we make stronger assumptions on the batch size \( b \).

II. PROBLEM FORMULATION

We consider a general setting of the stochastic linear bandit problem. In this problem, the decision-maker (agent) has to make a sequence of decisions and for each decision it incurs a stochastic, although not necessarily immediately observed, reward. More formally, given a decision set \( \mathcal{A} \subset \mathbb{R}^d \) (for \( d < \infty \)) and a time horizon \( n < \infty \), at each time step \( t \in \{1, 2, \ldots, n\} \), the agent chooses an action \( A_t \in \mathcal{A} \) and reveals reward \( X_t = (\theta_t, A_t) + \eta_t \) where \( \eta_t \) is the noise, and \( \theta_t \) is the instance unknown for the agent. We assume each \( \eta_t \) is 1-subGaussian: \( \mathbb{E}[e^{\lambda \eta_t}] \leq e^{\lambda^2/2}, \forall t \in \{1, n\}, \forall \lambda \in \mathbb{R} \). Although restrictive, this is a classical assumption for bandit literature, as it covers most reasonable noise distributions. Further, without loss of generality, we assume \( \|\theta_t\|_2 \leq 1 \).

The goal of the agent is to maximize the total reward \( S_n = \sum_{t=1}^n X_t \). To assess the performance of a policy \( \pi \), we consider regret – difference between the agent’s total reward and cumulative reward obtained by an optimal policy:

\[
R_n(\pi) = \mathbb{E}\left[ \sum_{t=1}^n \max_{a \in \mathcal{A}_t}(\theta_t, a) - S_n \right].
\]

In online learning, the decision maker immediately observes reward \( X_t \) after selecting action \( A_t \) at timestep \( t \). Consequently, in selecting \( A_{t+1} \), the decision maker can base his decision on past history: \( H_t = (A_1, X_1, \ldots, A_t, X_t) \in \mathbb{R}^{(d+1)t} = H_t \). Note that \( H_0 = \emptyset \). Let \( \mathcal{M}_1(X) \) be a set of all probability distributions over a set \( X \). As such, a policy is a finite sequence \( \pi = (\pi_t)_{t \leq \theta} \) of maps of histories to distributions over actions (decision rules), formally, \( \pi_t : H_{t-1} \to \mathcal{M}_1(A_t) \).

In contrast to online learning, our setting assumes only that rewards are released at specific predefined timesteps. Denote by \( T = t_1, \ldots, t_M \) a grid, which is a division of the time horizon \( n \) to \( M \) batches of equal size \( b, 1 = t_1 < \ldots < t_M = n, t_j - t_{j-1} = b \) for all \( j = 1, \ldots, M \). We assume that \( n = bM \), otherwise we can take \( n := \left\lceil \frac{n}{b} \right\rceil b \). Similarly, we define a batch policy as a finite sequence of \( \pi = (\pi_t)_t \leq n \). However, not the whole past history is available for the agent in timestep \( t \), formally, \( H_t = H_{t_j} \) for any \( t_j < t < t_{j+1} \). To distinguish between online and batch policies we denote the last as \( \pi^b = (\pi^b_t)_{t \leq n} \). Thus, given an arbitrary policy \( \pi \), we aim to establish upper and lower regret bounds of its batch specification \( \pi^b \).

III. PRELIMINARIES

Before proceeding, we will need to distinguish between “good” and “bad” policies on the basis of some properties. We first define a binary relation on a set of policies. We say the decision rule \( \pi_t = \pi_t(\cdot|H_{t-1}), \pi_t(\cdot|H_{t-1}) = \mathbb{P}(A_t = \cdot|H_{t-1}) \), is not worse than the decision rule \( \pi'_t = \pi'_t(\cdot|H_{t-1}) \) and write \( \pi_t \succeq \pi'_t \) if the expected reward under \( \pi_t \) is not less than the expected reward under \( \pi'_t \):

\[
\sum_{a \in \mathcal{A}_t}(\theta_t, a)\pi_t(a|H_{t-1}) \geq \sum_{a \in \mathcal{A}_t}(\theta_t, a)\pi'_t(a|H_{t-1}). \tag{1}
\]

If \( \pi_t \succeq \pi'_t \) can be replaced by >, we say that the decision rule \( \pi_t \) is better than the decision rule \( \pi'_t \) (and write \( \pi_t \succ \pi'_t \)).

