Linear Bandits with Stochastic Delayed Feedback

Claire Vernade [1]   **Alexandra Carpentier** [2]   **Tor Lattimore** [1]   **Giovanni Zappella** [3]   **Beyza Ermis** [3]
Michael Brueckner [3]

Abstract

Stochastic linear bandits are a natural and well-studied model for structured exploration/exploitation problems and are widely used in applications such as online marketing and recommendation. One of the main challenges faced by practitioners hoping to apply existing algorithms is that usually the feedback is randomly delayed and delays are only partially observable. For example, while a purchase is usually observable some time after the display, the decision of not buying is never explicitly sent to the system. In other words, the learner only observes delayed positive events. We formalize this problem as a novel stochastic delayed linear bandit and propose `OTFLinUCB` and `OTFLinTS`, two computationally efficient algorithms able to integrate new information as it becomes available and to deal with the permanently censored feedback. We prove optimal $\tilde{O}(d\sqrt{T})$ bounds on the regret of the first algorithm and study the dependency on delay-dependent parameters. Our model, assumptions and results are validated by experiments on simulated and real data.

1. Introduction

Content optimization for websites and online advertising are among the main industrial applications of bandit algorithms (Chapelle & Li, 2011; Chapelle, 2014). The dedicated services sequentially choose an option among several possibilities and display it on a web page to a particular customer. In most real world architectures, for each recommendation request, the features of the products are joined and hashed with those of the current user and provide a (finite) action set included in $\mathbb{R}^d$. For that purpose, linear bandits (Chu et al., 2011; Abbasi-Yadkori et al., 2011) are among the

most adopted as they allow to take into account the structure of the space where the action vectors lie.

A key aspect of these interactions through displays on webpages is the time needed by a customer to make a decision and provide feedback to the learning algorithm, also known as the *conversion indicator* (Chapelle, 2014; Diemert et al., 2017). For example, a mid-size e-commerce website can serve hundreds of recommendations per second, but customers need minutes, or even hours, to make a purchase. In (Chapelle, 2014), the authors ran multiple tests on proprietary industrial datasets, providing a good example of how delays affect the performance of click-through rate estimation. They extract 30 days of display advertising data and find that delays are on average of the order of several hours and up to several days.

On the other extreme of the time scale, some companies optimize long-term metrics for customer engagement (e.g., accounting for returned products in the sales results) which by definition can be computed only several weeks after the bandit has played the action. Moreover, after a piece of content is displayed on a page, the user may or may not decide to react (e.g., click, buy a product). In the negative case, no signal is sent to the system and the learner cannot distinguish between actions for which the user did not click and those where they did, but the conversion is delayed. Note that in the use cases we consider, when a feedback is received, the learner is able to attribute it to the past action that triggered it.

Two major requirements for bandit algorithms in order to be ready to run in a real online service are the ability to leverage contextual information and handle delayed feedback. Many approaches are available to deal with contextual information (Abbasi-Yadkori et al., 2011; Agarwal et al., 2014; Auer et al., 2002; Neu, 2015; Beygelzimer et al., 2011; Chu et al., 2011; Zhou, 2015). Delays have been identified as a major problem in online applications (Chapelle, 2014). We give an overview of the existing literature in Section 2. However, to the best of our knowledge, no algorithm was able to address this problem given the requirements defined above.

Contributions   Our main contribution is a novel bandit algorithm called `On-The-Fly-LinUCB` (`OTFLinUCB`).

[1]DeepMind, London, UK [2]Otto-Von-Guericke Universität, Magdeburg, Germany [3]Amazon, Berlin, Germany. Correspondence to: Claire Vernade <vernade@google.com>.

The algorithm is based on `LinUCB` (Abbasi-Yadkori et al., 2011), but with confidence intervals and least-squares estimators that are adapted to account for the delayed and censored rewards (Section 4). The algorithm is complemented by a regret analysis, including lower bounds (Section 6). We then provide a variant inspired by Thompson sampling and Follow the Perturbed Leader (Section 7) and evaluate the empirical performance of all algorithms in Section 8.

2. Related Work

Delays in the environment response is a frequent phenomenon that may take many different forms and should be properly modelled to design appropriate decision strategies. For instance, in an early work on applications of bandit algorithms to clinical trials, (Eick, 1988) uses 'delays' to model the survival time of the patients, in which case delays are the reward rather than external noise, which is a radically different problem to ours. In the same vein, in the sequential stochastic shortest path problem (Talebi et al., 2017), the learner aims at minimising the routing time in a network.

Another example is parallel experimentation, where delays force the learner to make decisions under temporary uneven information. For instance, (Desautels et al., 2014; Grover et al., 2018) consider the problem of running parallel experiments that do not all end simultaneously. They propose a Bayesian way of handling uncertain outcomes to make decisions: they sample *hallucinated* results according to the current posterior. The related problem of gradient-based optimization with delayed stochastic gradient information is studied by (Agarwal & Duchi, 2011).

