

Impatient Bandits: Optimizing Recommendations for the Long-Term Without Delay

Thomas M. McDonald*
tommcDonald955@gmail.com
University of Manchester
United Kingdom

Lucas Maystre
lucasm@spotify.com
Spotify
United Kingdom

Mounia Lalmas
mounia@acm.org
Spotify
United Kingdom

Daniel Russo
djr2174@gsb.columbia.edu
Columbia University & Spotify
United States

Kamil Ciosek
kamilc@spotify.com
Spotify
United Kingdom

ABSTRACT

Recommender systems are a ubiquitous feature of online platforms. Increasingly, they are explicitly tasked with increasing users' long-term satisfaction. In this context, we study a content exploration task, which we formalize as a multi-armed bandit problem with delayed rewards. We observe that there is an apparent trade-off in choosing the learning signal: Waiting for the full reward to become available might take several weeks, hurting the rate at which learning happens, whereas measuring short-term proxy rewards reflects the actual long-term goal only imperfectly. We address this challenge in two steps. First, we develop a predictive model of delayed rewards that incorporates all information obtained to date. Full observations as well as partial (short or medium-term) outcomes are combined through a Bayesian filter to obtain a probabilistic belief. Second, we devise a bandit algorithm that takes advantage of this new predictive model. The algorithm quickly learns to identify content aligned with long-term success by carefully balancing exploration and exploitation. We apply our approach to a podcast recommendation problem, where we seek to identify shows that users engage with repeatedly over two months. We empirically validate that our approach results in substantially better performance compared to approaches that either optimize for short-term proxies, or wait for the long-term outcome to be fully realized.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Sequential decision making*; • **Mathematics of computing** → *Probabilistic inference problems*.

KEYWORDS

multi-armed bandits; recommender systems; Bayesian modeling

*This work was completed as part of an internship at Spotify.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08...\$15.00
<https://doi.org/10.1145/3580305.3599386>

ACM Reference Format:

Thomas M. McDonald, Lucas Maystre, Mounia Lalmas, Daniel Russo, and Kamil Ciosek. 2023. Impatient Bandits: Optimizing Recommendations for the Long-Term Without Delay. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, August 6–10, 2023, Long Beach, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3580305.3599386>

1 INTRODUCTION

Many online platforms rely on recommender systems to assist users in finding relevant items among vast collections of content [29]. Applications are wide-ranging: recommender systems help individuals find books, movies or audio content [6, 25]; they help doctors find medical treatments for their patients [37], and students find learning resources [38], among many others. A key question underpins the design of any recommender system: What is a successful recommendation? Across many applications, there is an ongoing shift towards defining success at longer time-horizons [44, 45], as long-term metrics are often better suited to capture users' satisfaction and platforms' goals [15]. For example, e-commerce platforms may want to maximize long-term revenue, subscription-based services may want to increase retention, and social platforms may want to encourage habitual engagement measured over several weeks or months. In the context of podcast recommendations on an online audio streaming platform, recent work has shown that explicitly optimizing for long-term engagement (measured over a 60-day window post-recommendation) can significantly improve the user experience [24]. Most of the literature, however, implicitly assumes that there is sufficient data to estimate the long-term impact of recommendations.

1.1 Content Exploration Problem

In this paper, we focus on a specific aspect of recommender systems and seek to address a content exploration problem. On most online platforms, new content is released regularly. In order to learn about that content's appeal, we must first recommend it to users. This is known as the *cold-start problem*. After ensuring an adequate amount of information has been gathered, an effective system should rapidly shift recommendations away from poor content.

We formalize this task as a *multi-armed bandit* problem, where we seek to identify promising content through successive interactions with users [21]. Optimizing for long-term definitions of

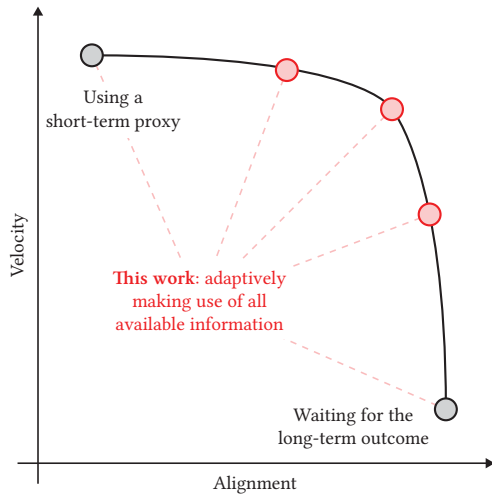


Figure 1: Short-term proxies enable a rapid feedback loop, but might be poorly aligned with long-term success metrics, which take longer to realize. Our method finds the optimal tradeoff by adaptively making use of all available information at a given time.

success in the bandit setting is challenging, as long-term metrics are—by construction—delayed [16]. This gives rise to an apparent tradeoff, illustrated in Figure 1, between using short-term proxies that are observable quickly (top-left) and ensuring that actions selected are aligned with long-term success (bottom-right). We propose a means of circumventing this tradeoff by exploiting the insight that *most long-term outcomes become increasingly predictable over time*.

Driven by practical applications, we assume that intermediate outcomes are progressively revealed over time, from the moment the action is selected up to the moment the full reward is observed. We call this the *progressive feedback* setting. We develop a probabilistic model that forms beliefs about the delayed rewards an arm generates on the basis of outcomes observed so far. As time passes, uncertainty diminishes and the model is able to make increasingly precise predictions. To facilitate this, we contend that historical data from distinct but similar applications (e.g., previous content releases) can be used to learn the association between intermediate and long-term outcomes. In effect, we propose a meta-learning approach that learns to infer long-term outcomes of interest from intermediate observations, revealed progressively over time. We then take advantage of this reward model to address our sequential decision-making problem, by combining the predictive model with a bandit algorithm. The bandit uses probabilistic predictions from the model to efficiently balance exploration and exploitation. Even if the first few intermediate outcomes are insufficient to perfectly infer the average delayed reward an arm generates, they might be sufficient to reveal that this arm is outperformed by others. In such cases, our bandit algorithm will shift effort away from the arm. Note that, in contrast to well-studied bandit settings where feedback is observed at once, either immediately or after a given

delay, the progressive feedback setting presents distinctive challenges: Information can be obtained actively, by selecting an action, or passively by letting time unfold and incrementally receiving new data about the outcomes of actions taken in the past.

Our methodology is very general, and can be applied to a wide range of problems. In this work, we consider a recently-studied podcast recommendation application [24]. In this application, actions correspond to podcast shows, and the reward is defined as the number of days a user engages with a show in the 59 days that follow a successful recommendation. Intermediate outcomes consist of binary activity indicators for each of the 59 days, observed with the corresponding delay. We evaluate our approach using data from the Spotify audio streaming platform, and show that *a)* the full reward can be accurately predicted after only a few days of observation, and that *b)* the content-exploration problem can be solved much quicker than approaches that rely on short-term proxies or wait for the full reward to become available.

Summary of Contributions. In this work, we make the following contributions.

- A Bayesian filtering approach to reward estimation, which enables us to incorporate all available information in order to predict delayed outcomes and quantify uncertainty (Section 3.1).
- A meta-learning approach, where the prior and noise covariance structures that power Bayesian filtering are themselves learned from data. The method learns across items how to make rapid inferences about a new item (Section 3.2).
- The *impatient bandit algorithm*, a novel algorithm for the progressive feedback setting, which uses intermediate information received at each round to iteratively update the Bayesian filter, and enables us to efficiently balance exploration and exploitation whilst providing recommendations that optimize for long-term engagement (Section 3.3).
- An application of our impatient bandit algorithm to a real-world podcast recommendation problem, presented alongside empirical results, that show that our proposed method considerably outperforms approaches based on fully-delayed feedback or short-term proxy metrics (Section 4).

2 RELATED WORK

We start by briefly discussing relevant related work on multi-armed bandits and applications to recommender systems.

Multi-Armed Bandits. Often used to model online platforms [23], multi-armed bandits (MAB) formalize a simple sequential decision-making problem, where at each round t an agent selects one of several possible actions and receives a corresponding reward r_t . The goal is usually to maximize the sum of rewards received over a given time horizon. The simplest and most widely studied bandit setting is the *strictly sequential feedback* scenario, where r_t is immediately observed [1, 35]. However many extensions have been proposed.