Define the informativeness of history by the number of times the optimal actions were chosen in it. Formally, let \( T_{a^*}(\pi, t) = \sum_{t'=1}^t 1\{A_t = a^*_t\} \) be the number of times policy \( \pi \) made optimal decisions in history \( H_t \). Then, we require that the decision rule based on a more informative history is at least as good as the decision rule based on a less informative history:

Assumption III.1 (Informativeness). Let \( T_{a^*}(\pi, t) \) and \( T_{a^*_t}(\pi, t) \) be numbers of times the optimal actions were chosen in histories \( H_t \) and \( H'_t \), correspondingly. If \( T_{a^*}(\pi, t) \geq T_{a^*_t}(\pi, t) \), then \( \pi_{t+1}(\cdot|H_t) \geq \pi_{t+1}(\cdot|H'_t) \).

Next, we assume that policy \( \pi = (\pi_t)_{t \leq n} \) improves over time if the “rate” of increasing of the regret decreases.

Assumption III.2 (Subliniarity). \( \frac{R_{a^*}(\pi)}{n_1} > \frac{R_{a^*_t}(\pi)}{n_2} \) for all \( n_1, n_2, 1 \leq n_1 < n_2 \leq n \).

Finally, we impose a monotonic lower bound on the probability of choosing the optimal action at timestep \( t \). We consider two assumptions: instance-independent monotonicity and instance-dependent. In the former case, we assume the existence of the universal lower bound \( f(t) \), regardless of instance \( \theta \); in the latter case – the existence of the lower bound depends on a specific instance of actions \( \theta \); \( f := f_\theta \).

Assumption III.3 (Instance-independent monotonicity). For any suboptimal action \( a, a \in \mathcal{A} \) and \( a \notin \arg \max_{a \in \mathcal{A}}(\theta, a) \), there exists a function \( f_a : [0, \infty) \to [0, 1] \) such that: (i) \( f_a \) is nonincreasing; (ii) \( \pi_t(a|H_{t-1}) \leq f_a(t) \) for all \( t > 0 \); and (iii) \( f(t) := 1 - \sum_a f_a(t) \) is a strictly increasing function for all \( t > 0 \) unless \( f(t) = 1 \).

Assumption III.4 (Instance-dependent monotonicity). For any suboptimal action \( a, a \in \mathcal{A} \) and \( a \notin \arg \max_{a \in \mathcal{A}}(\theta, a) \), there exists a function \( f_{a\theta} : [0, \infty) \to [0, 1] \), depending on \( \theta \), such that: (i), (ii), (iii) from Assumption III.3 hold; and (iv) \( f_{a\theta}(t) \) is nondecreasing in its instance argument in the following sense: \( f_{a\theta}(t) < f_{a\theta}(t) \) for all \( t > 0 \) if \( \min_{a \in \mathcal{A}} \Delta_0(a) < \min_{a \in \mathcal{A}} \Delta_0(a) \), where \( \Delta_0(a) = \max_{b \in \mathcal{A}}(\theta, b - a) \).

IV. BATCH LEARNING FOR STOCHASTIC LINEAR BANDITS

In this section, we provide lower and upper bounds on the best achievable performance for different linear bandit settings.
A. Stochastic linear bandits with 2 arms

We start with a more restricted analysis of 2-armed problem as it allows to derive a stronger result.

**Theorem IV.1.** Let $\pi^b$ be a batch specification of a given policy $\pi$, $K = 2$, and $M = \frac{H}{2}$. Suppose that assumptions III.1 and III.2 hold. Then, for $b > 1$,

$$R_n(\pi) < R_n(\pi^b) \leq bR_M(\pi).$$  \hspace{1cm} (2)

**Proof.** Let a naive agent (i) deliberately repeats each step $b$ times (instead of immediately updating its beliefs); (ii) after $b$ repetitions (after a batch ends), it updates its beliefs using only the first reward from the previous batch. Such policy would then constitute $bR_M(\pi)$ regret. For notational simplicity, we refer to this policy as online “short” policy (denote it by $\pi^c$).

**Step 1 (Within batch).** Fix $j \geq 1$. Let $\tau_{i,j} \geq \tau^b_{i,j} \geq \tau_{i,j}$. Define an average decision rule between timesteps $t_1$ and $t_2$ as $\bar{\pi}_{t_1,t_2} = \frac{1}{t_2-t_1+1} \sum_{t=t_1}^{t_2} \pi^c_t$; and an average decision rule in batch $j$ as $\bar{\pi}_j = \frac{1}{\tau_{j-1}+1} \sum_{t=\tau_{j-1}+1}^{\tau_j} \pi^c_t$. By the end of batch $j$, we have $\bar{\pi}_{t_1,t} = \max_{a,b} \{ \alpha_{t}(a,b) \}$ for any timestep $t < t < t+1$, where $a$ follows from Lemma V.1 (1); and (b) and (c) hold by the definition of batch policy. Thus, starting with $\pi_{t_1} \geq \pi^b_{t_1} \geq \pi_{t_1}$ at the beginning of batch $j$ leads us to $\tau_{j} > \bar{\pi}_{j} \geq \pi_{j}$. Moreover, by Lemma V.1 (2), we have $\pi_{t_1+1} = \pi^b_{t_1+1} \geq \pi_{t_1+1}$.