In online advertising, delays are due to the natural latency in users' responses. However, in many works on bandit algorithm, delays are ignored as a first approximation. In the famous empirical study of Thompson sampling (Chapelle & Li, 2011), a section is dedicated to analyzing the impact of delays on either Thompson sampling or `LinUCB`. While this is an early interest for this problem, they only consider fixed, non-random, delays of 10, 30 or 60 minutes. Similarly, in (Mandel et al., 2015), the authors conclude that randomized policies are more robust to this type of latencies. The general problem of online learning under *known* delayed feedback is addressed in (Joulani et al., 2013), including full information settings and partial monitoring, and we refer the interested reader to their references on those topics. The most recent and closest work to ours is (Zhou et al., 2019). The main difference with our approach is that they make strong assumptions on the distribution of the delays, while not having any censoring of the feedback. In that sense their problem is easier than ours because delays are fully observed. Nonetheless, the key idea of their algorithm is reminiscent to ours: they inscrease the exploration bonus by a quantity that corresponds to the amount of missing data

at each round, which is observable in their case, not in ours. Recent work (Li et al., 2019) address the case of unknown delays in the adversarial setting. The idea of ambiguous feedback, where delays are partially unknown is introduced in (Vernade et al., 2017).

Many models of delays for online advertising have been proposed to estimate conversion rates in an offline fashion: e.g. (Yoshikawa & Imai, 2018) (non-parametric) or (Chapelle, 2014; Diemert et al., 2017) (generalized linear parametric model).

An alternative, harder model relies only on anonymous feedback (Cesa-Bianchi et al., 2018; Pike-Burke et al., 2017; Arya & Yang, 2019): the rewards, when observed, cannot be directly linked to the action that triggered them in the past, and the learner has to deal with mixing processes. Finally, a recent full-information setting (Mann et al., 2018) suggests to incorporate intermediate feedback correlated with the rewards.

3. Learning Setting under Delayed Feedback

Notation All vectors are in $\mathbb{R}^d$ where $2 \leq d < \infty$ is fixed. For any symmetric positive definite matrix M and vector $x \in \mathbb{R}^d$, $\|x\|_M = \sqrt{x^T M x}$ and $\|x\|_2 = \|x\|_I$ is the usual L_2-norm of x, where I denotes the identity matrix in dimension d.

Learning setting Our setup involves a learner interacting with an environment over T rounds. The environment depends on an unknown parameter $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 \leq 1$ and a delay distribution $\mathcal{D}$ supported on the natural numbers. Note, in contrast to Vernade et al. (2017), we do not assume the learner knows $\mathcal{D}$. Then, in each round t,

1. The learner receives from the environment a finite set of actions $\mathcal{A}_t \subset \mathbb{R}^d$ with $|\mathcal{A}_t| = K_t < \infty$ and $a^\top \theta \in [0, 1]$ and $\|a\|_2 \leq 1$ for all $a \in \mathcal{A}_t$.

2. The learner selects an action A_t from $\mathcal{A}_t$ based on information observed so far.

3. The environment samples a reward $X_t \in \{0, 1\}$ and a delay $D_t \in \mathbb{N}$ that are *partially* revealed to the learner and where:

   3.a. $X_t \sim \mathcal{B}(A_t^\top \theta)$.

   3.b. D_t is sampled independently of A_t from $\mathcal{D}$.

4. Certain rewards resulting from previous actions are revealed to the learner. For $s \leq t$, let $C_{s,t} = \mathbb{1}\{D_s \leq t - s\}$, which is called the censoring variable and indicates whether or not the reward resulting from the decision in round s is revealed by round t. Then let $Y_{s,t} = C_{s,t} X_s$. The learner observes the collection

$\{Y_{s,t} : s \le t\}$ at the end of round t. If $X_t = 1$, we say that the action A_t converts.

The delays in combination with the Bernoulli noise model and censored observations introduces an interesting structure. When $X_s = 0$, then $Y_{s,t} = 0$ for all $t \ge s$, but $Y_{s,t} = 0$ is also possible when $X_s = 1$, but the reward from round s has been delayed sufficiently. On the other hand, if $Y_{s,t} = 1$, the learner can immediately deduce that $X_s = 1$.

The goal of the learner is to sequentially minimize the cumulative regret after T rounds, which is

$$R(T, \theta) = \sum_{t=1}^{T} \langle \theta, A_t^* \rangle - \langle \theta, A_t \rangle, \quad (1)$$

where $A_t^* = \arg\max_{a \in \mathcal{A}_t} \langle \theta, a \rangle$ is the action that maximises the expected reward in round t.

Remark 1. *The assumption that $a^\top \theta \in [0, 1]$ for all $a \in \mathcal{A}_t$ ensures the reward is well defined. A natural alternative is to replace the linear model with a generalized linear model. Our algorithms and analysis generalize to this setting in the natural way using the techniques of (Filippi et al., 2010; Jun et al., 2017). For simplicity, however, we restrict our attention to the linear model. The assumption that $\|\theta\|_2 \le 1$ and $\|a\|_2 \le 1$ for all actions are quite standard and the dependence of our results on alternative bounds is relatively mild.*

4. Concentration for Least Squares Estimators with Delays

This section is dedicated to disentangling the delays and the reward estimation. The combination of an unknown delay distribution and censored binary rewards makes it hopeless to store all past actions and wait for every conversion. For this reason our algorithm depends on a parameter m. If a reward has not converted within m rounds, the algorithm assumes it will never convert and ignores any subsequent signals related to this decision. There is also a practical advantage, which is that the learner does not need to store individual actions that occurred more than m rounds in the past. Define

$$\tilde{Y}_{s,t} = Y_{s,t}\mathbb{1}\{D_s \le m\} = X_s\mathbb{1}\{D_s \le \min(m, t - s)\},$$

which is the same as $Y_{s,t}$ except rewards that convert after more than m rounds are ignored. The learner then uses $\tilde{Y}_{s,t}$ to estimate a parameter that is proportional to θ using L_2-regularized least squares. Let $\lambda > 0$ be a regularization parameter and define

$$\hat{\theta}_t^{\text{w}} := \left(\sum_{s=1}^{t-1} A_s A_s^\top + \lambda I \right)^{-1} \left(\sum_{s=1}^{t-1} \tilde{Y}_{s,t} A_s \right) := V_t(\lambda)^{-1} B_t. \quad (2)$$

We now state our main deviation result.