One such extension is the case of *parallelized actions*. Rather than simply selecting a single action at each round and receiving a single corresponding reward, we can also consider a scenario in which multiple actions can be taken at each round in parallel [11, 18]. This extension is also referred to as the *batched* bandit setting. The

challenge arises from having to concurrently select several actions without knowing the reward associated with the other actions in the batch.

An additional variant of the MAB relevant to our work is the *delayed feedback* setting, which can be viewed as a generalization of the batch feedback setting [18]. In this setting, the reward r_t is only revealed after a delay of Δ rounds, i.e., at round $t + \Delta$. In this case, the agent is forced to make a series of decisions without knowledge of the results of all previous actions taken. Prior work in this area has utilized Thompson sampling [9, 18] and upper-confidence bound (UCB) [11, 17] algorithms to address this setting.

Thompson Sampling. Thompson sampling is a class of algorithms used for sequential decision-making in bandit settings, that efficiently balances *exploration* of the action space with *exploitation* of actions that are believed to be associated with comparatively large rewards [31]. Whilst the technique was introduced almost a century ago [36], it has become increasingly popular over the course of the last decade due to its strong empirical performance when applied to modern, large-scale online learning problems [12, 32].

In the face of uncertainty, Thompson sampling randomizes among all actions that are plausibly optimal. This allows for greater robustness to delayed feedback compared to algorithms based on upper-confidence bounds, which are deterministic and rely on rapidly changing beliefs to adjust which arm is sampled. There is both theoretical and empirical evidence that, due to its randomized nature, Thompson sampling is resilient to delayed feedback [9, 18, 27, 40]. This characteristic of the algorithm is crucial in our problem.

In our work, we use Thompson sampling in combination with a belief model that predicts a long-term metric using progressively revealed intermediate observations. We meta-learn this model on historical data. This aspect of our work connects to a series of recent papers that develop provable bounds on the loss in performance due to fitting the prior used in Bayesian bandit algorithms from past data [4, 5, 33].

Intermediate Feedback. Several recent papers have employed Thompson sampling in settings with delayed outcomes, but useful intermediate observations [8, 39, 42]. UCB algorithms have also been applied to this setting by Grover et al. [13], who consider a scenario where noisy observations of the true feedback are received at intermediate rounds between t and $t + \Delta$. Key differences between their work and ours include the fact that Grover et al. consider the problem of top- k best-arm identification with a stochastic delay, in contrast to our objective of cumulative regret minimization with a fixed delay. Another differentiating factor of greater consequence is the fact that they assume that the intermediate feedback consists of independent random variables, whereas our progressive feedback is crucially not i.i.d., which is how our model is able to effectively generalize. Additionally, our work employs Bayesian filtering to seamlessly perform inference, an approach explicitly motivated by a real use-case, where historical data allows us to fit an informed prior.

Outside of the literature on MABs, Prentice [26] and Athey et al. [2] formalize conditions under which intermediate feedback can be used to estimate long-term outcomes. They also find empirically that intermediate feedback can lead to both increased accuracy and precision in estimates of long-term outcomes.

Recommender Systems & Long-Term Goals. Bandits are a popular approach for addressing many types of recommendation problem. In a seminal paper, Li et al. [21] use a contextual variant of the bandit problem to personalize recommendations on a news platform, and more recently, Aziz et al. [3] use MABs to recommend podcasts by maximizing the impression-to-stream rate.

Optimizing recommendations for long-term user engagement is a problem that is of great practical interest in industry, and bandits have also been used previously to address this specific scenario. Wu et al. [41] address this problem using a UCB algorithm that models the temporal return behaviour of users to maximize the cumulative number of clicks from a group of users over a period of time.

Beyond bandits, more general reinforcement learning (RL) approaches have also been applied to the problem of maximizing long-term user engagement [44, 45]. “Full” RL enables principled reasoning about inter-temporal tradeoffs and delayed rewards, at the expense of increased complexity. Implementing effective RL algorithms that address realistic recommender system problems is non-trivial due to the challenging nature of off-policy learning and evaluation [43].

3 METHODOLOGY

We present our approach to solving the content exploration problem outlined in the introduction. We adopt the terminology of multi-armed bandits. We consider a set of N actions, $\mathcal{A} = \{a_1, \dots, a_N\}$, corresponding, e.g., to different recommendation candidates. At each round $t = 1, 2, \dots$, we select one or more actions. For every action we select, we observe a reward r_a after a delay of Δ rounds, i.e., at round $t + \Delta$. Informally, we seek to develop a methodology that helps us quickly identify and exploit actions with high mean reward $\bar{r}_a = \mathbb{E}[r_a]$. We assume that the reward r_a is a function of intermediate observations $z_{a,1}, \dots, z_{a,K}$, that become available progressively during the interval $[t, t + \Delta]$ after selecting the action. We call this the *progressive feedback* setting.

In Section 3.1, we consider a fixed action a and develop a Bayesian reward model that takes advantage of intermediate observations to estimate the mean reward \bar{r}_a . In Section 3.2, we take advantage of historical data to estimate the parameters of the reward model, effectively instantiating a *meta-learning* approach. Building on this model, in Section 3.3, we develop a bandit algorithm that efficiently balances exploration and exploitation in the progressive feedback setting.

Concrete Example. While this section introduces the methodology in a generic way, it is helpful to keep a concrete application in mind. In Section 4 we consider a podcast recommendation problem, where the actions \mathcal{A} correspond to podcast shows. The reward r is the cumulative engagement with a podcast show over a period of Δ days: $r_a = \sum_{i=i}^{\Delta} z_{a,i}$.

3.1 Bayesian Reward Model

We consider a fixed action a and, for conciseness, we omit a from all subscripts. Let r be the sample reward and $\bar{r} = \mathbb{E}[r]$ be the mean reward associated to selecting the action. Define the sample trace, $z = (z_1, \dots, z_K) \in \mathbf{R}^K$, as a vector containing intermediate outcomes. We assume that z_k is observed after $\Delta_k \leq \Delta$ rounds, and, without loss of generality, that $\Delta_1 \leq \dots \leq \Delta_K$. Correspondingly,

we define the average trace as $\bar{z} = \mathbb{E}[z]$. We postulate the following generative model of sample traces $\{z_m\}$:

$$\bar{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), \quad z_m = \bar{z} + \boldsymbol{\varepsilon}_m, \quad \boldsymbol{\varepsilon}_m \sim \mathcal{N}(\mathbf{0}, V) \text{ i.i.d.} \quad (1)$$

That is, we assume a priori that the average trace \bar{z} corresponding to the action is sampled from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ , and that a sample trace z_m is a noisy copy of \bar{z} , corrupted by additive zero-mean Gaussian noise with covariance matrix V , independently for each m . Furthermore, we assume that we can reconstruct the reward from all intermediate observations as

$$r = \mathbf{w}^\top z,$$

where $\mathbf{w} \in \mathbb{R}^K$ is a vector of weights. By the linearity of expectation, it follows that $\bar{r} = \mathbf{w}^\top \bar{z}$. We treat \mathbf{w} as given, and $\{\boldsymbol{\mu}, \Sigma, V\}$ as model parameters. We discuss how to learn them from data in Section 3.2.

Assume that we are at round t and that we have selected the action M times so far, at rounds $t_1 \leq \dots \leq t_m \leq t$. We represent the observations collected at round t as a dataset of M independent traces, $\mathcal{D} = \{(z_m, \ell_m) : m = 1, \dots, M\}$. Some traces might only be partially observed, and we use $\ell_m \doteq \max\{k : \Delta_k \leq t - t_m\}$ to index the last element of z_m that is observed at round t .

