Optimizing Audio Recommendations for the Long-Term

Lucas Maystre
Spotify
lucasm@spotify.com

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

Yu Zhao
Spotify
yzhao@spotify.com

Abstract

We study the problem of optimizing recommender systems for outcomes that realize over several weeks or months. Successfully addressing this problem requires overcoming difficult statistical and organizational challenges. We begin by drawing on reinforcement learning to formulate a comprehensive model of users' recurring relationship with a recommender system. We then identify a few key assumptions that lead to simple, testable recommender system prototypes that explicitly optimize for the long-term. We apply our approach to a podcast recommender system at a large online audio streaming service, and we demonstrate that purposefully optimizing for long-term outcomes can lead to substantial performance gains over approaches optimizing for short-term proxies.

1 Introduction

Recommendation systems are an essential component of modern online platforms. They help individuals find candidates to interview for job postings, partners to date, products to try, or media to engage with. Even though most users have recurring relationships with recommendation systems, nearly all recommendation systems rely on machine learning algorithms that are trained to optimize short-term metrics which provide, at best, an imperfect reflection of a recommendation's impact on users' long-term satisfaction.

This paper describes efforts at a large audio streaming service to optimize audio recommendations for their contribution to users' long-term listening habits. We draw inspiration from reinforcement learning (RL), which gives a formal language for studying the problem of learning across users to optimize recurring interaction with individual users. Despite the potential attractiveness of RL as a conceptual framework, there are few documented successful industrial applications. Our work provides evidence that optimizing for the longer-term can bring substantial value, but that in order to do so one needs to overcome substantial challenges that are not common in popular RL benchmarks.

1.1 Challenges in optimizing recurring user interactions

While recommendations are often optimized for a myopic measure of success, the real product and business goal is to power satisfying recurring user interactions. A hypothetical user journey is depicted in Figure 1 below. Highlighted in green on the first day is an interaction when the user discovers *Podcast X*, a hypothetical podcast that releases new episodes on a regular cadence. Sixty days later the user is still listening to *Podcast X*, and this deep connection to the content may have been nurtured by a decision to recommend the show again on intermediate days. In this view, the decision to recommend the show appears to have consequences that take months to realize. We would like to design a machine learning system that purposefully fosters a satisfying recurring experience, like this user's listening to *Podcast X*.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

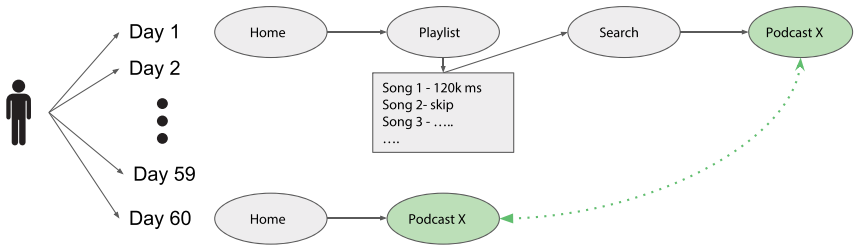


Figure 1: A depiction of a possible trajectory of user interactions over 60 days.

Despite these longer-term goals, it is plausible that a myopic recommendation strategy is nearly optimal. Perhaps the best user experience is to be presented with the items they are most likely to stream from in the moment. Also plausible, however, is that a user's streaming decisions in the short-term reflect their listening impulses, and a different recommendation criterion could help them maintain diverse and satisfying listening habits. Testing the latter hypothesis requires a working, testable methodology that purposefully optimizes for longer-term goals. Unfortunately, efforts to develop such a system run into crucial challenges. Figure 1 brings several of these into focus.

Measurement Consider some holistic outcome of the 60-day user journey, such as whether the user remains a subscriber after 60 days, or the total number of minutes they listen. One approach to unbiased measurement is to randomize recommendation decisions and then assess whether users who receive certain recommendations by chance tend to have "better" outcomes. The signal-to-noise ratio is poor, however, both because user behavior is inherently noisy and because a single recommendation is a tiny part of their overall 60-day experience.

Attribution After observing a positive 60 day trajectory, it is difficult to assign credit to individual actions in a coherent way. A subtlety is whether a decision is good or bad at this point in time may depend on how decisions will be made at other points in the future. Recommending the user try *Podcast X* for the first time might only be highly valuable if future recommendations are likely to resurface that content later.

Coordination Notice that in Figure 1 the user seems to transition through many different kinds of experiences. They see recommendations on the home page, but also see personalized results on the search page. Some of their interactions may be with a personalized mixed-audio playlist that might embed podcasts like *Podcast X*. Distinct, focused, teams optimize each such experience for immediate outcomes, like whether a user streams after a search result. Optimizing for the longer-term seems to require coordination among all such teams.

1.2 Summary of contributions

A primary contribution of our paper is to document a successful attempt to optimize a modern recommender system for the long-term. Drawing on intuition from Figure 1, our prototype aims to make recommendations that help users discover podcasts shows with which they are likely to deeply engage with. In Section 4, we demonstrate that such an effort can result in substantial product impact. For instance, one of our simple A/B tests compared two methods for making a single podcast recommendation: control, which optimizes the impression-to-stream rate and a treatment group, which also reasons about the downstream consequences of a first stream. For 63% of users, the two methods disagree about which show to recommend. In that case, being randomly placed in the treatment group was associated with **a 81% increase** in average listening minutes attributable to the recommendation. This algorithm is now used to power recommendations to hundreds of millions of users around the world.