**Step 2 (Between batches).** Fix $j \geq 1$. Let $\bar{\pi}_j > \bar{\pi}^b_{j} \geq \bar{\pi}_{j}$ for any batch $1 \leq l < j$. Let $H_{l-1}, H_{l-1}, H_{l-1}$ be histories collected by policies $\pi, \pi^b, \pi^l$ by the timestep $t_j$, correspondingly. Let $a^* = \arg \max_a \{ \theta_s, a \}$ be an optimal action. Define a number of times we received reward from action $a$ in batch $j$ by policy $\pi$ as $T'_n(\pi, j) = \sum_{t=\tau_{j-1}+1}^{\tau_j} \mathbb{I}\{A_t = a\}$. Note that $E[T'_n(\pi,j)] = (t_j - \tau_{j-1}) \pi^c_1(a)$. Since $K = 2$, $\bar{\pi}_j > \bar{\pi}^b_{j} \geq \bar{\pi}_{j}$ implies $\bar{\pi}_j(a^*) > \bar{\pi}^b_j(a^*) \geq \bar{\pi}_j(a^*)$, $1 \leq l < j$. Hence, $E[T'_n(\pi,j)] \geq E[T'_n(\pi^b,j)] \geq E[T'_n(\pi^l,j)] \geq \sum_t E[T'_n(\pi^l,t)]$. By applying Assumption III.1, we have that $\pi_{t_1} \geq \pi^b_{t_1} \geq \pi_{t_1}$.

**Step 3 (Regret throughout the horizon).** We assume the interaction begins with $\pi_1 = \pi^b_1 = \pi^l_1$. Then, from Step 1, by the end of the first batch, we have $\pi_1 > \pi^b_1 \geq \pi_1$ and $\pi_{t_2-1} > \pi^b_{t_2-1} \geq \pi_{t_2-1}$; and so on. Finally, summing over $M = \frac{H}{2}$ batches, we have:

$$R_n(\pi) = E \left[ \sum_{t=1}^{n} (\theta_s - A_t) - S_n \right] = E \left[ \sum_t (\theta_s - A_t) \right] = \sum_{t=1}^{M} E \left[ \theta_s - A_t \right] + E \left[ \sum_{t=1}^{M} (\theta_s - A_t) \bar{\pi}_j(a) \right],$$

and, using $\bar{\pi}_j < \bar{\pi}^b_j \leq \bar{\pi}_j$, we conclude the proof. \hfill \Box

**B. Stochastic linear bandits with finitely many arms**

In the case of $K = 2$, it immediately follows that the more often a decision rule chooses the optimal action, the better it is. In contrast, this is not generally true when $K > 2$. Indeed, some decision rules might value the optimal and the worst actions so that another decision rule that puts less weight on optimal action is better (because it chooses interrim actions more often at the expense of worst action). Consequently, the case of $K > 2$ requires additional steps in the analysis. Specifically, we utilize Assumption III.4 and introduce a meta-algorithm to derive result similar to Theorem IV.1.

**Algorithm I Approximate learning with delayed start**

| Input: horizon $n \geq 0$, candidate policy $\pi$, number of actions $K$, monotonic lower bound $f$, confidence level $\delta$ |
| Initialization: $\pi^0 \leftarrow \text{Unif}(1/K)$; $t \leftarrow 1$; $H_t \leftarrow 0$ |
| repeat |
| $A_t \leftarrow \pi^0$ |
| $H_t \leftarrow \text{UpdateHistory}$ |
| $t \leftarrow t + 1$ |
| until (CHECKPHASE($H_t$, $\delta$) is False and $t \not\in T$) or $t \leq n$ |
| repeat |
| $A_t \leftarrow \pi(1/H_{t-1})$ |
| $H_t \leftarrow \text{UpdateHistory}$ |
| $t \leftarrow t + 1$ |
| until $t \leq n$ |

From Assumption III.4, it follows that after timestep $\tau_{k_0} = \min\{t : f_{\hat{\theta}}(t) > 1/K\}$ policy $\pi$ in the worst-case scenario: (i) behaves better than uniform random policy; and (ii) acts more optimally within each consequent timestep. As such, while batch and online “short” policies ($\pi^b$ and $\pi^l$) remain the same, online policy $\pi$ gets better in the worst case and, therefore, chooses optimal action more often. Thus, it would be enough to split the horizon $n$ into two phases: before and after timestep $\tau_{k_0}$, and ensure that all three policy specifications behave similarly during the first phase, e.g., some naive policy $\pi^0$ operates independently of history $H_t$ during phase 1.