Theorem 1. *Let $\tau_m = \mathbb{P}(D_1 \le m)$ and $\delta \in (0, 1)$. Then the following holds for all $t \le T$ with probability at least $1 - 2\delta$.*

$$\|\hat{\theta}_t^{\text{w}} - \tau_m\theta\|_{V_t(\lambda)} \le 2f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t^{-1}(\lambda)}, \quad (3)$$

where

$$f_{t,\delta} = \sqrt{\lambda} + \sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(\frac{d\lambda + t}{d\lambda}\right)}. \quad (4)$$

Proof. Let $\eta_s = X_s - A_s^\top\theta$ and $\epsilon_s = \mathbb{1}\{D_s \le m\} - \tau_m$ be the two types of noise affecting our observations, both being centered Bernoulli and independent of the past conditionally on A_s. By (Lattimore & Szepesvári, 2019, Theorem 20.4) it holds with probability at least $1 - 2\delta$ that for all $t \le T$,

$$\left\| \sum_{s=1}^{t} A_s X_s \epsilon_s \right\|_{V_t^{-1}}^2 \le f_{t,\delta}^2, \text{ and, } \left\| \sum_{s=1}^{t} A_s \eta_s \right\|_{V_t^{-1}}^2 \le f_{t,\delta}^2 \cdot, \quad (5)$$

where we used that $X_s \in [0, 1]$ for the first inequality. We comment on that step further below.

The next step is to decompose B_t with respect to the value of the censoring variables of the learner. We first rewrite B_t by expliciting the value of the indicator function in each term of its sum:

$$B_t = \sum_{s=1}^{t-m-1} A_s X_s \mathbb{1}\{D_s \le m\}$$
$$+ \sum_{s=t-m}^{t-1} A_s X_s \mathbb{1}\{D_s \le t - s\}$$
$$= \sum_{s=1}^{t-1} A_s X_s \mathbb{1}\{D_s \le m\}$$
$$+ \sum_{s=t-m}^{t-1} A_s X_s (\mathbb{1}\{D_s \le t - s\} - \mathbb{1}\{D_s \le m\}),$$

where we added m terms in the first sum and removed them in the second one. The second sum now contains the terms that will eventually convert but have not been received yet.

Now assume both events in Equation (5) hold. Using the decomposition above, we have

$$\|\hat{\theta}_t^{\text{w}} - \tau_m\theta\|_{V_t(\lambda)}$$
$$\le \left\| V_t(\lambda)^{-1} \sum_{s=1}^{t-1} A_s X_s (\tau_m + \epsilon_s) - \tau_m\theta \right\|_{V_t(\lambda)}$$
$$+ \left\| \sum_{s=t-m}^{t-1} A_s X_s \right\|_{V_t(\lambda)^{-1}}$$

The last term can be naively bounded and gives the second term of Equation (3). We can bound the first term using Equation (5):

$$\left\| V_t(\lambda)^{-1} \sum_{s=1}^{t-1} A_s X_s(\tau_m + \epsilon_s) - \tau_m \theta \right\|_{V_t(\lambda)}$$
$$\leq \left\| \sum_{s=1}^{t-1} A_s X_s \epsilon_s \right\|_{V_t(\lambda)^{-1}} + \left\| \sum_{s=1}^{t-1} A_s \eta_s \right\|_{V_t(\lambda)^{-1}}$$
$$\leq 2 f_{t,\delta} \, ,$$

where both the first and second inequalities follow from the triangle inequality applied to $\|\cdot\|_{V_t^{-1}}$ and the last from the assumption that the events in Equation (5) hold. $\qquad \square$

Remark 2. *The initial step of the proof in Eq (5) might seem loose but we explain why this term cannot easily be bounded more tightly. Note that the variance of ϵ_s is $\tau_m(1-\tau_m)$ and thus, when applying (Lattimore & Szepesvári, 2019, Theorem 20.4), we could obtain a tighter bound by taking it into acount, which would lead to $\tau_m^2 f_{t\delta}^2$. But had we included it there, it would have appeared in the expression of the upper bound, i.e. in the algorithm, and the learner would have needed its knowledge to compute the upper bound. This was the choice made by (Vernade et al., 2017). By removing it, we pay the price of slightly larger confidence intervals (more exploration) for not having to give prior information to the learner. We discuss other possible approaches in conclusion.*

Practical considerations As we mentioned already, a practical advantage of the windowing idea is that the learner need not store actions for which the feedback has not been received indefinitely. The cut-off time is often rather long, even as much as 30 days (Chapelle & Li, 2011).