A second contribution of this paper is to develop a formal model of our recommendation problem using the language of reinforcement learning. First, we provide a generic formulation of an RL problem which requires grappling with the attribution, coordination, and measurement challenges described above (Section 2). Then, we describe a practical approach to modeling a user's state in the recommendation system, as well as formal assumptions that underlie our algorithms (Section 3). With these assumptions in place, our relatively simple prototype algorithms can be understood formally as a *policy iteration update* of the status-quo recommendation algorithm currently in use.

We hope that these two contributions—providing evidence of the potential value in optimizing in the long-term together with a formal model that distills the main challenges—can help spur further progress.

1.3 Brief discussion of related work

In this abbreviated paper, we restrict our discussion to two key streams of related work. First, in recent years there have been various efforts to apply reinforcement learning to optimize recommender systems for longer-term goals. See for example [6, 8, 3, 1] . Most of this work tries to optimize for a somewhat brief time period. For instance [3] apply policy gradient methods to optimize candidate generation on Youtube over a timespan of 4-10 hours. Our prototype methods aim to optimize over a 60 day horizon, for which the unique structural modeling in Section 3 appears to be critical.

Another approach to optimizing for the long-term is to develop surrogate or proxy metrics—functions of short-term outcomes that appear to align decisions with long-term goals [2, 7]. For instance, continuing with Youtube as an example, [5] very recently demonstrated how optimizing for certain intuitive metrics like diversity can improve long-term outcomes. Our core approach is quite different. For instance, one of our testable prototypes involves training a new vector embedding of podcast shows that can be used to predict whether users with certain tastes will "stick" to the show after trying it. Nevertheless, it is worth mentioning that our causal decomposition in Section 3 is closely related to surrogacy assumptions.

2 Generic formulation

We formulate the problem of optimizing recurring user interactions using the language of reinforcement learning. At this stage, our formulation is quite generic. In Section 3, we focus on a particular use-case and in that context impose structural assumptions and build a unique state-representation.

2.1 High-level problem description

A recommendation system interacts with users across many periods. We have in mind that a period represents an entire day. We think of recommendations as being encoded in a one-dimensional feed which users scroll through in sequence. Most recommendations are skipped without much thought given. When a user does stream (i.e. listen to) an item, the recommender observes how long they listen. We think of the feed itself as being very long, storing the sequence of items a user will be recommended on a given day. Our formulation does not allow the recommender to adjust a recommendation based on a user's responses to items recommended earlier in the same period. This choice reflects that we are primarily interested in modeling impact over quite a long time horizon—like multiple months—rather than the problem of creating a rapidly responsive feed. An item may appear multiple times on the feed, reflecting that a user may see the same recommendation multiple times within a day. The same item may be recommended and streamed on many days. We have in mind an item being something like a long playlist of songs—which can be repeatedly consumed—or a podcast show—which may have new episodes released on a regular cadence.

Historical data is available from the recommender. This data reflects the performance of a *status-quo policy*: A complex algorithm that determines the choice of recommendations in various positions of the feed. Our goal is to improve upon the status-quo policy by modifying a specific component: a differentiated logic for determining the recommendation on the feed at a single, specified position. This is partly motivated by the use case of Section 4. Intellectually, it forces us to grapple with issues of coordination described in the introduction.

2.2 Optimizing repeated interactions with a single user

Consider a particular user of a recommendation system, viewed as being drawn at random from a population of users. The user creates an account at period T_0, deactivates at some period T_1, and in between they interact with the recommender on each period. We assume that a user's lifetime is geometrically distributed, with $T_1 - T_0 \mid T_0 \sim \text{Geometric}(1-\gamma)$ for some $\gamma \in (0,1)$. This choice lets us easily model a recommender system *in steady state*. We further assume that user's lifetime is independent of all else, including the recommender's decisions. A choice to disengage due to poor

recommendations can still by modeled through a user's decision to consume no content in the future due to past decisions. To simplify some notation, we require (without loss of generality) that $T_0 \leq 0$ and $T_1 \geq 0$. This means that the user exists and has not yet deactivated at time $t = 0$.

In each period $t \in \{T_0, \cdots, T_1\}$, the user is in some context $X_t \in \mathbb{X}$ and observes a collection of recommendations $A_t = (A_{t,1}, \cdots, A_{t,L}) \in \mathbb{A}^L$ where $\mathbb{A}$ is finite list of items. The user engages with item ℓ for $Y_{t,\ell} \in \mathbb{R}_+$ seconds, with $Y_{t,\ell} = 0$ indicating the item is immediately skipped. Let $Y_t = (Y_{t,1}, \cdots, Y_{t,L})$. We assume the user does not listen to items that are not recommended. The sequence of contexts $(X_t : t \in \{T_0, \cdots, T_1\})$ follows an exogenous Markov chain, representing factors like a user's age, device, or the day of the week. Formally, conditioned on X_t, X_{t+1} is independent of A_t and Y_t as well as all prior contexts, recommendations, and streaming decisions.