3.1.1 Iterative Belief Updates. We consider the problem of estimating \bar{r} given \mathcal{D} . Instead of reasoning about \bar{r} directly, we begin by addressing the problem of estimating \bar{z} . We take a Bayesian approach and seek to compute the posterior distribution

$$p(\bar{z} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \bar{z}) \mathcal{N}(\bar{z} \mid \boldsymbol{\mu}, \Sigma).$$

Given our generative model (1), we will show that the posterior remains Gaussian, even in the presence of partially observed traces. Finally, writing $p(\bar{z} \mid \mathcal{D}) \doteq \mathcal{N}(\bar{z} \mid \boldsymbol{\mu}', \Sigma')$ and given that \bar{r} is a linear function of the mean trace \bar{z} , we have that

$$\bar{r} \mid \mathcal{D} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2),$$

where $\boldsymbol{\mu} = \mathbf{w}^\top \boldsymbol{\mu}'$ and $\sigma^2 = \mathbf{w}^\top \Sigma' \mathbf{w}$.

We describe the process by which we fold in a single trace into the belief. The full posterior can be obtained by repeating this procedure iteratively, M times. For conciseness, we drop the subscript m and denote the trace and cutoff index as (z, ℓ) , respectively. We denote by $\mathbf{A}_{i,:j}$ the submatrix obtained by taking the i first rows and the j first columns of a matrix \mathbf{A} . Similarly, we denote by $\mathbf{a}_{:i}$ the first i elements of the vector \mathbf{a} . Thanks to the self-conjugacy property of the Gaussian distribution, we can write the posterior distribution of \bar{z} after observing the ℓ first elements of the trace z as a multivariate Gaussian with mean vector and covariance matrix

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \Sigma_{:,K;\ell} (\Sigma_{:, \ell; \ell} + V_{\ell; \ell})^{-1} (z_{:, \ell} - \boldsymbol{\mu}'_{:, \ell}),$$

$$\Sigma' = \Sigma + \Sigma'_{:,K;\ell} (\Sigma_{:, \ell; \ell} + V_{\ell; \ell})^{-1} \Sigma_{:, \ell; K},$$

respectively. We refer the reader to Rasmussen et al. [28, Section A.2] for more details on these update equations. The complete iterative procedure is provided in Algorithm 1.

A Note on Gaussian Noise. The assumption in (1) that each trace z is Gaussian with mean \bar{z} might seem restrictive at first sight. For example, in Section 4, we consider binary observation vectors $z \in \{0, 1\}^A$, for which a Gaussian is arguably a poor model. In fact, given that our ultimate goal is to infer \bar{z} from several traces, the

Algorithm 1 Computing the posterior of \bar{z} .

Input: Parameters $\boldsymbol{\mu}, \Sigma, V$, dataset \mathcal{D}

```

1:  $\boldsymbol{\mu}' \leftarrow \boldsymbol{\mu}$ 
2:  $\Sigma' \leftarrow \Sigma$ 
3: for  $(z, \ell) \in \mathcal{D}$  do
4:    $\mathbf{A} \leftarrow \Sigma'_{:,K;\ell} (\Sigma'_{:, \ell; \ell} + V_{\ell; \ell})^{-1}$ 
5:    $\boldsymbol{\mu}' \leftarrow \boldsymbol{\mu} + \mathbf{A} (z_{:, \ell} - \boldsymbol{\mu}'_{:, \ell})$ 
6:    $\Sigma' \leftarrow \Sigma' + \mathbf{A} \Sigma'_{:, \ell; K}$ 
7: end for

```

impact of this assumption is relatively benign. To see this, assume that we are given M full traces z_1, \dots, z_M such that $z_m = \bar{z} + \boldsymbol{\varepsilon}_m$, where $\{\boldsymbol{\varepsilon}_m\}$ are independently and identically distributed but not necessarily Gaussian. It can be shown that the empirical average $\hat{z} = M^{-1} \sum_m z_m$ is a sufficient statistic for \bar{z} given $\{z_m\}$. For M large, we can invoke the central limit theorem to argue that a Gaussian approximation for \hat{z} (and, correspondingly, a Gaussian approximation for the individual traces z_1, \dots, z_M) is accurate for the purpose of estimating \bar{z} .

Optimizing the Implementation. For simplicity, we have described Bayesian inference in our model as a sequential procedure. In practice, there are several ways in which Algorithm 1 can be made more computationally efficient. These include *a*) updating the posterior using multiple traces in a single batch, instead of processing each trace independently; *b*) performing incremental updates by reusing beliefs from previous rounds; and *c*) only updating beliefs for actions that have received new observations.

3.2 Training the Reward Model

A crucial aspect of our method is the ability to take advantage of past data to learn the model parameters $\{\boldsymbol{\mu}, \Sigma, V\}$. Specifically, we assume access to historical data about a different set of actions \mathcal{A}' . In the context of a recommender system, for example, this could be existing content for which we already have a sufficient amount of interaction data. For each $a \in \mathcal{A}'$, denote by $\mathcal{H}_a = \{(z_{am}, r_{am}) : m = 1, \dots, M_a\}$ the data corresponding to action a .

For each action $a \in \mathcal{A}'$, we begin by computing the empirical mean trace vector and noise covariance matrix

$$\hat{z}_a = M_a^{-1} \sum_{z \in \mathcal{H}_a} z, \quad \hat{V}_a = M_a^{-1} \sum_{z \in \mathcal{H}_a} (z - \hat{z}_a)(z - \hat{z}_a)^\top,$$

respectively. We then estimate the model parameters $\boldsymbol{\mu}, \Sigma, V$ by using empirical averages, as

$$\boldsymbol{\mu} = |\mathcal{A}'|^{-1} \sum_{a \in \mathcal{A}'} \hat{z}_a,$$

$$\Sigma = |\mathcal{A}'|^{-1} \sum_{a \in \mathcal{A}'} (\boldsymbol{\mu} - \hat{z}_a)(\boldsymbol{\mu} - \hat{z}_a)^\top,$$

$$V = |\mathcal{A}'|^{-1} \sum_{a \in \mathcal{A}'} \hat{V}_a.$$

In principle, more advanced estimation methods might be used, such as type-II maximum likelihood, also known as *empirical Bayes* [28]. In practice, however, we have found that the simple empirical averages described above are very effective.

Intuitively, the covariance matrices Σ and V play a critical role in our approach. They encode the correlations between outcomes observed at different points in time. If intermediate outcomes observed early on are highly predictive of later outcomes, we expect that we can accurately estimate \bar{z} (and thus \bar{r}) without waiting for the full Δ rounds required to observe r . We will revisit this from an empirical perspective in Section 4.2.

A Note on the Weights. Our approach assumes that the reward is a given linear function of the trace. For example, in Section 4, we consider a problem where the reward is defined as $r = \sum_k z_k$, corresponding to $\mathbf{w} \doteq \mathbf{1}$. In practice, one might try to fit long-term objectives to a linear model, by solving a regression problem

$$\arg \min_{\mathbf{w}} \sum_{a \in \mathcal{A}} \sum_{(z, y) \in \mathcal{H}'_a} (y - \mathbf{w}^\top \mathbf{z})^2,$$

where y is a target that is not exactly a linear function of \mathbf{z} . In this case, it is important to note that the reward $r = \mathbf{w}^\top \mathbf{z}$ is an approximation of the true objective y . We briefly elaborate on this in Appendix A.

3.3 Bandit Algorithm

Equipped with a model capable of making inferences about the arms' mean rewards given intermediate observations, we can now develop a bandit algorithm that works effectively in the progressive-feedback setting, where information about the reward is revealed progressively over multiple rounds.

Although several different objectives for the bandit problem exist in the literature, in this work we focus on the goal of minimizing the *cumulative expected regret*. In the case of a single action being selected at each round, we define the cumulative expected regret at round T as

$$\mathbb{E} [R_T] = \mathbb{E} \left[\sum_{t=1}^T (\bar{r}^* - \bar{r}_t) \right],$$

where \bar{r}^* is the mean reward obtained by selecting the best action, r_t is the mean reward corresponding to the action selected at round t , and the expectation is taken over the algorithm's internal randomization over actions [34]. We extend this definition to the case where we select multiple actions in parallel at each round, as

$$\mathbb{E} [R_T] = \mathbb{E} \left[\sum_{t=1}^T \left(\bar{r}^* - B^{-1} \sum_{i=1}^B \bar{r}_{t,i} \right) \right],$$

where B is the number of actions per round, and $\bar{r}_{t,i}$ is the mean reward associated to the i th action performed at round t .