Choosing the window The windowing parameter is often a constraint of the system and the lerner cannot choose it. Our results show the price of this external censoring on the regret. If the learner is able to choose m the choice is somewhat delicate. The learner effectively discards $1 - \tau_m$ proportion of the data, so ideally τ_m should be large, which corresponds to large m. But there is a price for this. The learner must store m actions and the confidence interval also depends (somewhat mildly) on m. When the mean $\mu = \mathbb{E}_{D \sim \mathcal{D}}[D]$ of the delay distribution $\mathcal{D}$ is finite and known, then a somewhat natural choice of the windowing parameter is $m = 2\mu$. By Markov's inequality this ensures that $\tau_m \geq 1/2$. The result continues to hold if the learner only knows an upper bound on μ.

Precisely how m should be chosen depends on the underlying problem. We discuss this issue in more detail in Section 6 where the regret analysis is provided.

5. Algorithm

We are now equipped to present OTFLinUCB, an optimistic linear bandit algorithm that uses concentration analysis from the previous section. The pseudocode of OTFLinUCB is given in Algorithm 1. It accepts as input a confidence level $\delta > 0$, a window parameter $m > 0$ and a regularization parameter $\lambda > 0$. In each round the algorithm computes the estimator $\hat{\theta}_t^{\mathrm{w}}$ using Equation (3) and for each arm $a \in \mathcal{A}_t$ computes an upper confidence bound on the expected reward defined by

$$U_t(a) = \langle a, \hat{\theta}_t^{\mathrm{w}} \rangle + \left(2 f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}} \right) \|a\|_{V_t(\lambda)^{-1}} \, . \tag{6}$$

Then action A_t is chosen to maximize the upper confidence bound:

$$A_t = \arg\max_a U_t(a) \, ,$$

where ties are broken arbitrarily.

Implementation details The algorithm needs to keep track of $V_t(\lambda)$ and B_t as defined in Equation (2). These can be updated incrementally as actions are taken and information is received. The algorithm also uses $V_t(\lambda)^{-1}$, which can be updated incrementally using the Sherman-–Morrison formula. In order to recompute $\alpha_{t,\delta}$ the algorithm needs to store the m most recent actions, which are also used to update B_t.

Computation complexity The computation complexity is dominated by three operations: (1) Updating $V_t(\lambda)$ and computing its inverse, which takes $O(d^2)$ computation steps using a rank-one update, and (2) computing the radius of the confidence ellipsoid, which requires $O(md^2)$ computations, one for each of the last m actions. Finally, (3) iterating over the actions and computing the upper confidence bounds, which requires $O(K_t d^2)$ computations. Hence the total computation per round is $O((K_t + m)d^2)$.

Space complexity The space complexity is dominated by: (1) Storing the matrix $V_t(\lambda)$ and (2) storing the m most recent actions, which are needed to compute the least squares estimator and the upper confidence bound. Hence, the space complexity is $O(md + d^2)$.

Improved computation complexity Because $V_t(\lambda)$ changes in every round, the radius of the confidence set needs to be recomputed in each round, which requires $O(md^2)$ computations per round. A minor modification reduces the computation complexity to $O(d^2)$. The idea is to notice that for any $a \in \mathbb{R}^d$ and $s \leq t$,

$$\|a\|_{V_s(\lambda)^{-1}} \geq \|a\|_{V_t(\lambda)^{-1}} \, .$$

Hence, one can store a buffer of scalars $\{\|A_s\|_{V_s(\lambda)^{-1}}, t - m \le s \le t - 1\}$ at the memory cost of $O(m)$. This slightly increases the upper confidence bounds, but not so much that the analysis is affected as we discuss in the next section.

Algorithm 1 OTFLinUCB

Input: Window parameter $m > 0$, confidence level $\delta > 0$ and $\lambda > 0$.

for $t = 2, \ldots, T$ **do**

    Receive action set $\mathcal{A}_t$

    Compute width of confidence interval:

$$\alpha_{t,\delta} = 2f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}$$

    Compute the least squares estimate $\hat{\theta}_t^{\mathrm{w}}$ using (2)

    Compute the optimistic action:

$$A_t = \arg\max_{a \in \mathcal{A}_t} \langle a, \hat{\theta}_t^{\mathrm{w}} \rangle + \alpha_{t,\delta} \|a\|_{V_t(\lambda)^{-1}}$$

    Play A_t and receive observations

end for

---

6. Regret Analysis

Our main theorem is the following high probability upper bound on the regret of OTFLinUCB. The proof combines the ideas from (Abbasi-Yadkori et al., 2011) with a novel argument to handle the confidence bound in Theorem 1, which has a more complicated form than what usually appears in the analysis of stochastic linear bandits.

Theorem 2. *With probability at least $1 - 2\delta$ the regret of* OTFLinUCB *satisfies*

$$R(T, \theta) \le \frac{4f_{n,\delta}}{\tau_m} \sqrt{2dT \log\left(\frac{d\lambda + T}{d\lambda}\right)}$$
$$+ \frac{4md}{\tau_m} \log\left(\frac{d\lambda + T}{d\lambda}\right).$$