User behavior. The user's response to recommendations A_t is influenced by their history prior to day t, $H_t = \{(X_\tau, A_\tau, Y_\tau) : \tau \in \{T_0, \cdots, t-1\}\} \in \mathbb{H}$, the context X_t, a latent and unobservable type of the user $\omega \in \Omega$, and an unobservable shock $\xi_t \in \Xi$ drawn independently and identically across time and independently from all else. Formally, there is some fixed function ψ that determines the user response as

$$Y_t = \psi(H_t, A_t, X_t, \omega, \xi_t).$$

The variable ξ_t is meant to capture idiosyncratic randomness in user behavior. The latent type ω is meant to model persistent user preferences that are unknown to the recommender but might be partially inferred from their behavior. The dependence of future behavior on a user's history means that recommendations in one period can influence the efficacy of recommendations in the future. Our algorithms do not explicitly model ψ, but it may help the reader to understand the structural assumptions of Section 3 as, at least implicitly, restricting the form of ψ.

Exhaustive state representation. The exhaustive encoding of the state of a user includes all information available to the recommender at time t as $S_t = (X_t, H_t)$. Take $\mathbb{S}$ to be the set of possible states excluding the null state. The recommender's action A_t is necessarily a function of the state variable, and perhaps some exogenous randomness, since it has no other information on which to base its decisions. This state variable trivially obeys the Markov property $\mathbb{P}(S_{t+1} \in \cdot \mid S_t, A_t) = \mathbb{P}(S_{t+1} \in \cdot \mid S_1, A_1, \cdots, S_t, A_t)$, since S_t encodes all information in $(A_1, S_1, \cdots A_{t-1}, S_{t-1})$. Reducing this to a tractable representation is a challenge; In Section 3, we construct a fixed-length state representation that is tailored to our use case.

Recommendation policies. A policy is a function $\pi : \mathbb{S} \to \mathbb{A}^L$. We take $\pi_\ell : \mathbb{S} \to \mathbb{A}$ to be the i^{th} component of a policy $\pi : \mathbb{S} \to \mathbb{A}^L$. Our goal will be to optimize $\pi_\star$, where $\star \in [L]$ denotes the position under consideration, while treating components $\pi_{\setminus\star} = (\pi_\ell : \ell \neq \star)$ as fixed. We focus on rules $\pi_\star$ with restricted range $\mathbb{A}_\star \subset \mathbb{A}$. We think of $\mathbb{A}_\star$ as representing a particular type of content, like podcasts, whereas some elements of $\mathbb{A}$ could be quite different. Without loss of generality, take $\star = 1$ so $\pi = (\pi_\star, \pi_{\setminus\star})$.

Reward and value functions. The holistic reward function $R : \mathbb{R}_+^L \times \mathbb{A}^L \to \mathbb{R}$ associates a user's daily interactions, as captured by Y_t, A_t, with a measure of success. A policy π has value function $V_\pi : \mathbb{S} \to \mathbb{R}$ defined as

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{T_1} R(Y_t, A_t) \mid S_0 = s \right] = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(Y_t, A_t) \mid S_0 = s \right], \tag{1}$$

where the second equality uses the fact that user lifetimes follow a geometric distribution[1], the subscript π indicates that actions are selected according to π, and $s \in \mathbb{S}$ is not the null (inactive) state. If $R(Y_t, A_t) = \mathbb{1}(\sum_{\ell=1}^{L} Y_{t,\ell} > 0)$ is an indicator of activity in the period, then the sum of rewards assesses habitual use of the service. The value-to-go $V_\pi(s)$ measures whether habitual use is expected given a user's state.

Policy improvement. We focus on the problem of improving upon the status-quo policy. A key object is the state-action value function under the status-quo policy, given by the function $Q_\pi : \mathbb{S} \times \mathbb{A}^L \to \mathbb{R}$ defined as

$$Q_\pi(s, a) = \mathbb{E}_\pi \left[R(A_t, Y_t) + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a \right].$$

[1] If $\tau \sim \text{Geom}(1-\gamma)$, then $\mathbb{P}(\tau \geq t) = \gamma^t$. Similarly, by the memoryless property of geometric distributions, $\mathbb{P}(\tau \geq t \mid \tau \geq t') = \gamma^{t-t'}$ for $t \geq t'$.

This measures the expected future reward accrued when the user is in state $s \in \mathbb{S}$, recommendation action $a \in \mathbb{A}^L$ is taken, and the status-quo policy π governs interactions thereafter. Since we focus on optimizing the recommendation of a single element in the feed, and are focused on improving upon the status-quo policy π, we consider the partial state-action value function

$$Q^\star_\pi(s, a_\star) = Q_\pi(s, [a_\star, \pi_{\setminus\star}(s)]), \tag{2}$$

which assumes that the status-quo policy is used to determine other items in the feed and is used in future periods. While (2) appears to consider a single period decision-problem, it can be used to derive policies with superior performance in long-horizon problems. In particular, it can be shown that the policy iteration update π^+, defined by $\pi^+_\star(s) \in \arg\max_{a_\star \in A_\star} Q^\star_\pi(s, a_\star)$ and $\pi^+_\ell(s) = \pi_\ell(s)$ for $\ell \neq \star$, satisfies $V_{\pi^+}(s) \geq V_\pi(s)$ for every $s \in \mathbb{S}$. At a high-level, our goal is to find π^+.