However, $\tau_{k_0}$ depends on vector of rewards $\theta_s$, which is unknown for the agent. But what if one could find an estimate $\hat{\pi}$ of true $\tau_{k_0}$ such that $f_{\hat{\theta}}(\hat{\pi})$ is greater than $1/K$ with high probability for some confidence level $\delta$ (i.e., $P(f_{\hat{\theta}}(\hat{\pi}) > 1/K) > 1 - \delta$)? Then we could split the original horizon $n$ into two parts (phases) and: (i) run a uniform random policy in phase 1 (before timestep $\tau$); and (ii) apply policy $\pi$ in phase 2 (after timestep $\tau$). Algorithm 1 illustrates this intuition.

**Theorem IV.2.** Let $\pi^b$ be a batch specification of a given policy $\pi$, $K < \infty$, $M = \frac{H}{2}$, and CHECKPHASE be with the failure probability $\delta$ (Theorem IV.3). Suppose that assumptions III.1, III.2, III.4 hold. Assume that policy $\pi$ represents a policy constructed by Algorithm 1 for policy $\pi$. Then, for $b > 1$,

$$R_n(\hat{\pi}) < R_n(\hat{\pi}) \leq bR_M(\hat{\pi})$$

with probability $1 - \delta$, where $R_n(\pi)$ is the worst-case regret.

**Proof.** First, we split total regret in two terms and notice that, since uniform random policy $\pi^b$ operates during the first phase, the first term is the same for both policies ($\pi$ and $\hat{\pi}$).

$$R_n(\hat{\pi}) = R_{1:n-1}(\pi^0) + R_{\tau:n}(\pi),$$

$$R_n(\pi) = R_{1:n}(\pi^0) + R_{\tau:n}(\pi),$$

where $R_{1:n}(\pi) = E \left[ \sum_{t=1}^{n} \max_{a \in A_t} (\theta_s, a) - S_n \right]$. Note that phase 1 can only end with the end of the batch and therefore:

$$R_n(\hat{\pi}) = R_{1:n-1}(\pi^0) + bR_{\tau:n}(\pi) = R_{1:n}(\pi^0) + bR_{\tau:n}(\pi) = bR_M(\pi).$$
Next, we express the second term as a sum of instantaneous regrets and exploit Assumption III.4:

\[ R_{\tau,n}(\pi) = E(\sum_{t} (\theta_\ast, A_\ast - A_t)) = \sum_{t,a} (\theta_\ast, A_\ast - a) \pi_t(a) H_{t-1} \leq \sum_{t,a} (\theta_\ast, A_\ast - a) f_{\theta,a}(t) = R_{\tau,n}(\pi). \]

Next, we replicate Theorem IV.1 for \( R_{\tau,n}(\pi^b) \).

**Step 1 (Within batch).** See Step 1 from Theorem IV.1.

**Step 2 (Between batches).** Fix \( j \geq 1 \). Let \( \pi_{t_1} > \pi_{t_2}^j \geq \pi_t^j \) for any batch \( 1 \leq l < j \). Let \( H_{t_1-1}, H_{t_2-1}, H_{t-1} \) be histories collected by policies \( \pi, \pi^b, \pi^t \) by the timestep \( t_j \), correspondingly. Let \( A_t^j = \arg \max A_t(\theta_a, a) \) be an optimal action for round \( t_j \) and, unlike Theorem IV.1, denote \( T_{\theta,a}(\pi,j) = \sum_{t_1} \mathbb{1}\{A_t = A_t^j\} \) - number of times policy \( \pi \) made optimal choice in batch \( j \).

Recall that \( f_\theta \) is strictly increasing with probability \( 1 - \delta \) for all \( t > \tilde{r} \). As a consequence, using the fact that: (i) \( f_{\theta,a} \) is nonincreasing for any suboptimal action \( a \); (ii) \( f_\theta \) is increasing with high probability, \( \pi_t \geq \pi_t^j \) implies \( \pi_t(A_t) > \pi_t^j(A_t) \geq \pi_t^j(A_t^j) \) for \( 1 \leq l < j \) and for \( 0 < t < t_j \) with probability \( 1 - \delta \). Hence, \( E[T_{\theta,a}(\pi,j)] = E[T_{\theta,a}(\pi,j)] \leq E[T_{\theta,a}(\pi,j)] \) for \( 1 \leq l < j \) and, therefore, \( \sum \mathbb{E}_{T_{\theta,a}(\pi,j)} = \sum \mathbb{E}_{T_{\theta,a}(\pi,j)} \leq \sum \mathbb{E}_{T_{\theta,a}(\pi,j)} \).

By applying Assumption III.1, we have that \( \pi_{t_1} > \pi_{t_2}^j \geq \pi_t^j \).