Proof. Let $\alpha_{t,\delta} = 2f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}$, which is chosen so that the upper confidence bound for action a in round t is

$$U_t(a) = \langle a, \hat{\theta}_t^{\mathrm{w}} \rangle + \alpha_{t,\delta} \|a\|_{V_t(\lambda)^{-1}}.$$

and $A_t = \arg\max_{a \in \mathcal{A}_t} U_t(a)$. By Theorem 1, with probability at least $1 - 2\delta$ it holds that $\|\hat{\theta}_t^{\mathrm{w}} - \tau_m\theta\|_{V_t(\lambda)} \le \alpha_{t,\delta}$ for all t. Assume for the remainder that the above event

holds. Then

$$\langle A_t^*, \theta \rangle = \frac{1}{\tau_m} \langle \tau_m\theta, A_t^* \rangle$$
$$\le \frac{1}{\tau_m} \left(\langle \hat{\theta}_t^{\mathrm{w}}, A_t^* \rangle + \alpha_{t,\delta} \|A_t^*\|_{V_t(\lambda)^{-1}} \right) = \frac{U_t(A_t^*)}{\tau_m}$$
$$\le \frac{U_t(A_t)}{\tau_m} = \frac{1}{\tau_m} \left(\langle \hat{\theta}_t^{\mathrm{w}}, A_t \rangle + \alpha_{t,\delta} \|A_t\|_{V_t(\lambda)^{-1}} \right).$$

Therefore the regret in round t is bounded by

$$\langle A_t^* - A_t, \theta \rangle \le \frac{1}{\tau_m} \langle A_t, \hat{\theta}_t^{\mathrm{w}} - \tau_m\theta \rangle + \frac{\alpha_{t,\delta}}{\tau_m} \|A_t\|_{V_t(\lambda)^{-1}}$$
$$\le \frac{2\alpha_{t,\delta}}{\tau_m} \|A_t\|_{V_t(\lambda)^{-1}},$$

where the second inequality follows from Cauchy-Schwarz. We now substitute the value of $\alpha_{t,\delta}$ and bound the overall regret by

$$R(T, \theta) \le \frac{4}{\tau_m} \sum_{t=1}^{T} f_{t,\delta} \|A_t\|_{V_t^{-1}}$$
$$+ \frac{2}{\tau_m} \sum_{t=1}^{T} \|A_t\|_{V_t(\lambda)^{-1}} \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}. \quad (7)$$

The first sum is bounded in the same way as the standard setting (Abbasi-Yadkori et al., 2011):

$$\frac{4}{\tau_m} \sum_{t=1}^{T} f_{t,\delta} \|A_t\|_{V_t(\lambda)^{-1}} \le \frac{4f_{n,\delta}}{\tau_m} \sqrt{T \sum_{t=1}^{T} \|A_t\|_{V_t(\lambda)^{-1}}^2}$$
$$\le \frac{4f_{n,\delta}}{\tau_m} \sqrt{2dT \log\left(\frac{d\lambda + T}{d\lambda}\right)},$$

where the first inequality follows from Cauchy-Schwarz and the second from the elliptical potential lemma (Lattimore & Szepesvári, 2019, Lemma 19.4). For the second sum in Equation (7) we introduce a new trick. Using the fact that $ab \le (a^2 + b^2)/2$,

$$\frac{2}{\tau_m} \sum_{t=1}^{T} \|A_t\|_{V_t(\lambda)^{-1}} \sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}}$$
$$\le \frac{1}{\tau_m} \sum_{t=1}^{T} \sum_{s=t-m}^{t-1} \left(\|A_t\|_{V_t(\lambda)^{-1}}^2 + \|A_s\|_{V_t(\lambda)^{-1}}^2 \right)$$
$$\le \frac{1}{\tau_m} \sum_{t=1}^{T} \sum_{s=t-m}^{t-1} \left(\|A_t\|_{V_t(\lambda)^{-1}}^2 + \|A_s\|_{V_s(\lambda)^{-1}}^2 \right)$$
$$\le \frac{2m}{\tau_m} \sum_{t=1}^{T} \|A_t\|_{V_t(\lambda)^{-1}}^2 \le \frac{4md}{\tau_m} \log\left(\frac{d\lambda + T}{d\lambda}\right),$$

where in the second inequality we used the fact that for $s \le t$, $V_s(\lambda) \le V_t(\lambda)$[1], so that $V_s(\lambda)^{-1} \ge V_t(\lambda)^{-1}$. Sub-

---

[1] here we denote $A \le B$ if for any $x \in \mathbb{R}_+^d$, $\|x\|_A \le \|x\|_B$

stituting the previous two displays into Equation (7) completes the proof. □

Remark 3. *The choice of m is left to the learner. It influences the bound in two ways: (1) The lower-order term is linear in m, which prevents the user from choosing m very large. On the other hand, τ_m is increasing in m, which pushes the user in the opposite direction. Designing an adaptive algorithm that optimizes the choice of m online remains a challenge for the future.*

Lower bound We now provide a non-asymptotic minimax lower bound for K-armed stochastic Bernoulli bandits showing that in the windowed setting there is an unavoidable dependence on τ_m. Note, an asymptotic problem-dependent bound for this setting was already known (Vernade et al., 2017). Although our results are specialized to the finite-armed bandit model, we expect that standard analysis for other action sets should follow along the same lines as (Lattimore & Szepesvári, 2019) (§24).

Theorem 3. *For any policy π and $K > 1$ and $T \geq 1$ and $\tau_m \in (0, 1)$ there exists a K-armed Bernoulli bandit such that $R(T, \theta) \geq c \min\{T, \sqrt{TK/\tau_m}\}$, where $c > 0$ is a universal constant.*

Interestingly, the dependence on τ_m appears in the square root, while in our upper bounds it is not. We speculate that the upper bound is loose. In fact, were τ_m known it would be possible to improve our upper bounds by using confidence intervals based on Bernstein's inequality that exploit the reduced variance that is a consequence of τ_m being small. When τ_m is unknown you might imagine estimating the variance. We anticipate this should be possible, but the complexity of the algorithm and analysis would greatly increase.