2.3 Learning across users

Our work fits more naturally into an *offline* reinforcement learning formulation. We are given access to a batch of trajectories of user interactions

$$\mathcal{D} = \{(X^u_t, A^u_t, Y^u_t) : t \in \{T^u_0, \cdots, T^u_1\}\}_{u \in U},$$

ranging over users in a finite set U whose interactions were with the status-quo policy π. Our ultimate system will need to involve quantities that can be estimated using such data. For pedantic reasons[2], we assume that $\pi_\star$ is a strictly randomized. That is, we assume $\mathbb{P}_\pi\left(A_{t,\star} = a_\star \mid S_t\right) > 0$ for each $a_\star \in \mathbb{A}_\star$. Much of the offline RL literature calls π either a "behavioral policy", or a "logging policy". We use the term status-quo policy, to emphasize that it represents not just the data collection rule but also the natural benchmark against which the performance of other policies is judged.

3 A pragmatic decomposition

In this section, we specify reasonable assumptions on behavior, on the reward, and on the state, that make the generic problem described in Section 2 tractable. The resulting approach builds on the standard myopic approach (e.g., optimizing for immediate streams), preserving many of its strengths, while systemically optimizing for longer term performance.

The direct impact of a decision to recommend a particular content on a given day is an increase in the user's likelihood of listening on that day. The direct impact of their consumption on that day is a change to their relationship with *that content*, represented as a particular component of a user's "content-state" (Section 3.3). A deepened relationship with a particular content is expected to be predictive of greater future rewards. A key intuition underlying this design is that a user's engagement with a particular item—for instance their first discovery of a podcast show—should be quite predictive of changes to their future engagement with that particular content.

3.1 Separable rewards

We consider reward functions that are based on a user's engagement on period t, defined as a vector $C_t \in \mathbb{R}^{|\mathbb{A}|}$ with components

$$C_{t,a} = \textstyle\sum_{\ell=1}^L Y_{t,\ell} \mathbb{1}(A_{t,\ell} = a).$$

Our methodology requires immediate rewards are *additively separable* functions of consumption.

Assumption A1. *There exists a function $r : \mathbb{R}_+ \to \mathbb{R}$ and weights $\{\lambda_a \in \mathbb{R}_+ : a \in \mathbb{A}\}$ such that $R(A_t, Y_t) = \sum_{a \in \mathbb{A}} \lambda_a r(C_{t,a})$ a. s. Overloading notation, we define $R(C_t) \doteq R(A_t, Y_t)$.*

With the choice $r(c) = c$ and $\lambda_a = 1$ for each a, single-period rewards track total minutes of engagement. A strictly concave content-level reward $r(\cdot)$ prioritizes engagement with many pieces of content over extreme engagement with a single item. The scalar weights do not play an important role in our work, but provide added flexibility that could be useful in certain applications.

[2]We wish to write $\mathbb{E}_\pi\left[R(Y_t) \mid S_t\,,\, A_{t,\star} = a_\star\right]$ to evaluate the expected reward earned under a given choice of action $a_\star$ or write $Y_{t,2} \perp A_{t,\star} \mid S_t$ to indicate that the consumption of the second recommendation is not influenced by the first. Both statements are only mathematically rigorous if a randomized policy is used.

3.2 Restricting to direct effects of a recommendation

It is natural to expect that recommending a podcast increases the chance the user listens to *that* podcast. We refer to this as the *direct* impact of a recommendation. As a simplifying assumption, we assume that recommendations have no indirect effects. In words, the next assumption states that, given what is known about a user (i.e. S_t), and given that item $a_\star$ is not recommended, discounted future reward associated with consumption of item $a_\star$ does not depend on which item is recommended.

Assumption A2 (No indirect impact of recommendation)**.** *For every* $a_\star \in \mathbb{A}_\star$*,*

$$\textstyle\sum_{\tau=0}^{\infty} \gamma^\tau r\left(C_{t+\tau,a_\star}\right) \perp A_{t,\star} \mid S_{t,}\, , A_{t,\star} \neq a_\star$$

Recommendations likely do have indirect effects, ranging from substitution effects to spillover benefits. To build a simple and pragmatic recommender system, we do not model them.

3.3 Markov-sufficient state representation

We form a state representation with three components: a fast-moving user-embedding that encodes beliefs about a user's preferences in the moment, a slow-moving user-embedding that encodes beliefs about their long-term tastes, and a content-relationship state that encodes their pattern of engagement with individual items.

Fast-moving user-embedding. We posit that there is a latent variable u_t that encodes a user's preferences in their current context. We assume access to an embedded vector $\hat{u}_t$, computed based on a user's interaction history and observable context (e.g. current device and time of day), which represents the recommender's understanding of the user's preferences at the current moment. Some modern recommender systems use deep-learning based sequence modeling trained to predict what the user is likely to engage with at the next moment [4]. The last hidden layer of such a model might be thought of as encoding $\hat{u}_t$.

Slow-moving taste-embedding. Recall that each user is associated with a latent type $\omega \in \Omega$ (Section 2.2). We think of this as representing deep personal tastes of the user that are hardly influenced by recommendations. We assume access to a an embedded vector $\hat{\omega}_t$, computed based on a user's interaction history, which represents the recomender's understanding of a user's taste. We have in mind that $\hat{\omega}_t$ changes slowly. Unlike $\hat{u}_t$, which is used to predict what the user will stream in this moment, we use the taste representation to project whether a user will form a deep habitual attachment with an item if they choose to listen to it.