**Step 3 (Regret throughout the horizon).** Applying Step 3 from Theorem IV.1 to \( R_{\tau,n}(\pi), R_{\tau,n}(\pi^b), \) and \( R_{\tau,n}(\pi^t) \) gives:

\[ R_{\tau,n}(\pi) < R_{\tau,n}(\pi^b) \leq b R_{\tau,n}(\pi^t). \] (5)

Putting (3)-(4) and (5) together, with probability \( 1 - \delta \), we get:

\[ \bar{R}_n(\bar{\pi}) := R_{\bar{\tau},n}(\pi^b) + R_{\tau,n}(\pi) \leq R_{\bar{\tau},n}(\pi^b) = b \bar{R}_n(\bar{\pi}). \]

\[ \check{R}_n(\check{\pi}) := R_{\check{\tau},n}(\pi^b) + R_{\tau,n}(\pi) \leq R_{\check{\tau},n}(\pi^b) = b \check{R}_n(\check{\pi}). \]

**CHECKPHASE procedure.** Since function \( f_\theta \) decreases in its instance argument in a certain sense (point (iv), Assumption III.4), it suffices to find a more “difficult” environment with instance \( \bar{\theta} \) (and corresponding \( \bar{r} := \min\{t : f_\theta(t) > 1/K\} \), such that \( \mathbb{P}(f_{\theta,a}(\bar{\theta}) > 1/K) > 1 - \delta \). Intuitively, the smaller the difference between a suboptimal and the optimal actions \( (\Delta_\theta(a) = \max_{b \in A_t(\theta, b) - \theta}) \), the more difficult it is to distinguish the optimal action and, therefore, the probability of choosing the optimal action should be smaller. Imagine now that after timestep \( t \), we have some estimate \( \bar{\theta}_t \) of the unknown parameter vector \( \theta \), and confidence set \( C_t \) that contains \( \theta_t \), with high probability. Then, we can choose \( \bar{\theta} \) that underestimates the true reward along the current best action, \( \check{\theta} \), and overestimates the true rewards along all other actions, making the minimal suboptimal gap smaller. Algorithm 2 formalizes this logic.

**Theorem IV.3.** CHECKPHASE procedure presented in Algorithm 2 is with the failure probability \( \delta \).

Proof. Since \( K < \infty \), we can compose an auxiliary reward vector \( (\mu_i)_{i \leq K} \), where \( \mu_i = (\theta_i, a_i) \) is true reward of action \( a_i \). We consider the case \( K = 2 \) first and assume that action 1 is optimal. Following Algorithm 2, at timestep \( t \) the agent computes estimates \( \hat{\mu}_i(t) \) and their confidence intervals \( LC\hat{B}_{\theta,a} = \hat{\mu}_a(t) - \sqrt{n/4t} \) for \( l \neq i \) do

\[ UCB\hat{B}_{\theta,a} = \hat{\mu}_a(t) + \sqrt{n/4t} \]

end for

\( \theta = \{LC\hat{B}_{\theta,a} \cup_{\delta_i(a)} UCB\hat{B}_{\theta,a}\} \)

if \( f_\theta(t) > 1/K \) and \( 2K/\delta^2 < \delta \) then return False

end if

return True

Algorithm 2 CHECKPHASE

Input: history \( H_t \), confidence level \( \delta \)

\[ i = \arg \max_j \hat{\mu}_i(t), l = \arg \max_{\delta_i(a)} \hat{\mu}_j(t) \]

for \( l \neq i \) do

\[ \text{for } \theta = \{LC\hat{B}_{\theta,a} \cup_{\delta_i(a)} UCB\hat{B}_{\theta,a}\} \]

if \( f_\theta(t) > 1/K \) and \( 2K/\delta^2 < \delta \) then return False

end if

return True

C. Stochastic linear bandits with infinitely many arms

We now provide analysis for the most general setting - stochastic linear bandit problem with \( A_t \subset \mathbb{R}^d \). Intuitively, if \( |A_t| = \infty \) (e.g., \( A_t = \{a : ||a||_2 \leq t\} \), there is a unique optimal action \( A_t^\ast \) for each timestep \( t \) and any instance \( \theta_t \), but no second best action. In that sense, each environment instance would be of the same “difficulty” for the agent. As such, we leverage a more strict assumption of monotonicity (Assumption III.3) and provide instance-independent analysis.

In this case, we do not need to estimate \( \tau \), as it is given by stronger Assumption III.3. Thus, it is enough to adjust Algorithm 1 by performing phase 1 (the first loop) until \( t < \min\{\tau, n\} \) and replacing the random uniform policy \( \pi_0 \) with some naive policy to obtain the equivalent result.