7. Thompson sampling

The standard implementation of Thompson sampling for linear bandits without delays and Gaussian noise is to sample $\tilde{\theta}_t \sim \mathcal{N}(\hat{\theta}_t, V_t(\lambda)^{-1})$ where $\hat{\theta}_t$ is the usual regularized least squares estimator

$$\hat{\theta}_t = V_t(\lambda)^{-1} \sum_{s=1}^{t-1} A_s X_s .$$

The algorithm then chooses

$$A_t = \arg\max_{a \in \mathcal{A}_t} \langle a, \tilde{\theta}_t \rangle .$$

This algorithm corresponds to Thompson sampling (or posterior sampling) when the prior is Gaussian with zero mean and λI covariance. No frequentist analysis exists for this algorithm, but empirically it performs very well. In the delayed setting the $(X_s)_{s=1}^{t-1}$ are not available to the learner at

time t. Nevertheless, it is possible to propose a randomized algorithm in the spirit of Thompson sampling. To motivate our choices, recall that the standard concentration analysis for least squares regression by (Abbasi-Yadkori et al., 2011) shows that with high probability

$$\|\hat{\theta}_t - \theta\|_{V_t(\lambda)} \leq f_{t,\delta} .$$

In the delayed setting, on the other hand, Theorem 1 shows that

$$\|\hat{\theta}_t^{\mathrm{w}} - \theta\|_{V_t(\lambda)} \leq f_{t,\delta} + \sum_{s=t-m}^{t-1} \|A_s\|_{V_s(\lambda)^{-1}} \quad (8)$$

$$= f_{t,\delta} \left(1 + \frac{\sum_{s=t-m}^{t-1} \|A_s\|_{V_s(\lambda)^{-1}}}{f_{t,\delta}} \right) . \quad (9)$$

An ansatz guess for a sampling algorithm that uses the delayed least squares estimator is to compute $\hat{\theta}_t^{\mathrm{w}}$ and then sample

$$\tilde{\theta}_t \sim \mathcal{N}\left(\hat{\theta}_t^{\mathrm{w}}, \beta_{t,\delta} V_t(\lambda)^{-1} \right) ,$$

where $\beta_{t,\delta} = 1 + (\sum_{s=t-m}^{t-1} \|A_s\|_{V_t(\lambda)^{-1}})/f_{t,\delta}$. The choice of $\beta_{t,\delta}$ is rather heuristic. A more conservative choice would be the right-hand side of Equation (9). The resulting algorithm roughly corresponds to sampling from the confidence set used by our optimistic algorithm. Although this sacrifices certain empirical advantages, we expect the analysis techniques by (Agrawal & Goyal, 2013; Abeille & Lazaric, 2017) could be applied to prove a frequentist regret bound for this algorithm.

Remark 4. *Algorithms based on adding noise to an empirical estimate are often referred to as 'follow the perturbed leader', which has been effectively applied in a variety of settings (Abeille & Lazaric, 2017; Kveton et al., 2018). An advantage of sampling approaches is that the optimization problem to find A_t is a linear program, which for large structured action sets may be more efficient than finding the arm maximizing an upper confidence bound.*

Remark 5. *A genuine implementation of Thompson sampling would require a prior on the space of delay distributions as well as the unknown parameter. We are not hopeful about the existence of a reasonable prior for which computing or sampling from the posterior is efficient.*

8. Experiments

In this section we illustrate the two realistic settings handled by this work. The first case below, with geometrically distributed delays, corresponds to the empirical study done by (Chapelle, 2014), already reproduced in simulations by (Vernade et al., 2017). The second case, with arbitrary
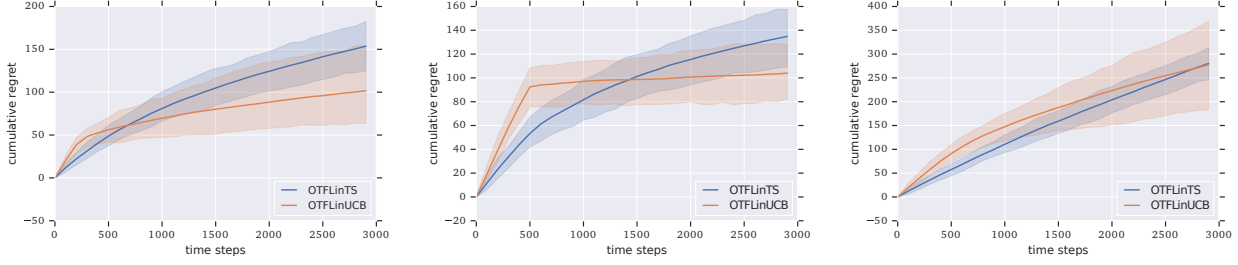
Figure 1. Results of our simulations. From left to right, the plots report the performance of the two algorithms with $(m, \mu) = \{(100, 100), (500, 100), (100, 500)\}$. Results are averaged over 100 independent runs.