Before describing our algorithm, we first present a brief overview of Thompson sampling [30, 31]. Slivkins et al. [34] give a generalized formulation of Thompson sampling for bandits with immediately observable rewards, which we simplify here for ease of exposition. In a strictly sequential multi-armed bandit, when an agent takes an action $a_t \in \mathcal{A}$, a corresponding reward $r_t \sim q_\theta(\cdot | a_t)$ is observed. We place a prior distribution p over the model parameters θ . The action to be taken at each round is chosen by computing $a_t \leftarrow \arg \max_{a \in \mathcal{A}} \mathbb{E}_{q_\theta} [r_t | a_t = a]$, yielding a realized observation, which we then condition on to update p . Rather than taking a *greedy* approach, whereby $\hat{\theta}$ is the expectation of θ with respect to p , Thompson sampling instead samples the parameters from p (i.e. $\hat{\theta} \sim p$). This is a subtle, but powerful difference, as it ensures

Algorithm 2 Impatient Bandit Algorithm

Input: Actions \mathcal{A} , number of actions per round B

- 1: **for** $t = 1, \dots, T$ **do**
- 2: **for** $a \in \mathcal{A}$ **do**
- 3: Update \mathcal{D}_a with new observations
- 4: $p(\bar{z}_a) \leftarrow \mathcal{N}(\bar{z}_a | \mu_a, \Sigma_a)$ via Algorithm 1 on \mathcal{D}_a
- 5: $p(\bar{r}_a) \leftarrow \mathcal{N}(\bar{r}_a | \mathbf{w}^\top \mu_a, \mathbf{w}^\top \Sigma_a \mathbf{w})$
- 6: **end for**
- 7: **for** $i = 1, \dots, B$ **do**
- 8: **for** $a \in \mathcal{A}$ **do**
- 9: Sample mean reward $\hat{r}_a \sim p(\bar{r}_a)$
- 10: **end for**
- 11: Take action $a_{t,i} \leftarrow \arg \max_{a \in \mathcal{A}} \{\hat{r}_a\}$
- 12: **end for**
- 13: **end for**

that the algorithm does not purely exploit actions that yield large rewards in the first few rounds of feedback, ignoring other, possibly better actions. Due to the non-zero variance of the belief on the mean reward associated with each action, Thompson sampling may select an action other than that which the greedy algorithm would deem optimal. This mechanism trades off exploration and exploitation effectively, and is known to achieve low cumulative regret [31].

3.3.1 Impatient Bandit Algorithm. Our approach builds on the Thompson sampling algorithm, applying it to the progressive feedback setting. In our case, the parameters θ simply correspond to the average rewards $\{\bar{r}_a : a \in \mathcal{A}\}$. The key to our approach is to make use of the reward model developed in Section 3.1 to infer beliefs $p(\bar{r}_a)$. By updating beliefs based intermediate outcomes, we enable the sampling step in the Thompson sampling to take full advantage of *all* information collected up to round t , and not only of fully observed rewards. We call the resulting procedure the *impatient bandit* and describe it in Algorithm 2.

4 APPLICATION TO PODCASTS

We consider a concrete application of our content-exploration problem to podcast recommendations on Spotify, a leading online audio streaming platform.¹ In Section 4.1, we begin by describing how our generic methodology can be applied to optimizing long-term user engagement with podcasts, and present a real-world dataset of podcast consumption traces. In Section 4.2, we study the reward model in isolation, and evaluate its predictive accuracy. In Section 4.3, we consider a sequential decision-making task in the progressive-feedback setting, and compare the empirical performance of our impatient bandit against competing approaches.

To complement the experiments presented in this section, we provide a companion software package with a reference implementation of our algorithm.² While we are unable to publicly release the data due to confidentiality reasons, our package includes a synthetic dataset that leads to comparable findings.

¹See: <https://newsroom.spotify.com/company-info/>.

²See: <https://github.com/spotify-research/impatient-bandits>.

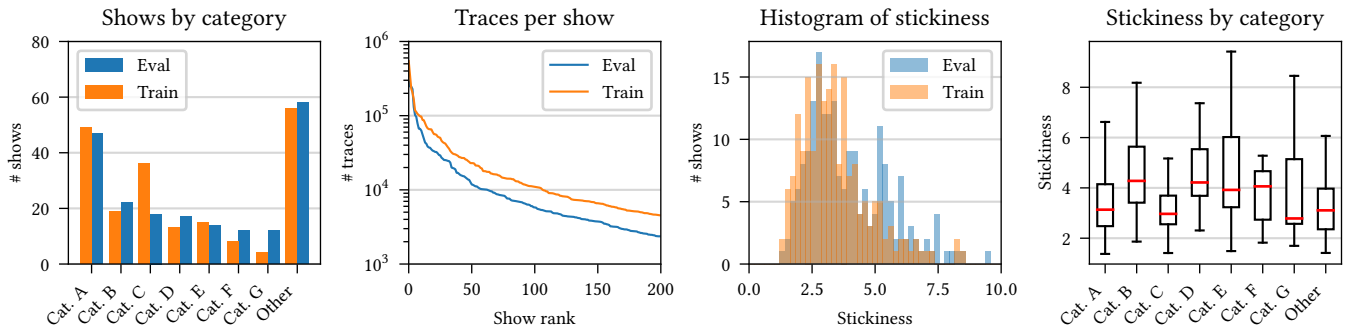


Figure 2: Summary statistics of a dataset of podcast shows and corresponding consumption traces. There is large heterogeneity in show-stickiness (center-right), even when controlling for category (boxplot, right).

4.1 Problem Formulation & Data

Traditionally, podcast recommender systems optimize for short-term rewards, such as the click-through-rate [3]. Recently, Maystre et al. [24] show that explicitly optimizing podcast recommendations for long-term outcomes can lead to substantial impact on a real-world, large-scale recommendation problem. They propose a system that reasons simultaneously about the *clickiness* (i.e., the click-through-rate) and the *stickiness* of a recommendation. Stickiness is defined in terms of the downstream consequences of a successful recommendation. In particular, the authors suggest counting the number of days users engage with a podcast show discovered through a recommendation over the 59 days that follows a first listen. In this work, we adopt their definitions and optimization metrics, but consider a specific subset of the overall recommendation problem. We focus on estimating stickiness (i.e., we do not model the click-through rate), and seek to quickly identify new podcast shows that have high average stickiness. This lets us investigate the challenging problem of estimating long-term rewards for new content *in isolation*, without being confounded by other aspects of the overall recommendation problem. Maystre et al. [24] discuss how to estimate the click-through rate, and how to personalize models to take into account users’ preferences, but they do not address the content exploration problem we study here.

Formally, we instantiate the methodology described in Section 3 as follows. The set of actions \mathcal{A} corresponds to N candidate podcast shows that are new and that we need to explore. We define the reward $r \in \{0, \dots, 59\}$ as the number of days a user engages with a show in the 59 days that follow a successful recommendation.³ This reward is observed with a delay of $\Delta = 60$ days. We refer to the mean reward \bar{r}_a corresponding to show a as the *stickiness* of the show. We collect intermediate outcomes $z_k = \mathbf{1}\{\text{the user engaged on day } k\}$ into an activity trace $\mathbf{z} \in \{0, 1\}^{59}$. Naturally, each activity indicator z_k is observed with delay $\Delta_k = k + 1$. From these definitions, it follows that $r = \sum_k z_k = \mathbf{w}^\top \mathbf{z}$, where $\mathbf{w} = \mathbf{1}$ is the all-ones vector. The distinct set \mathcal{A}' and historical data \mathcal{H}_a , $a \in \mathcal{A}'$ correspond to a set of established shows and the corresponding historical consumption traces, respectively. We seek to develop a bandit algorithm that learns to maximize the long-term engagement attributable to

each recommendation. This is a clear instance of the progressive feedback setting; Every day, actions must be taken with only partial knowledge about the outcome of decisions made in the previous 59 days.