Content-relationship state. A user's content state $Z_t \in \mathbb{R}^{|\mathbb{A}|\times k}$ consists of a k-dimensional embedded representation of a user's consumption history of each piece of content. Recall that $C_{t,a} = \sum_{\ell=1}^{L} Y_{t,\ell}\mathbb{1}(A_{t,\ell} = a)$ denotes consumption from item a. For Z_t to have an interpretation as a "state" variable, we assume it can be updated incrementally as $Z_{t+1,a} = \phi(Z_{t,a}, C_{t,a})$, starting from $Z_{T_0} = 0$. In general, the map ϕ could be learned by training a recurrent neural network on some surrogate prediction task.

Our next two assumptions relate to how the direct impact of a recommendation is modeled. Together, they imply that the fixed-length variable $(\hat{u}_t, \hat{\omega}_t, Z_t)$ forms a valid Markov state, replacing the need to maintain a complete history of user interactions as in the exhaustive representation S_t. First, we assume that $\hat{u}_t$ effectively encodes all information in the past that is relevant to predicting a user's short-term behavior. Formally, given that a user is currently active (i.e. $S_t \neq \emptyset$), the user vector is $\hat{u}_t$, and item A_t is recommended, the streaming decisions Y_t are independent of the state S_t.

Assumption A3 (Sufficiency of fast-moving embedding)**.** *For any period* t*, we have* $Y_t \perp S_t \mid \hat{u}_t, A_t, S_t \neq \emptyset$

Next, we assume that a user's future engagement with an item depends on the user's full state only through their current taste embedding and their next content-relationship state.

Assumption A4 (Sufficiency of taste and content states)**.** *For any item* $a^* \in \mathbb{A}_\star$ *and period* t

$$\textstyle\sum_{\tau=1}^{\infty} \gamma^\tau r(C_{t+\tau,a_\star}) \perp S_t \mid \hat{\omega}_t, Z_{t+1,a_\star}$$

In words, our assumption is that the user's response to the recommendation (as reflected by their engagement state at the next period), together with the system's estimate of the user's taste at the time when the recommendation was made, is sufficient for the purposes of projecting whether the user will form a habitual attachment to that item.

3.4 A simple policy improvement update

For an item $a \in \mathbb{A}$, a taste-embedding vector $\hat{\omega}$ and a content state $z_a \in \mathbb{R}^k$, define

$$V_\pi^{(a)}(z_a; \hat{\omega}) = \mathbb{E}_\pi \left[\sum_{t=0}^{T_1} r(C_{t,a}) \mid Z_{0,a} = z_a \,,\, \hat{\omega}_{-1} = \hat{\omega} \,,\, S_0 \neq \emptyset \right]. \tag{3}$$

For example, if $r(c) = \mathbb{1}(c > 0)$ is an indicator of consumption on a given day, $\hat{V}_\pi^{(a)}(z_a; \hat{\omega})$ measures how often a user with a given content-relationship state to that piece of content is likely to return to it in the future. Thanks to our assumptions, the state-action value function takes a simple form.

Theorem 1. *Under Assumptions A1–A4, if $C_{t,a_\star}$ has discrete support, and $S_t \neq \emptyset$, then*

$$\begin{aligned} Q_\pi^\star(S_t, a_\star) = & \underbrace{\textstyle\sum_{a \in \mathbb{A}} \lambda_a \mathbb{E}_\pi \left[\sum_{\tau=0}^{\infty} \gamma^\tau r(C_{t+\tau,a}) \mid A_t \neq a \,,\, S_t \right]}_{\textit{Independent of recomendation } a_\star} \\ & + \lambda_{a_\star} \textstyle\sum_c \underbrace{\left[\mathbb{P}\left(C_{t,a_\star} = c \mid \hat{u}_t, A_{t,\star} = a_\star\right) - \mathbb{P}\left(C_{t,a_\star} = c \mid \hat{u}_t, A_{t,\star} \neq a_\star\right) \right]}_{\textit{Impact of recomendation on item's consumption prob.}} \\ & \times \underbrace{\left[r(c) + \gamma \hat{V}_\pi^{(a_\star)}\left(\phi\left(Z_{t,a_\star}\, c\right) \,,\, \hat{\omega}_t\right) \right]}_{\textit{Long-term value of consumption } c} \end{aligned}$$

A proof is provided in Appendix A. If $C_{t,a}$ does not have discrete support, the summation can replaced with integration. Implementing the policy improvement step $\arg\max_{a \in \mathbb{A}_\star} Q_\pi^\star(S_t, a_\star)$ requires two components: A model of the causal impact of recommending an item on short-term consumption of that item and an estimator $\hat{V}_\pi^{(a_\star)}$ of the long-run reward due to a user's engagement with that item.