**Theorem IV.4.** Let \( \pi^b \) be a batch specification of a given policy \( \pi \), \( M = \frac{n}{\pi} \). Suppose that Assumptions III.1, III.2, III.3...
hold. Assume that policy \( \tilde{\pi} \) represents a policy constructed by adjusted Algorithm 1 for policy \( \pi \). Then, for \( b > 1 \),
\[
R_n(\tilde{\pi}) < R_n(\tilde{\pi}^b) \leq b R_M(\tilde{\pi}),
\]
where \( R_n(\pi) \) is the worst case-regret.

**Proof.** The proof of the theorem immediately follows from the Theorem IV.2 by replacing \( \tilde{\pi} \) and \( \tilde{f}_0 \) with \( \gamma \) and \( f \). \( \square \)

V. SUPPLEMENTARY DISCUSSIONS

A. Intuition behind Assumption III.1

Policies that satisfy Assumption III.1 are good in leveraging exploration and exploitation decisions in the following sense: 1) if the policy gets to a situation when the optimal action has a smaller confidence interval on average (i.e., it chose the optimal action more often), and the past choices were reasonable (i.e., history \( H_t \) brought either the highest reward or the most information), then this policy will make a better consequent decision; 2) if another history \( H'_t \) were fed to the policy, that would imply not optimal leveraging of exploration and exploitation decisions and, as a result, lead to (i) fewer choices of optimal actions (or higher uncertainty); and (ii) worse subsequent decisions. In other words, a policy with history \( H_t \) will be ahead of any different history \( H'_t \) from exploration-exploitation perspective. Together with Assumption III.2 that leads to better subsequent decisions.

B. Intuition behind Assumption III.2

**Lemma V.1.** Let \( \pi = (\pi_t)_{1 \leq t \leq n} \) be a policy such that Assumption III.2 holds. Then, 1) \( \pi_n > \pi_n \), where \( \pi_t = \frac{\sum_{a \in A_t} \pi_t(a)}{t} \) is an average decision rule; 2) \( \pi_t > \pi_t \forall t \) such that \( 1 \leq t \leq n \), where \( \pi_t + \pi_s \) is an elementwise addition of two probability vectors for some \( t, s \).

**Proof.** 1. First, we need to show that \( \pi_t \) is a decision rule for some \( t \), i.e., \( \sum_{a \in A_t} \pi_t(a) = 1 \) and \( \pi_t(a) \geq 0 \) for all \( a \in A_t \). Indeed, \( \sum_{a \in A_t} \pi_t(a) = \frac{\sum_{i=1}^{\infty} \sum_{a \in A_t} \pi_t(a)}{t} = \frac{t}{t} = 1 \).

Since \( \pi_t(a) \geq 0 \) for all \( a \in A_t \) and for all \( 1 \leq s \leq t \), \( \pi_t(a) \geq 0 \) for all \( a \in A_t \). Next, we convert \( \mathbb{E}[S_n] \) into the sum over times steps and actions:
\[
\mathbb{E}[S_n] = \mathbb{E} \left[ \sum_{t,a} (\theta_t, a) I\{A_t = a\} \right] = \sum_{t,a} (\theta_t, a) \pi_t(a) [H_{t-1}]
\]

Fix \( n_1, n_2 : n_1 < n_2 \). From Assumption III.2, \( \frac{R_n(a)}{n_1} > R_n(a) \), and expressing the regret by its definition we get
\[
\mathbb{E} \left[ \sum_{t=1}^{n_1} (\theta_t, a^*) - S_{n_1} \right] > \mathbb{E} \left[ \sum_{t=n_1+1}^{n_2} (\theta_t, a^*) - S_{n_2} \right]
\]
and hence \( \mathbb{E}[S_{n_1}] - \mathbb{E}[S_{n_1}] > 0 \). Finally, \( \mathbb{E}[S_{n_2}] - \mathbb{E}[S_{n_1}] = \sum_{t=1}^{n_2} \sum_{a} (\theta_t, a) \pi_t(a) [H_{t-1}] > 0 \).

The result is completed by rearranging the sums and using the definition of \( \pi_t \), \( \pi_s \).

2. For \( t < n \) we have \( R_n(\pi) = \frac{R_{n+1}(\pi)}{t+1} \). By subtracting \( \frac{R_{n+1}(\pi)}{t+1} \) from both sides we get \( \frac{R_{n}(\pi)}{t} > \frac{R_{n+1}(\pi)}{t+1} \). By rearranging the terms, we can rewrite this inequality as \( (\theta_t, a^*) - X_{t+1} < \frac{R_{n+1}(\pi)}{t+1} \). Using the fact that \( \sum_{a \in A_t} \sum_{i=1}^{t+1} \pi_t(a) ) = \sum_{a \in A_t} \pi_{t+1}(a) \), we conclude that \( \sum_{a} (\theta_t, a) \pi_{t+1}(a) > \sum_{a} (\theta_t, a) \pi_{t+1}(a) \).