4.1.1 Dataset. We consider a dataset of podcast consumption traces collected on the Spotify audio streaming platform between September 2021 and May 2022. The data is divided into a training set and an independent validation set. Each subset consists of a sample of 200 podcast shows first published on the platform during a given three-month period. For each of these shows, the data contains a representative sample of users that discover the show during the same three-month period. For each user, we obtain a longitudinal trace that captures their engagement with the show on each day starting from the day of discovery,⁴ in the form of a 59-dimensional binary vector. The training and validation sets cover podcast shows appearing during the periods September–December 2021 and January–March 2022, respectively. Each subset covers a distinct set of shows.

The podcast shows included in the dataset span a wide range of categories, from *Arts* to *True Crime*. Figure 2 (left) shows that the distribution of shows over categories is comparable across the two periods.⁵ In total, the dataset consists of 8.77M activity traces, corresponding to a total of 26M cumulative active-days. The number of traces per show ranges between 2.4K and 295K, with a median of 5.8K (Figure 2, center-left). For each show, we define the ground-truth stickiness by means of the empirical average (across users) of the cumulative active-days. Figure 2 (center-right) shows that there is substantial heterogeneity in stickiness across shows, with the lower quartile, median, and upper quartile at 2.6, 3.4, and 4.6 days, respectively. This suggests that the downstream impact of a discovery can be very different across shows. We note that the stickiness histogram is comparable across the two subsets. Finally, some categories appear to be somewhat stickier than others, but within-category variability is significantly larger than between-category variability (Figure 2, right).

³For the purposes of this paper, note that such a long horizon crystallizes the challenges of optimizing for the long-term, and forces us to develop methods that explicitly address these challenges.

⁴We define a discovery as the first stream that happens on the platform.

⁵For confidentiality reasons, we obfuscate the names of the categories.

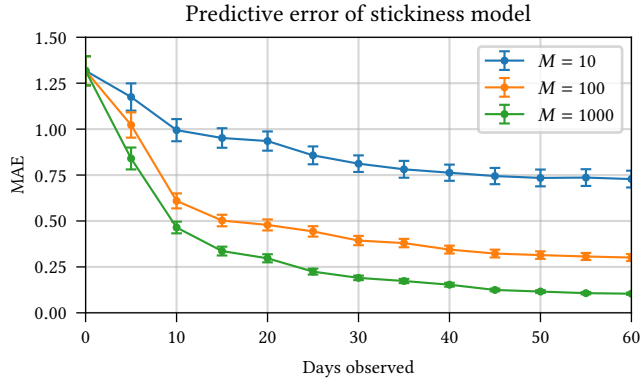


Figure 3: Mean absolute error (\pm standard error) of stickiness estimates as a function of days of data observed.

4.2 Evaluating the Reward Model

We focus first on evaluating our Bayesian reward model in isolation. We estimate μ , Σ and V by using the shows and consumption traces contained in the training dataset. For each show in the validation dataset, we randomly sample 2000 user traces. From this subset, we use M traces to infer the stickiness of each show (via Algorithm 1), and we use the remaining $(2000 - M)$ traces for computing the ground truth empirical stickiness. In Figure 3, we visualize how the predictive accuracy of our stickiness model varies as a function of both number of days observed, and number of user traces observed. We see that stickiness predictions can be relatively accurate after observing only 10 days of data. The predictions improve as time passes, and having access to more user traces further increases predictive accuracy.

We now study the noise and prior covariance matrices V and Σ , respectively. We investigate how the variance of the sample reward, $\mathbb{V}[r | z_t, \bar{z}]$, and the variance of the mean reward, $\mathbb{V}[\bar{r} | \bar{z}_t]$, are progressively explained away as t increases, i.e., as we condition on more and more days observed. Normalizing the t th conditional variance by the total (unconditional) variance, we obtain the fraction of total variance explained by the first t intermediate outcomes. Technical details are provided in Appendix B, alongside visualizations of the covariance matrices as heatmaps.

In Figure 4 (left), we look at the noise covariance V . The diagonal straight line represents a hypothetical scenario where daily activity indicators z are distributed independently and identically around \bar{z} , resulting in us gaining a constant amount of information about r for each additional day of observed data. The *empirical* line corresponds to the actual covariance matrix learned by our approach. We can see that around 10 days worth of data is sufficient to capture over 50% of the aleatoric uncertainty in the reward r . There are two factors that account for this. The first is that, as time progresses, user activity reduces, so the variance is larger early on in the 60-day window; This would be the case even if activity was entirely independent across days. The second and more interesting factor, is that activity is correlated across days, therefore knowledge of activity up to a given day allows us to predict future activity. The *uncorrelated* line corresponds to the hypothetical case where the diagonal of the covariance matrix matches that of V , but there is no

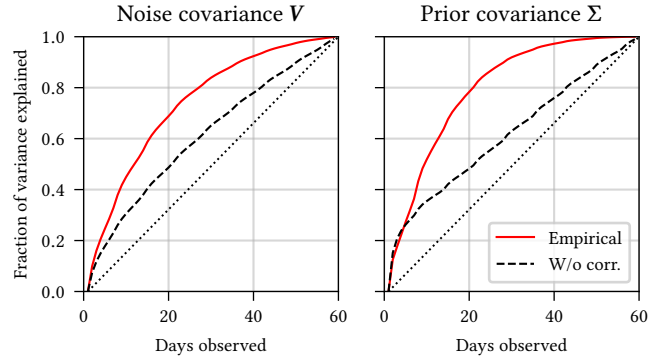


Figure 4: Explained variance as a function of days of activity data observed.

correlation (i.e., off-diagonal elements of the matrix are set to zero). The gap between the empirical and uncorrelated curves illustrates how much information we gain by exploiting the fact that past activity is predictive of future activity.

Similarly, in Figure 4 (right), we look at the prior covariance Σ . Intuitively, this lets us explore how much of the variance of \bar{r} would be explained if we were to observe the first t elements of each of a set of M independent sample traces, as $M \rightarrow \infty$. We can draw similar conclusions from this plot as to those mentioned in the context of V . However the trend is even more stark here, as 50% of variance is explained by just eight days of data, and 95% of the variance is explained within a month.

4.3 Sequential Decision-Making Task

We now turn our focus to the evaluation of the impatient bandit algorithm, and to the comparison of its empirical performance with competing approaches. The way observed feedback is used is one of the main points of differentiation between the approaches we consider. As such, we refer to our approach as *progressive*, since we makes use of all observations as they are revealed over time. We contrast the performance of our approach to three baselines.

Delayed. The case in which we solely receive full observations, Δ days after an action is taken is referred to as the *delayed* feedback baseline. This naive approach does not attempt to take advantage of intermediate outcomes.

Day-two proxy. We treat the second day of activity as a proxy for stickiness and discard all subsequent information. This baseline captures an intuitive outcome that is clearly related to the goal of maximizing habitual engagement: Does the user return to the show the day after discovering it? This baseline is representative of short-term proxies widely used in recommender systems, such as the click-through-rate, the dwell time, or the conversion rate [7, 10, 20].

Oracle. Finally, we include an *oracle* baseline, which assumes that the full 60-day activity trace is received immediately after an action is taken. This is clearly unrealistic, but it is useful to include as it provides an upper-bound on the performance of any model.