4 Experiments

Building on the structural assumptions of Section 3, we have developed prototypes for optimizing podcast engagement at a large online audio streaming service. They have been tested at scale, shown to offer substantial improvement over more traditional myopic recommendation strategies, and rolled out in production. For product and engineering reasons, we consider the *discovery* and *resurfacing* settings separately. In the discovery setting, the set of recommendation candidates only contains podcast shows that the user has never engaged with before. This corresponds to shows with empty content-state. In the resurfacing setting, the set of candidates only contains shows the user has engaged with before. This corresponds to shows with non-empty content-state. In the remainder of this section, we focus on the discovery setting. Additional prototypes and experiments encompassing the resurfacing setting will be described in a forthcoming version of this paper. Throughout this section, we define our reward as $r(c) = \mathbb{1}(c > 0)$. Informally, this choice favors engagement that is repeated across periods and diverse across content. When evaluating approaches, we consider additional metrics that capture various product and business goals.

4.1 Methodology

In the discovery setting, we assume that, for a given user, the set $\mathbb{A}^\star$ is such that $Z_{t,a} = 0$ for all $a \in \mathbb{A}^\star$. In our application, the set of items $\mathbb{A}$ is large, i.e., $|\mathbb{A}| \gg L$. For any given item under consideration at position $\star$, the probability that a user discovers that item at period t outside of our recommendation is very small. Thus, we approximate

$$\begin{aligned} &\mathbb{P}_\pi\left(C_{t,a_\star} > 0 \mid \hat{u}_t, A_{t,\star} = a_\star\right) \approx \mathbb{P}\left(Y_{t,\star} > 0 \mid \hat{u}_t, A^u_{t,\star} = a_\star\right), \\ &\mathbb{P}_\pi\left(C_{t,a_\star} > 0 \mid S_t, A_{t,\star} \neq a_\star\right) \approx 0. \end{aligned}$$

This also implies $V_\pi^{(a)}(0; \hat{\omega}) \approx 0$. Plugging these into Theorem 1, we obtain

$$Q^\star(S_t, a_\star) \approx \underbrace{\mathbb{P}\left(Y_{t,\star} > 0 \mid \hat{u}_t, A_{t,\star} = a_\star\right)}_{\text{impression-to-stream probability}} \times \underbrace{\left[1 + \gamma V_\pi^{(a)}(D; \hat{\omega})\right]}_{\text{long-term value}} + \text{const}, \tag{4}$$

where $D \doteq \phi(0, 1)$ is the content-state that immediately follows a discovery. This decomposition suggests the following strategy. At training time, we learn two independent models that predict the

impression-to-stream probability and long-term value, respectively, given a user-content pair. At test time, we rank candidates by using pointwise scores obtained by multiplying the predictions of the two models.

Impression-to-stream model. We model the impression-to-stream probability as

$$\mathbb{P}\left(Y_{t,\star} > 0 \mid \hat{u}_t, A^u_{t,\star} = a\right) \doteq \text{logistic}[f_\theta(\hat{u}_t, v_a)], \tag{5}$$

where $\hat{u}_t$ is a known vector describing the user, v_a is a known vector describing show a, and f_θ is a parametric function, in our case a neural network. By using a randomized policy for position $\star$, we collect a dataset $\mathcal{D}_\star = \{(\hat{u}_n, v_n, r_n) : n \in [N_\star]\}$, mapping user-show pairs $(\hat{u}_n, v_n)$ to an outcome $r_n \in \{0, 1\}$ indicating whether the impression in position $\star$ led to a stream. We train the model f_θ by minimizing the negative log-likelihood of θ under $\mathcal{D}_\star$.

Long-term value model. We collect a second dataset $\mathcal{D}_V = \{(\hat{\omega}_n, a_n, V_n) : n \in N\}$ mapping user-show pairs (ω_n, a_n) to $V_n \in \mathbb{N}$ indicating how many days the user engaged with the show starting from the day of discovery. For practical reasons, we track engagement only up to 60 days. Importantly, $\mathcal{D}_V$ includes discoveries that originate from anywhere on the service, including but not limited to discoveries induced by recommendations in position $\star$. In practice, only a small fraction of observations in $\mathcal{D}_V$ originate from $\star$. We consider two models of the long-term value,

$$V^{(a)}_\pi(D; \hat{\omega}) \doteq V^{(a)}, \qquad\qquad V^{(a)}_\pi(D; \hat{\omega}) \doteq \langle \omega, \theta_a \rangle, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product. We refer to the former as the *unpersonalized* model, and to the latter as the *personalized* model. Estimates $\{\hat{V}^{(a)}\}$ and $\{\hat{\theta}_a\}$ are obtained by the empirical mean and the least-squares solution on $\mathcal{D}_V$, respectively. The estimates $\{\hat{\theta}_a\}$ serve as a new vector embedding of items designed to encode whether users with given tastes profiles tend to "stick" to an item upon discovering it. They seem to represent fundamentally different information about items and their properties which recomendation surfaces might leverage for many tasks, beyond our product changes.

4.2 Experimental Setup

We intervene on a specific recommender system in the service's mobile app. A single podcast show is chosen among a small pool of content curated by editors, and displayed close to the top of the first screen seen by users upon app launch. The set of recommendation candidates varies by market, with $|\mathbb{A}_\star|$ ranging from 3 to 74. We design a randomized controlled trial with one control group and three treatment groups. In the control group, we select the show with the highest impression-to-stream probability (5). In the treatment groups, we select the show with the highest expected long-term impact (4). Treatment groups differ based on the long-term value model. They include the *personalized* and *unpersonalized* models in (6), as well as a variant obtained by taking the *square-root* of the unpersonalized model's prediction. The latter is a heuristic that explores a different trade-off between short-term and long-term outcomes.