Algorithm 2 `OTFLinTS`:

1: **Input:** Window parameter $m > 0$, confidence level $\delta > 0$, $\lambda > 0$.
2: **for** $t = 2, \ldots, T$ **do**
3:     Receive action set $\mathcal{A}_t$
4:     Compute width of confidence interval:

$$\beta_{t,\delta} = 1 + \frac{\sum_{s=t-m}^{t-1} \|A_s\|_{V_s(\lambda)^{-1}}}{f_{t,\delta}}$$

5:     Compute the least squares estimate $\hat{\theta}_t^{\mathrm{w}}$ using (2)

6:     Sample $\tilde{\theta}_t \sim \mathcal{N}(\hat{\theta}_t^{\mathrm{w}}, \beta_{t,\delta} V_t(\lambda)^{-1})$

7:     Compute action $A_t = \arg\max_{a \in \mathcal{A}_t} \langle a, \tilde{\theta}_t \rangle$

8:     Play A_t and receive observations
9: **end for**

---

heavy-tailed delays, corresponds to another use-case we extracted from the data released by (Diemert et al., 2017).[2]

Remark 6. *To the best of our knowledge, there is no competitor for this problem. In particular, despite the similarity of the algorithm DUCB of (Zhou et al., 2019) with ours, it cannot be implemented when delays are not observed. Specifically, DUCB, maintains a quantity G_t which is equal to the exact amount of missing data (delayed feedback not converted yet). In our case, this quantity is not observable. The same comment applies for the QPM-D algorithm of (Joulani et al., 2013) and similar queue-based approaches. On the other end, existing algorithms for unknown delays are not derived for linear bandits with arbitrary action sets.*

Well-behaved delays In the datasets analyzed[3] by (Chapelle, 2014), delays are empirically shown to have an exponential decay. As in (Vernade et al., 2017), we run real-
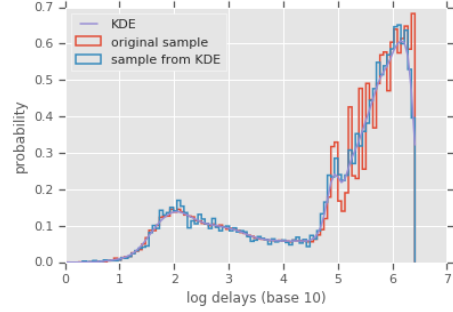

Figure 2. Empirical distribution of the $\log_{10}$ delays.

istic simulations based on the orders of magnitude provided in their study. We arbitrarily choose $d = 5$, $K = 10$. We fix the horizon to $T = 3000$, and we choose a geometric delay distribution with mean $\mu = \mathbb{E}[D_t] \in \{100, 500\}$. In a real setting, this would correspond to an experiment that lasts 3h, with average delays of 6 and 30 minutes[4] respectively. The online interaction with the environment is simulated: we fix $\theta = \{1/\sqrt{d}, \ldots, 1/\sqrt{d}\}$ and at each round we sample and normalize K actions from $\{0, 1\}^d$. All result are averaged over 50 independent runs. We show three cases on Figure 1: $(m, \mu) = (100, 100)$ ($\tau_m = 0.63$) when delays are short and even though the timeout is set a bit too low, the algorithm manages to learn within a reasonable time range; $(m, \mu) = (500, 100)$ ($\tau_m = 0.993$) when the window parameter is large enough so almost all feedback is received; and $(m, \mu) = (100, 500)$ when the window is set too low and the regret suffers from a high $1/\tau_m = 5.5$. Note that the notion of 'well-tuned' window is dependent on the memory constraints one has. In general, the window is a system parameter that the learner cannot easily set to their convenience. In all three cases, `OTFLinUCB` performs better than `OTFLinTS`, which we believe is due to the rough posterior approximation. Comparing the two leftmost figures, we see the impact of increasing the window size: the log regime starts earlier but the asymptotic regret is similar – roughly

---

[2]The code for all data analysis and simulations is available at https://sites.google.com/view/bandits-delayed-feedback
[3]The datasets were not released.

[4]Note that in (Chapelle, 2014), time is rather measured in hours and days.

$R(T) = 100$ in both cases. The rightmost figure, compared with the leftmost one, shows the impact of a high level of censoring. Indeed, with $1/\tau_m = 5.5$, only few feedback are received within the window and the algorithm will need much longer to learn. This causes the regret increase, and also explains why at $T = 3000$, both algorithms are still in the linear regime.
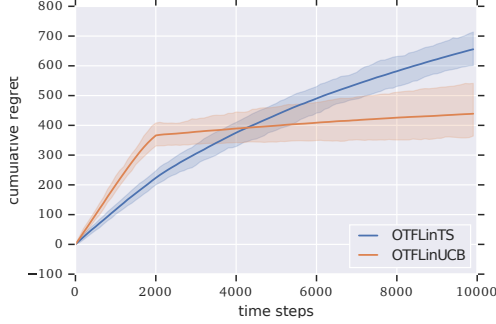

Figure 3. Simulation with realistic delays and $m = 2000$, sampled according to the distribution of Figure 2.

Heavy-tailed delays. Interestingly, the more recent dataset released by (Diemert et al., 2017)[5] features heavy-tailed delays, despite being sourced from a similar online marketing problem in the same company. We run simulations in the same setting as before, but this time we generate the delays according to the empirical delay distribution extracted from the dataset. On Figure 2, we show the distribution of the log-delays fitted with a gaussian kernel using the Scipy library. A high proportion of those delays are between 10^5 and 10^6 so we rescaled them by a factor 0.01 to maintain the experiment horizon to the reasonable value of $T = 10^4$. Nevertheless, many rewards will not be observed within the horizon. We compare `OTFLinUCB` and `OTFLinTS` with window parameter $m = 2000$. The results are shown on Figure 3 and show that `OTFLinUCB` is able to cope with heavy-tailed delays reasonably well. It is interesting to observe that the behavior of both algorithms is similar to the central plot of Figure 1, when the window is larger than the expectation of the delays.