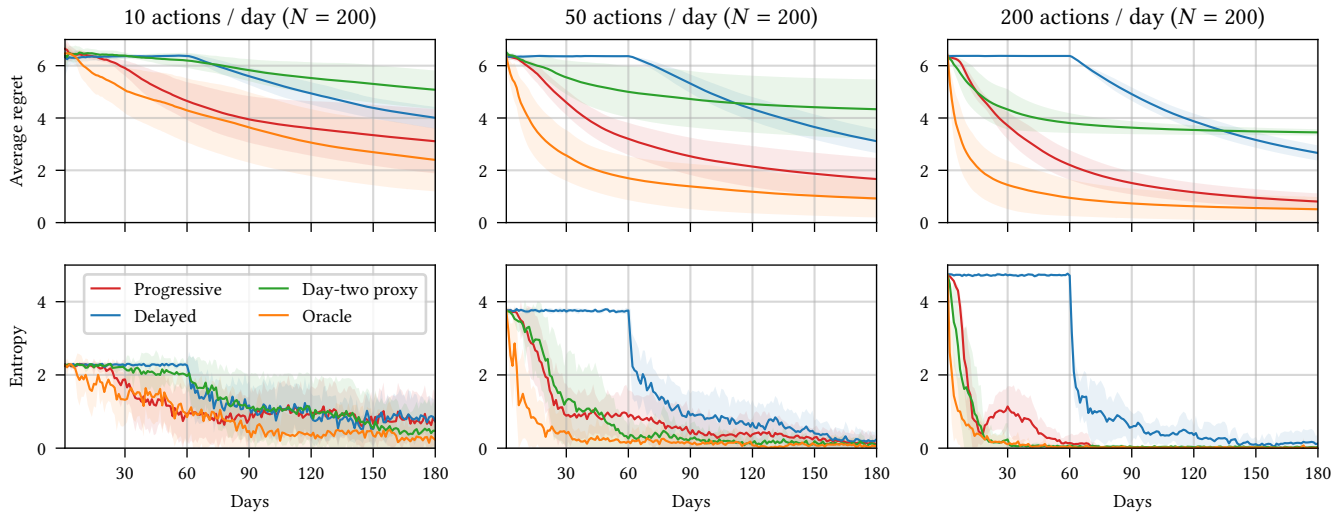


Figure 5: Average per-step regret and entropy of set of actions taken at each round, for $N = 200$ podcast shows.

These baselines have been chosen to illustrate the benefits of incorporating progressive feedback into a bandit algorithm, and the effectiveness of our approach in making use of this intermediate information. We use Thompson sampling for all of our baselines to ensure that any performance differences are due to the manner in which feedback is being considered, rather than to the relative strengths and weaknesses of different families of bandit algorithms. For similar reasons, we do not compare to any works that study other aspects of recommendation unrelated to this study, such as personalization.

To mimic a realistic deployment setting in which the prior would be computed using data from the past, we compute our prior using the training data, and then run our algorithm on the unseen evaluation dataset. A single prior is computed using all available traces from all 200 shows in our training set, this is then used for all of the experiments in this section. We run the bandit for 180 rounds (corresponding to approximately 6 months), repeating each experiment 10 times to generate confidence intervals for the average regret. Three different experimental setups are considered, with varying numbers of actions taken per day.

4.3.1 Results. Figure 5 (top row) visualizes the average per-step regret for each of these experimental settings, which ideally should tend to zero as $t \rightarrow \infty$. Across all of the experiments, the performance of the delayed approach is poor as it is forced to make uninformed decisions for the first Δ rounds of evaluation due to the inherent delay in feedback being received. Additionally, the oracle, as expected, outperforms the other approaches due to the unrealistic amount of information it has access to. The day-two proxy approach performs well at first, comparably to our approach across the initial month of evaluation, but past this stage the limitations of optimizing for this proxy become clear. The proxy is not well aligned, and the per-step regret rapidly plateaus.

Our progressive approach exhibits superior performance compared to the competing delayed and day-two proxy approaches; in fact, the performance of our approach is closer to that of the oracle.

As we increase the number of actions per round, we see a slight reduction in per-step regret across all approaches.

Figure 5 (bottom row) provides an alternative perspective on the outcome of these experiments, visualizing the entropy of the set of actions taken at each round. Should a bandit converge on recommending a single show repeatedly at each round, the entropy would tend to zero. The entropy plots show that, early on in the evaluation phase, our progressive algorithm tends to diversify across actions more than the oracle and day-two proxy. The interpretation of this is that our approach is performing a broader exploration of the action space, a characteristic that can be very useful in a realistic, deployment setting, which we discuss below. Not only does Figure 5 let us compare the empirical performance of all four approaches, it also enables us to differentiate the effects of observational noise from the effects of delayed feedback. For example, the large gap in per-step regret between the oracle and delayed approaches is entirely due to the delay in feedback, as both approaches receive full user traces of length Δ . On the other hand, the gap in per-step regret between the oracle and day-two proxy approaches is due to the fact that the second day of activity is a noisy proxy for the true stickiness, thus the day-two proxy approach tends to rapidly converge on a small subset of sub-optimal shows (this can be seen from its entropy, which quickly approaches zero).

In Figure 6, we present additional results for a scenario in which we have a smaller action space, consisting of a subset of 50 shows sampled from the original evaluation dataset discussed previously. This is clearly a simpler problem setting, as evidenced by the fact that all of the approaches tend more quickly to lower values of average regret in this case except for the day-two proxy feedback. Besides this observation, the results follow largely similar trends to those seen in Figure 5.

Changing Show Set. In addition to considering a static library of shows, we also briefly consider a setting where we have a library of shows that is constantly evolving over time. Specifically, at each round, one randomly selected show is removed from the library and

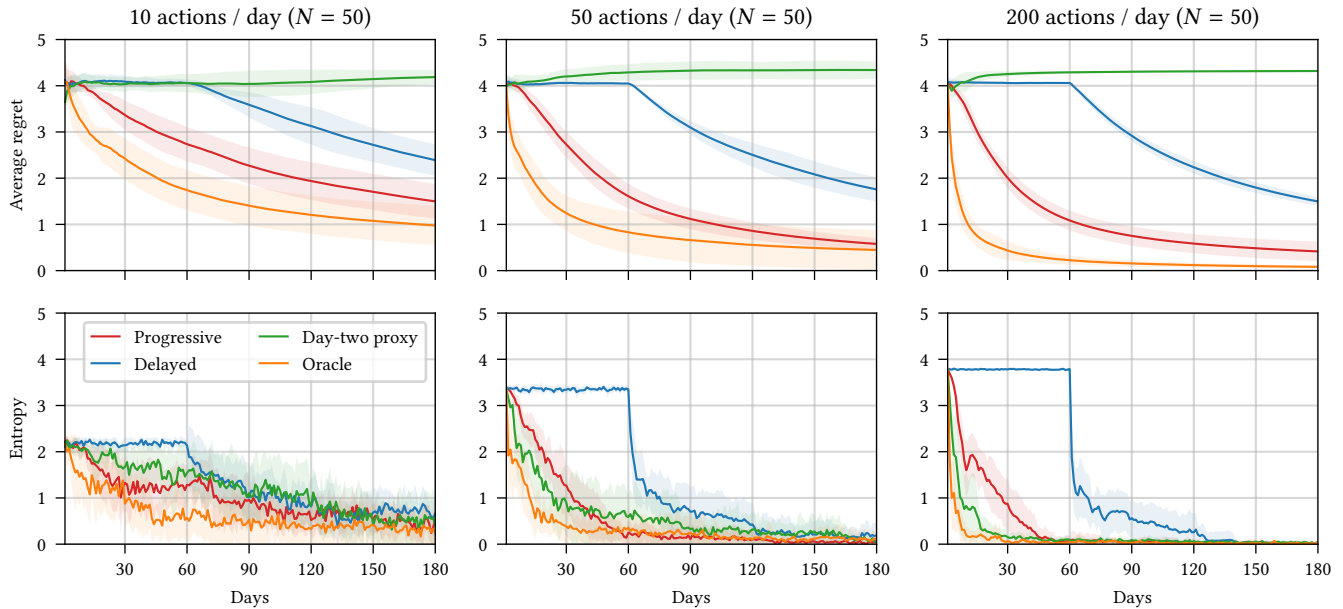


Figure 6: Average per-step regret and entropy of set of actions taken at each round, for $N = 50$ podcast shows.