Every user included in the trial sees the recommendation at most once, at the time of the first visit following the launch of the experiment. The experiment runs for two weeks, and we record users' behaviors over 60 days post-impression.[3] Our primary hypothesis is as follows: The recommendation policies used in the treatment groups will generate fewer discoveries, but they will offset this with greatly increased engagement per discovery.

4.3 Experimental Results

In total, the experimental data we collect consists of approximately 30 million impressions. Only a small fraction of these impressions result in a stream. Pre-experiment, we compute for each user counterfactual recommendations for all groups. For 37% of users, all policies choose the same show (due to the small size of the candidate pool). In this section, we report results on the 63% of impacted users (i.e., users for which at least one of the treatment policies differs from the control policy). In Figure 2 we report the average difference for three metrics. *60-day activity* corresponds to the number of days a user is active with the recommended show, over the sixty days that follow an impression. This metric matches the objective the long-term policies explicitly try to maximize. *First-stream*

[3]We choose a 60-day window for practical reasons. It is long enough to capture recurring podcast listening habits, but short enough to measure in practical experiments.

Figure 2: Average difference between treatment and control for three engagement metrics. 60-day activity and minutes capture long-term impact, first-stream captures the short-term impact.

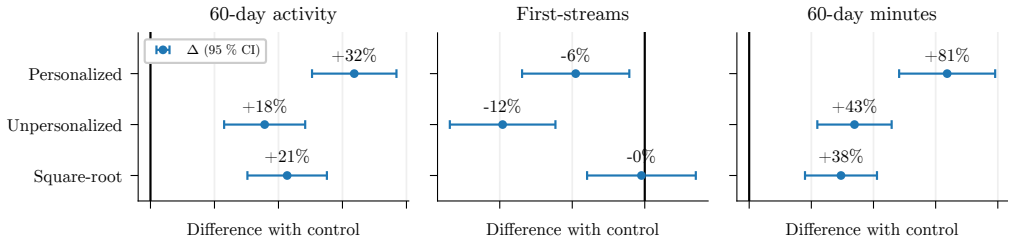
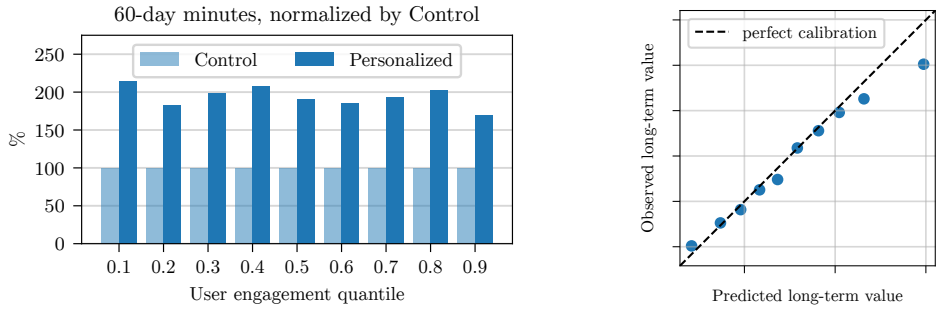


Figure 3: Left: the personalized long-term policy creates more engaged users at all engagement levels. Right: long-term value predictions data accurately match experimental outcomes.



indicates whether an impression resulted in a stream. This matches the objective of the control policy. We also report *60-day minutes*, which is similar to 60-day activity but sums up minutes instead of counting active days.

We observe that explicitly grappling with the long-term goal leads to substantial metric gains. By reasoning about the downstream impact generated by a discovery, we increase the total consumption minutes attributable to a recommendation by 81%. One might suspect that the treatment policies impact a small group of users disproportionately (e.g., by creating a small number of users that stream for a long time). Figure 3 (left) shows that this is not the case: Users who discover a show through our recommendation are affected similarly across all engagement quantiles.

Beyond raw performance gains, we also observe that optimizing for the long-term is distinctly different from optimizing for the short-term. Despite relatively small candidate sets, the long-term policies choose an action that is different from the short-term (control) policy for 63% of users. Maximizing the number of discoveries is not well aligned with optimizing for long-term engagement. In fact, policies that maximize the long-term impact trade off some discoveries for a chance to create fewer but more impactful discoveries (Fig 2, middle).

Finally, our experimental results enable us to validate Assumption A4 empirically. As shown in Figure 3, learning a long-term value model using observational data leads to predictions that hold in the experimental regime as well.

5 Conclusion

Motivated by the challenges described in the introduction, we have formulated a reinforcement learning problem in which a recommendation policy has recurring user interactions across many days while controlling only 1 out of L possible recommendations per day. With careful structural modeling, we showed that a policy iteration update to the status-quo policy takes a simple form. This lets us preserve many of the strengths of the current myopic strategy while systematically optimizing for the longer-term. We focused the remainder of this abbreviated paper on describing a single experiment at a leading a audio streaming service. Those experiments reveal a striking impact on the quality of recommendations in promoting lasting user podcast discoveries.