9. Discussion

We introduced the delayed stochastic linear bandit setting and proposed two algorithms. The first uses the optimism principle in combination with ellipsoidal confidence intervals while the second is inspired by Thompson sampling and follow the perturbed leader.

There are a number of directions for future research, some of which we now describe.

---

[5]https://ailab.criteo.com/criteo-attribution-modeling-bidding-dataset/

Improved algorithms Our lower bound suggests the dependence on τ_m in Theorem 2 can be improved. Were this value known we believe that using a concentration analysis that makes use of the variance should improved the dependence to match the lower bound. When τ_m is not known, however, the problem becomes more delicate. One can envisage various estimation schemes, but the resulting algorithm and analysis are likely to be rather complex.

Generalized linear models When the rewards are Bernoulli it is natural to replace the linear model with a generalized linear model. As we remarked already, this should be possible using the machinery of (Filippi et al., 2010).

Thompson sampling Another obvious question is whether our variant of Thompson sampling/follow the perturbed leader admits a regret analysis. In principle we expect the analysis by (Agrawal & Goyal, 2013) in combination with our new ideas in the proof of Theorem 2 can be combined to yield a guarantee, possibly with a different tuning of the confidence width $\beta_{t,\delta}$. Investigating a more pure Bayesian algorithm that updates beliefs about the delay distribution as well as unknown parameter is also a fascinating open question, though possibly rather challenging.

Refined lower bounds Our current lower bound is proven when $\mathcal{A}_t = \{e_1, \ldots, e_d\}$ is the standard basis vectors for all t. It would be valuable to reproduce the results where $\mathcal{A}_t$ is the unit sphere or hypercube, which should be a straightforward adaptation of the results in (Lattimore & Szepesvári, 2019, §24).

Acknowledgements

References

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

Abeille, M. and Lazaric, A. Linear thompson sampling revisited. In *AISTATS 2017-20th International Conference on Artificial Intelligence and Statistics*, 2017.

Agarwal, A. and Duchi, J. C. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pp. 873–881, 2011.

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646, 2014.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135, 2013.

Arya, S. and Yang, Y. Randomized allocation with nonparametric estimation for contextual multi-armed bandits with delayed rewards. *arXiv preprint arXiv:1902.00819*, 2019.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26, 2011.

Cesa-Bianchi, N., Gentile, C., and Mansour, Y. Nonstochastic bandits with composite anonymous feedback. In *Conference On Learning Theory*, pp. 750–773, 2018.

Chapelle, O. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1097–1105. ACM, 2014.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

Diemert, E., Meynet, J., Galland, P., and Lefortier, D. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, Halifax, NS, Canada, August, 14, 2017*, pp. To appear. ACM, 2017.

Eick, S. G. The two-armed bandit with delayed responses. *The Annals of Statistics*, pp. 254–264, 1988.

Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2010.

Grover, A., Markov, T., Attia, P., Jin, N., Perkins, N., Cheong, B., Chen, M., Yang, Z., Harris, S., Chueh, W., et al. Best arm identification in multi-armed bandits with delayed feedback. In *International Conference on Artificial Intelligence and Statistics*, pp. 833–842, 2018.

Joulani, P., Gyorgy, A., and Szepesvári, C. Online learning under delayed feedback. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1453–1461, 2013.

Jun, K.-S., Bhargava, A., Nowak, R., and Willett, R. Scalable generalized linear bandits: Online computation and hashing. In *Advances in Neural Information Processing Systems*, pp. 99–109, 2017.

Kveton, B., Szepesvári, C., Wen, Z., Ghavamzadeh, M., and Lattimore, T. Garbage in, reward out: Bootstrapping exploration in multi-armed bandits. 2018.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. Cambridge University Press (preprint), 2019.

Li, B., Chen, T., and Giannakis, G. B. Bandit online learning with unknown delays. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 993–1002, 2019.

Mandel, T., Brunskill, E., and Popović, Z. Towards more practical reinforcement learning. In *24th International Joint Conference on Artificial Intelligence, IJCAI 2015*. International Joint Conferences on Artificial Intelligence, 2015.

Mann, T. A., Gowal, S., Jiang, R., Hu, H., Lakshminarayanan, B., and Gyorgy, A. Learning from delayed outcomes with intermediate observations. *arXiv preprint arXiv:1807.09387*, 2018.

Neu, G. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 3168–3176, 2015.

Pike-Burke, C., Agrawal, S., Szepesvari, C., and Grunewalder, S. Bandits with delayed anonymous feedback. *arXiv preprint arXiv:1709.06853*, 2017.

Talebi, M. S., Zou, Z., Combes, R., Proutiere, A., and Johansson, M. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

Vernade, C., Cappé, O., and Perchet, V. Stochastic bandit models for delayed conversions. In *Conference on Uncertainty in Artificial Intelligence*, 2017.

Yoshikawa, Y. and Imai, Y. A nonparametric delayed feedback model for conversion rate prediction. *arXiv preprint arXiv:1802.00255*, 2018.

Zhou, L. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*, 2015.

Zhou, Z., Xu, R., and Blanchet, J. Learning in generalized linear contextual bandits with stochastic delays. In *Advances in Neural Information Processing Systems 32*, pp. 5198–5209, 2019.