In what follows, we show that Assumption III.2 is essential because if it does not hold, the lower bounds in Theorems IV.1, IV.4, IV.2 is greater than the upper bound: \( R_n(\pi) > b R_M(\pi) \).

**Lemma V.2.** Suppose that the opposite to Assumption III.2 holds for a given policy \( \pi \). Then \( R_n(\pi) > b R_M(\pi) \).

**Proof.** Suppose Assumption III.2 does not hold for \( \pi \). Then, an online “short” policy could perform better as it omits suboptimal choices. Indeed, using Assumption III.2 for horizons \( M \) and \( n \), we have \( \frac{R_M(\pi)}{n} \geq \frac{R_M(\pi)}{n} \). Therefore, multiplying by \( n \) and \( M \) respectively, using \( n = Mb \), we get \( b R_M(\pi) > M R_n(\pi) \). Dividing by \( M \), we get the result. \( \square \)

C. Equivalence of Assumptions III.2 and III.3

There is a strong connection between Assumptions III.2 and III.3: the strictly increasing lower bound on the probability of choosing the optimal action (Assumption III.3) is equivalent to the decreasing rate of the worst-case total regret (adjusted Assumption III.2). Moreover, these assumptions are equivalent under certain conditions, summarized in the following lemma.

**Lemma V.3.** Consider a stochastic bandit problem with \( K = 2 \) arms, then Assumptions III.2 and III.3 are equivalent.

**Proof.** Recall that in 2-armed stochastic bandit \( \theta_t = (\mu_0, \mu_1) \) - a vector of true rewards for each action (arm). Assume that arm 1 is optimal. Define a number of times policy \( \pi \) played an arm \( b \) by timestep \( t \) as \( T_b(t) = \sum_{s=1}^{t} I\{A_s = 1\} \).

Using the regret decomposition lemma (Lemma 4.5, [19]), we can derive: \( R_n(\pi) > R_n(\pi) \). From frequentist probability perspective, \( \mathbb{E}[T_2(n)] \) can be interpreted as probability of choosing arm 2: \( \mathbb{P}(A_{n+1} = 2) \). As such, \( f_2(t) := \mathbb{P}(A_{n+1} = 2) \) is a decreasing function, \( f(t) := 1 - f_2(t) \) is an increasing function, and \( \pi_t(2H_{t-1}) \leq f_2(t) \).

Thus, all the conditions of Assumption III.3 are satisfied. \( \square \)

D. Illustration of Assumptions IV.3

**Lemma V.4** (Lemma 1.2, [20]). Let \( T_n(a) \) be the number action \( a \) is chosen by UCB algorithm run on instance \( \theta_* = (\mu_1, ..., \mu_K) \) of the stochastic multi-armed bandit problem and \( \Delta_a = \max \mu_a - \mu_a \). Then, for any action \( a \neq \arg \max \mu_a \),
\[
\mathbb{E}[T_n(a)] \leq \frac{4 \ln t}{\Delta_a^2} + 8.
\]

From frequentist probability perspective, we can think of \( \mathbb{E}[T_n(a)] \) as probability of choosing arm \( a \) and, thus, we can set \( f_{\theta_*}(a) = \frac{4 \ln (t+1)}{\Delta_a^2} + 8/t \) which is a nondecreasing upper bound of the probability of choosing suboptimal arm \( a \) (i.e., points (i) and (ii) hold). Next, define \( f_{\theta_0}(a) = 1 - \Delta_a \arg \max \mu_a, f_{\theta_0}(a) \). Note that \( f_{\theta_0}(a) \) is decreasing function for all \( a \) and all \( t \) and \( f_{\theta_0}(a) \) is increasing in the instance argument. Thus, we get a monotonically lower bound of the probability of choosing the optimal arm.
VI. EMPIRICAL ANALYSIS

Theoretical results posit that meta-algorithms are run optimally, i.e., utilizing monotonic lower bound \( f \), which can be indeed achieved (see Sections V-A - V-D). However, in experiments it suffices to use a general batch learning specification described in Section II. In fact, the analysis we conduct in this section is even stronger as the bounds depicted in figure 1 hold for actual regret rather than worst-case regret.

We perform experiments on simulated stochastic bandits and a contextual bandit. \(^3\) We examine the effect of batch learning on Thompson Sampling (TS) and Upper Confidence Bound (UCB) policies for stochastic problems; linear TS (LinTS) [8] and linear UCB (LinUCB) [1] for the contextual problem.