References

[1] M. M. Afsar, T. Crump, and B. Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[2] S. Athey, R. Chetty, G. Imbens, and H. Kang. Estimating treatment effects using multiple surrogates: The role of the surrogate score and the surrogate index. *arXiv preprint arXiv:1603.09326*, 2016.

[3] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. Top-k off-policy correction for a REINFORCE recommender system. Melbourne, Australia, Jan. 2019.

[4] C. Hansen, C. Hansen, L. Maystre, R. Mehrotra, B. Brost, F. Tomasi, and M. Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of RecSys'20*, Online, Sept. 2020.

[5] Y. Wang, M. Sharma, C. Xu, S. Badam, Q. Sun, L. Richardson, L. Chung, E. H. Chi, and M. Chen. Surrogate for long-term user experience in recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4100–4109, 2022.

[6] Q. Wu, H. Wang, L. Hong, and Y. Shi. Returning is believing: Optimizing long-term user engagement in recommender systems. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1927–1936, 2017.

[7] J. Yang, D. Eckles, P. Dhillon, and S. Aral. Targeting for long-term outcomes. *arXiv preprint arXiv:2010.15835*, 2020.

[8] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 167–176, 2018.

A Proof of Theorem 1

Proof. Without loss of generality, we take $t = 0$. Using first Assumption A1, then Assumption A2, we find

$$
\begin{aligned}
Q^{\star}_{\pi}(S_0, a_\star) &= \mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}R\left(Y_{\tau}\right) \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&= \mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}\sum_{a\in\mathbb{A}}\lambda_a r(C_{\tau,a}) \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&= \sum_{a\in\mathbb{A}}\lambda_a\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{\tau,a}) \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&= \sum_{a\in\mathbb{A}_\star}\lambda_a\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{\tau,a}) \mid A_{0,\star} \neq a\,,\, S_0\right] \\
&\quad + \lambda_{a_\star}\left(\underbrace{\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{\tau,a_\star}) \mid A_{0,\star} = a_\star\,,\, S_0\right]}_{:=(*)} - \underbrace{\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{\tau,a_\star}) \mid A_{0,\star} \neq a_\star\,,\, S_0\right]}_{:=(**)}\right).
\end{aligned}
$$

We now simplify the term (*) by applying Assumptions A3 and A4. We have,

$$
\begin{aligned}
&\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{\tau,a_\star}) \mid A_0 = a_\star\,,\, S_0\right] \\
&\quad= \mathbb{E}_{\pi}\left[r(C_{0,a_\star}) + \gamma\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{1+\tau,a_\star}) \mid A_0 = a_\star\,,\, S_0\right] \\
&\quad\overset{(a)}{=} \mathbb{E}_{\pi}\left[r(C_{0,a_\star}) + \gamma\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{1+\tau,a_\star}) \mid Z_{1,a_\star}, \hat{\omega}_0, S_0, A_{0,\star} = a_\star\right] \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&\quad\overset{(b)}{=} \mathbb{E}_{\pi}\left[r(C_{0,a_\star}) + \gamma\mathbb{E}_{\pi}\left[\sum_{\tau=0}^{\infty}\gamma^{\tau}r(C_{1+\tau,a_\star}) \mid Z_{1,a_\star}, \hat{\omega}_0\right] \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&\quad\overset{(c)}{=} \mathbb{E}_{\pi}\left[r(C_{0,a_\star}) + \gamma V_{\pi}^{(a_\star)}\left(Z_{1,a_\star}, \hat{\omega}_0\right) \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&\quad\overset{(d)}{=} \mathbb{E}_{\pi}\left[r(C_{0,a_\star}) + \gamma V_{\pi}^{(a_\star)}\left(\phi\left(Z_{0,a_\star}\,,\, C_{0,a_\star}\right)\,,\, \hat{\omega}_0\right) \mid A_{0,\star} = a_\star\,,\, S_0\right] \\
&\quad\overset{(e)}{=} \sum_{c}\mathbb{P}_{\pi}\left(C_{0,a_\star} = c \mid S_0, A_{0,\star} = a_\star\right)\left[r(c) + \gamma V_{\pi}^{(a_\star)}\left(\phi\left(Z_{0,a_\star}\,,\, c\right)\,,\, \hat{\omega}_0\right)\right] \\
&\quad\overset{(f)}{=} \sum_{c}\mathbb{P}_{\pi}\left(C_{0,a_\star} = c \mid \hat{u}_0, A_{0,\star} = a_\star\right)\left[r(c) + \gamma V_{\pi}^{(a_\star)}\left(\phi\left(Z_{0,a_\star}\,,\, c\right)\,,\, \hat{\omega}_0\right)\right]
\end{aligned}
$$

Applying the same steps to simplify (**) completes the argument. Step (a) above uses the law of iterated expectations. Step (b) follows from Assumption A4. Step (c) applies the definition in (3). Step (d) applies the incremental definition of the content-state, $Z_{t+1,a} = \phi(Z_{t,a}\,,\, C_{t,a})$. Step (e) uses the fact that $Z_{0,a_\star}$ and $\hat{\omega}_0$ are known functions S_0 (which contains all information known about the user at the time a recommendation is made). Only the consumption $C_{0,a_\star}$ is random conditioned on S_0. Step (f) follows from Assumption A3, as $C_{0,a_\star}$ is a deterministic function of Y_0. □