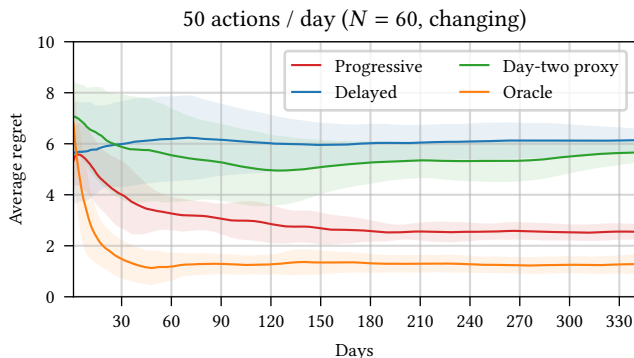


Figure 7: Average per-step regret in a scenario where we have a *changing* set of 60 podcast shows.

is replaced with a new show. From the results shown in Figure 7, we can see that our algorithm once again considerably outperforms the delayed and day-two proxy feedback schemes, even in this challenging setting where new content is constantly entering the system and exploration is always necessary.

5 CONCLUSION & FUTURE WORK

In this work we have introduced a new type of bandit algorithm that efficiently optimizes for delayed rewards, assuming that intermediate outcomes correlated with the final reward are revealed progressively over time. This is achieved by way of a meta-learning approach. We begin by learning the parameters of a Bayesian filter by using historical data from a related but distinct problem. Then, we combine this probabilistic reward model with Thompson sampling, effectively balancing exploration and exploitation. The key

to our success is that the Bayesian filter is able to make accurate inferences on delayed rewards using intermediate outcomes.

We have evaluated our framework empirically on a podcast content exploration problem. Using real-world platform data, experimental results show that our approach, which utilizes all available intermediate information to estimate a long-term reward, significantly outperforms approaches that only use short-term proxies or wait until the reward is available.

We have presented a non-personalized methodology for optimizing recommendations over an extended period of time. A natural avenue for future work is extending this to a personalized setting. Conceptually, we do not foresee any major difficulty. In Appendix C, we sketch an contextual extension of our Bayesian filter that conditions beliefs on user embeddings. Another avenue of research would be to build a theoretical understanding of the favorable empirical performance observed in practical applications. Can we formally characterize the benefits of progressive feedback over delayed rewards in terms of the average regret?

Finally, we would like to emphasize that the general framework we present can also benefit other application domains, beyond recommendations on online content platforms. For example, we believe our algorithm could be used to allocate resources in hyperparameter optimisation problems [22] by identifying more or less promising hyperparameter configurations in the early stages of training from an array of intermediate validation metrics. This could significantly reduce the computational cost of training large models and its environmental impact, which has become a major concern in the ML community in recent years [19].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback, which greatly contributed to improving this paper.

REFERENCES

- [1] Shipra Agrawal and Navin Goyal. 2012. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*. JMLR Workshop and Conference Proceedings, 39–1.
- [2] Susan Athey, Raj Chetty, Guido W Imbens, and Hyunseung Kang. 2019. *The surrogate index: Combining short-term proxies to estimate long-term treatment effects more rapidly and precisely*. Technical Report. National Bureau of Economic Research.
- [3] Maryam Aziz, Jesse Anderton, Kevin Jamieson, Alice Wang, Hugues Bouchard, and Javed Aslam. 2022. Identifying New Podcasts with High General Appeal Using a Pure Exploration Infinitely-Armed Bandit Strategy. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 134–144.
- [4] Hamsa Bastani, David Simchi-Levi, and Ruihao Zhu. 2022. Meta dynamic pricing: Transfer learning across experiments. *Management Science* 68, 3 (2022), 1865–1881.
- [5] Soumya Basu, Branislav Kveton, Manzil Zaheer, and Csaba Szepesvári. 2021. No regrets for learning the prior in bandits. *Advances in Neural Information Processing Systems* 34 (2021), 28029–28041.
- [6] James Bennett and Stan Lanning. 2007. The Netflix Prize. In *Proceedings of KDD Cup '07*. San Jose, CA, USA.
- [7] Veronika Bogina and Tsvi Kuflik. 2017. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks. In *RecTemp@RecSys*. 57–59.
- [8] Stefano Caria, Maximilian Kasy, Simon Quinn, Soha Shami, Alex Teytelboym, et al. 2020. An adaptive targeted field experiment: Job search assistance for refugees in Jordan. (2020).
- [9] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems* 24 (2011).
- [10] Antoine Dedieu, Rahul Mazumder, Zhen Zhu, and Hossein Vahabi. 2018. Hierarchical Modeling and Shrinkage for User Session Length Prediction in Media Streaming. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 607–616.
- [11] Thomas Desautels, Andreas Krause, and Joel W Burdick. 2014. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research* 15 (2014), 3873–3923.
- [12] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. Omnipress.
- [13] Aditya Grover, Todor Markov, Peter Attia, Norman Jin, Nicolas Perkins, Bryan Cheong, Michael Chen, Zi Yang, Stephen Harris, William Chueh, et al. 2018. Best arm identification in multi-armed bandits with delayed feedback. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 833–842.
- [14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (second ed.). Springer.
- [15] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. 2015. Focusing on the Long-Term: It's Good for Users and Business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) (KDD '15). Association for Computing Machinery, New York, NY, USA, 1849–1858. <https://doi.org/10.1145/2783258.2788583>
- [16] Liangjie Hong and Mounia Lalmas. 2019. Tutorial on online user engagement: Metrics and optimization. In *Companion Proceedings of The 2019 World Wide Web Conference*. 1303–1305.
- [17] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. 2013. Online learning under delayed feedback. In *International Conference on Machine Learning*. PMLR, 1453–1461.
- [18] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. 2018. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 133–142.
- [19] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700* (2019).
- [20] Mounia Lalmas, Heather O'Brien, and Elad Yom-Tov. 2014. Measuring user engagement. *Synthesis lectures on information concepts, retrieval, and services* 6, 4 (2014), 1–132.
- [21] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 661–670.
- [22] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research* 18, 1 (2017), 6765–6816.
- [23] David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. 2019. Multi-armed bandits in the wild: Pitfalls and strategies in online experiments. *Information and Software Technology* 113 (2019), 68–81.
- [24] Lucas Maystre, Dan Russo, and Yu Zhao. 2023. Optimizing Audio Recommendations for the Long-Term: A Reinforcement Learning Perspective. (Feb. 2023). Preprint, arXiv:2302.03561v2 [cs.LG].
- [25] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Nandits. In *Proceedings of RecSys '18*. Vancouver, BC, Canada.
- [26] Ross L Prentice. 1989. Surrogate endpoints in clinical trials: definition and operational criteria. *Statistics in Medicine* 8, 4 (1989), 431–440.
- [27] Chao Qin and Daniel Russo. 2022. Adaptivity and confounding in multi-armed bandit experiments. *arXiv preprint arXiv:2202.09036* (2022).
- [28] Carl Edward Rasmussen, Christopher KI Williams, et al. 2006. *Gaussian processes for machine learning*. Vol. 1. Springer.
- [29] F. Ricci, L. Rokach, and B. Shapira. 2015. *Recommender Systems Handbook* (second ed.). Springer.
- [30] Daniel Russo and Benjamin Van Roy. 2016. An information-theoretic analysis of Thompson sampling. *The Journal of Machine Learning Research* 17, 1 (2016), 2442–2471.
- [31] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.
- [32] Steven L Scott. 2010. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26, 6 (2010), 639–658.
- [33] Max Simchowitz, Christopher Tosh, Akshay Krishnamurthy, Daniel J Hsu, Thodoris Lykouris, Miro Dudik, and Robert E Schapire. 2021. Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems* 34 (2021), 26382–26394.
- [34] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.
- [35] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995* (2009).
- [36] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.
- [37] Thi Ngoc Trang Tran, Alexander Felfernig, Christoph Trattner, and Andreas Holzinger. 2021. Recommender systems in the healthcare domain: State-of-the-art and research issues. *Journal of Intelligent Information Systems* 57 (2021), 171–201.
- [38] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachler, Ivana Bosnic, and Erik Duval. 2012. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *Journal of Intelligent Information Systems* 5, 4 (2012), 318–335.
- [39] Han Wu and Stefan Wager. 2022. Partial Likelihood Thompson Sampling. *arXiv preprint arXiv:2203.00820* (2022).
- [40] Han Wu and Stefan Wager. 2022. Thompson Sampling with Unrestricted Delays. *arXiv preprint arXiv:2202.12431* (2022).
- [41] Qingyun Wu, Hongning Wang, Liangjie Hong, and Yue Shi. 2017. Returning is believing: Optimizing long-term user engagement in recommender systems. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 1927–1936.
- [42] Jeremy Yang, Dean Eckles, Paramveer Dhillon, and Sinan Aral. 2020. Targeting for long-term outcomes. *arXiv preprint arXiv:2010.15835* (2020).
- [43] Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. 2019. Deep reinforcement learning for search, recommendation, and online advertising: a survey. *ACM SIGWEB newsletter* Spring (2019), 1–15.
- [44] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.
- [45] Lixin Zou, Long Xia, Zhuoye Ding, Jiaying Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2810–2818.