We present simulation results for the Bernoulli bandit problem, where the best action has a reward probability of 0.5 and \( K - 1 \) have a probability of 0.5 - \( \Delta \). In total, we consider six environments for \( K \in \{2, 5, 10\} \) and \( \Delta \in \{0.1, 0.02\} \). Figures 1(a) - 1(d) show regret representing the effect of batch size across three dimensions: number of action \( K \), suboptimal gap value \( \Delta \), and policy. Regret has an upward trend as batch size increases for all settings. Taking a look at the environment parameters, \( K \) and \( \Delta \), we see that the higher the suboptimality gap or number of actions, the stronger the impact of batching.

We consider batch learning in a marketing campaign on the logged dataset from our industrial partner. In the current dataset, a selected set of customers received one of the three campaigns randomly. The data contains a sample of a campaign selection combined with customer information. We adopt an unbiased offline evaluation method [21] to compare various bandit algorithms and batch size values. We use conversion rate (CR) as the metric of interest, defined as the ratio between the number of successful interactions and the total number of interactions. To protect business-sensitive data, we only report relative conversion rate; therefore, Figures 1(e)-1(f) demonstrate the CR returned by the off-policy evaluation algorithm relatively to online performance.

It is important to note that both experiments demonstrate results consistent with theoretical analysis conducted in Sections IV. As the upper bound in Theorem IV.1 suggests, the performance metric (i) reacts evenly to the increasing/decreasing batch size and (ii) doesn’t violate the imposed bounds.

We presented a new perspective of batched bandits emphasizing the importance of the batch size effect. We showed the actual effect of batch learning for various bandit settings (which is linear in the worst case), and confirmed these results empirically. Practically speaking, we have investigated one component of the performance-computational cost trade-off and demonstrated that it deteriorates gradually depending on the batch size. Thus, practitioners should choose the optimal batch size based on computational capabilities.

VII. CONCLUSION

REFERENCES

[1] L. Li, W. Chu, J. Langford, and R. E. Schapire, “A contextual-bandit approach to personalized news article recommendation,” WWW ’10.
[2] K.-S. Jun, A. Bhargava, R. Nowak, and R. Willett, “Scalable generalized linear bandits: Online computation and hashing,” in NeurIPS, 2017.
[3] M. Dimakopoulou, Z. Zhou, S. Athey, and G. Imbens, “Estimation considerations in contextual bandits,” 2018.
[4] A. Swaminathan and T. Joachims, “Counterfactual risk minimization: Learning from logged bandit feedback,” 2015.
[5] Z. Zhou, S. Athey, and S. Wager, “Offline multi-action policy learning: Generalization and optimization,” 2018.
[6] S. Athey and S. Wager, “Policy learning with observational data,” 2020.
[7] D. Bertsimas and A. Mersereau, “A learning approach for interactive marketing to a customer segment,” Operations Research, 2007.
[8] O. Chapelle and L. Li, “An empirical evaluation of Thompson sampling,” in Advances in Neural Information Processing Systems, 2011.
[9] D. N. Hill, H. Nassif, Y. Liu, A. Iyer, and S. Vishwanathan, “An efficient bandit algorithm for real-time multivariate optimization.” ACM, 2017.
[10] N. Cesa-Bianchi, O. Dekel, and O. Shamir, “Online learning with switching costs and other adaptive adversaries,” in NeurIPS, 2013.
[11] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg, “Batched bandit problems,” The Annals of Statistics, 2016.
[12] Z. Gao, Y. Han, Z. Ren, and Z. Zhou, “Batched multi-armed bandits problem,” in Advances in Neural Information Processing Systems, 2019.
[13] Y. Han, Z. Zhou, Z. Zhou, J. Blanchet, P. W. Glynn, and Y. Ye, “Sequential batch learning in finite-action linear contextual bandits,” 2020.
[14] H. Esfandiari, A. Karbasi, A. Mehrabian, and V. Mirrokni, “Regret bounds for batched bandits,” Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
[15] C. Kalkani and A. Özgür, “Batched thompson sampling,” 2021.
[16] P. Joulin, A. György, and C. Szepesvari, “Online learning under delayed feedback,” in ICML, 2013.
[17] Z. Zhou, R. Xu, and J. Blanchet, “Learning in generalized linear contextual bandits with stochastic delays,” in NeurIPS, 2019.
[18] C. Vernade, A. Carpenter, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner, “Linear bandits with stochastic delayed feedback,” in ICML 2020, 2020.
[19] T. Lattimore and C. Szepesvári, Bandit Algorithms. Cambridge University Press, 2020.
[20] S. Agrawal, K. Stratos, and L. J. Sun, “Lecture: Ucb algorithm,” https://seor100.github.io/mab/Lecture%203.pdf, 2016.
[21] L. Li, W. Chu, J. Langford, and X. Wang, “Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms,” WSDM ’11, 2011.