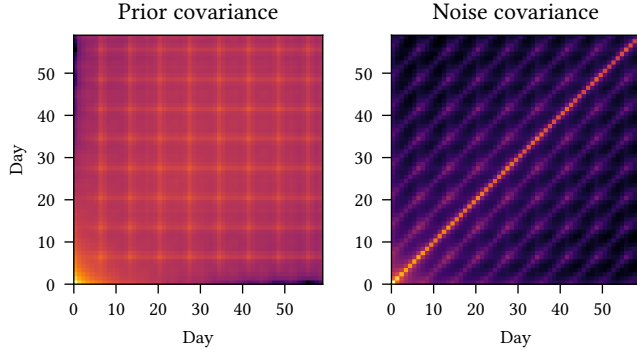


Figure 8: A visualization of the prior and noise covariance matrices.

A COMMENTS ON REWARD MODEL

In Section 3.2, we mention that if the definition of the problem at hand does not directly imply a linear relation between a given set of intermediate observations and a long-term reward of interest, one might try to learn a model of target outcomes y by solving a regression problem

$$\arg \min_{\mathbf{w}} \sum_{(z,y) \in \mathcal{D}} (y - \mathbf{w}^\top \mathbf{z})^2$$

on some historical data \mathcal{D} . In general, the reward $r = \mathbf{w}^\top \mathbf{z}$ will no longer be identical to the true long-term target y .

In this case, the reward can be thought of as a *surrogate index*, as defined in Athey et al. [2]. Provided that several assumptions hold, this approach is principled. Among others, y needs to be independent of the selected action a given r (a.k.a. the surrogacy assumption). This assumption requires \mathbf{z} to contain sufficient information on a as it relates to y . Furthermore, the reward $r = \mathbf{w}^\top \mathbf{z}$ learned on historical training data should generalize to data coming in during evaluation (a.k.a. the comparability assumption). In practice, it might be important to test these assumptions empirically.

A.1 Non-Linear Extension

The assumption that the reward r is linear in the trace \mathbf{z} is not as restrictive as it might appear at first sight. It is easy to extend the model to capture non-linear relationships between \mathbf{z} and r , while staying in the same linear-Gaussian framework that we rely on throughout Section 3.

As a concrete example, consider a reward r that depends on $\mathbf{z} \in \mathbb{R}^2$ in a non-linear way, for example $r = z_1^2 - 3z_2 + 6z_1z_2$. We can augment the trace into a new vector $\mathbf{z}' = (z'_1, z'_2, z'_3, z'_4, z'_5) = (z_1, z_2, z_1^2, z_2^2, z_1 \cdot z_2)$. Now, we can represent any quadratic relationship between \mathbf{z} and r as a linear relationship between \mathbf{z}' and r . In particular, our example yields $r = \mathbf{w}^\top \mathbf{z}'$ with $\mathbf{w} = (0, -3, 1, 0, 6)$. By instantiating the reward model over \mathbf{z}' instead of \mathbf{z} , we can thus model non-linear (quadratic) relations between intermediate outcomes and long-term reward. This idea can be extended to higher-order polynomials or (perhaps better) to regression splines [14], and capture non-linear relationships in a flexible way.

B COVARIANCE VISUALIZATIONS

In Figure 8, we show visualizations of the prior and noise covariance matrices \mathbf{V} and Σ obtained by training the model on the data

described in Section 4.1.1. In the prior covariance matrix we see a clear weekly trend, and whilst the entries around the first few days of activity dominate, there is still a rich covariance structure across the whole 60-day period. From the noise covariance, we can conclude that the daily observations are clearly not independent, but there is still a significant degree of day-to-day variability which is not explained.

Technical Details on Figure 4. This figure provides an additional perspective on the matrices \mathbf{V} and Σ . On Figure 4 (left) we consider the noise covariance matrix \mathbf{V} . We have

$$\begin{aligned} \mathbb{V}[r \mid \mathbf{z}_{:t}, \bar{\mathbf{z}}] &= \mathbb{V}[\mathbf{1}^\top \mathbf{z} \mid \mathbf{z}_{:t}, \bar{\mathbf{z}}] = \mathbf{1}^\top \tilde{\mathbf{V}}_t \mathbf{1} \doteq \tilde{\sigma}_t^2, \\ \tilde{\mathbf{V}}_t &= \mathbf{V}_{t+1:t+1} - \mathbf{V}_{t+1:t} \mathbf{V}_{t:t}^{-1} \mathbf{V}_{t:t+1}, \end{aligned}$$

where the last equality makes use of standard Gaussian identities [28, Section A.2]. We normalize the conditional variance by the total (unconditional) variance to obtain the fraction of total variance explained by the first t intermediate outcomes, i.e., we report $\tilde{\sigma}_t^2 / \tilde{\sigma}_0^2$.

On Figure 4 (right), we proceed similarly for the prior covariance matrix Σ , computing

$$\begin{aligned} \mathbb{V}[\bar{r} \mid \bar{\mathbf{z}}_{:t}] &= \mathbb{V}[\mathbf{1}^\top \bar{\mathbf{z}} \mid \bar{\mathbf{z}}_{:t}] = \mathbf{1}^\top \tilde{\Sigma}_t \mathbf{1} \doteq \tilde{\sigma}_t^2, \\ \tilde{\Sigma}_t &= \Sigma_{t+1:t+1} - \Sigma_{t+1:t} \Sigma_{t:t}^{-1} \Sigma_{t:t+1}, \end{aligned}$$

and we report the normalized value $\tilde{\sigma}_t^2 / \tilde{\sigma}_0^2$.

C CONTEXTUAL EXTENSION

Our methodology can be extended to the contextual setting, and we briefly sketch this extension here. For conciseness, let us consider the case of disjoint linear payoffs [21], and let us fix a single action and omit the subscript a . Instead of modeling the K -dimensional average trace $\bar{\mathbf{z}}$, we now model a $(d \times K)$ -dimensional matrix Θ . We assume that the expected reward for selecting the action is $\mathbf{x}^\top \Theta \mathbf{w}$, where \mathbf{x} is a context vector (e.g., describing a user's preferences) that can change across rounds. Intuitively, the k th column of Θ describes coefficients of the k th context-dependent average intermediate outcome.

We can extend the Bayesian filter we describe in Section 3.1 to model a belief over the random matrix Θ instead of the random vector $\bar{\mathbf{z}}$. This is achieved simply by vectorizing the matrix, that is, $\text{vec}(\Theta) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with $\boldsymbol{\mu}$ of dimension dK and Σ of dimension $dK \times dK$. We can condition the belief updates on the context using standard closed-form formulas. For simplicity, consider a full trace \mathbf{z} observed in context \mathbf{x} ; The posterior belief update is given by

$$\begin{aligned} \mathbf{A} &\leftarrow \Sigma(\Sigma + \mathbf{x}\mathbf{x}^\top \otimes \mathbf{V})^{-1} \\ \boldsymbol{\mu}' &\leftarrow \boldsymbol{\mu} + \mathbf{A}(\mathbf{x} \otimes \mathbf{z} - \boldsymbol{\mu}) \\ \Sigma' &\leftarrow \Sigma + \mathbf{A}\Sigma, \end{aligned}$$

where \otimes denotes the Kronecker product.

The simple training procedure described in Section 3.2 cannot be easily extended to the contextual case, since the quantities involved in the averages are context-dependent. Instead, we suggest using type-II maximum likelihood, a standard hyperparameter selection procedure. We leave a detailed development of a contextual version of our approach for future work.