Preference Elicitation for Music Recommendations

Ofer Meshi [* 1] **Jon Feldman** [* 2] **Li Yang** [* 1] **Ben Scheetz** [* 2] **Yanli Cai** [1] **MohammadHossein Bateni** [3]
Corbyn Salisbury [2] **Vikram Aggarwal** [1] **Craig Boutilier** [1]

Abstract

The cold start problem in *Recommender Systems (RSs)* makes the recommendation of high-quality content to new users difficult. While *Preference Elicitation (PE)* can be used to "onboard" new users, PE in music recommendation presents unique challenges to classic PE methods, including: a vast item (music track) corpus, considerable within-user preference diversity, multiple consumption modes (or downstream tasks), and a tight query "budget." We develop a PE framework to address these issues, where the RS elicits user preferences w.r.t. item *attributes* (e.g., artists) to quickly learn coarse-grained preferences that *cover* a user's tastes. We describe heuristic algorithms that dynamically select PE queries, and discuss experimental results of these methods onboarding new users in YouTube Music.

1. Introduction

Recommender Systems (RSs) play a crucial role in making content accessible to users in domains ranging from e-commerce and product recommendation to the recommendation of content such as news, video, music, and more (Abel et al., 2011; Hallinan & Striphas, 2016; Linden et al., 2003; Pal et al., 2020; Covington et al., 2016). Since RSs typically employ a user's interactions to improve future recommendations, they often face the *cold start problem*, i.e., the inability to make high-quality recommendations to new users with little-to-no history (Lam et al., 2008; Bobadilla et al., 2012). While the cold-start problem can sometimes be addressed using informative priors—assuming they exist—it is often most natural to ask the user for some preliminary information about their preferences during an *onboarding process*, using some form of explicit *preference elicitation (PE)* (Rashid et al., 2008). Elicited preferences can be used to improve a user's (initial) recommendations.

In this work, we study PE for the onboarding of new users of a music recommendation platform. Onboarding in most domains should be relatively fast and lightweight for the user, which may impose a soft "budget" on the number of PE queries that can be asked. However, music recommendation presents a number of challenges for PE-based onboarding:

- User music preferences are generally diverse, reflecting multiple interests (e.g., genres, artists, styles, instruments) that often vary with context (e.g., activity, mood, companions). PE during onboarding should provide reasonable coverage of these interests.
- Music corpora are massive. Thus, PE queries involving recommendable items (individual tracks), while precise, are too fine-grained for onboarding given the tight "budget." As such, PE using coarser-grained attributes (e.g., artists, genres) are more suitable (but must still inform track-level preferences).
- We get incomplete feedback from the user w.r.t. recommendation quality, since in any particular listening context, a user may be in the mood for a particular part of their diverse taste. Additionally, in music platforms such as YouTube Music, there are a variety of recommendation tasks that leverage elicited preferences (e.g., search, home page, radio). Hence, there is often no single downstream objective against which to optimize the PE process.
- The onboarding process should keep the user sufficiently engaged. Hence, each PE query should be perceived by the user as useful and adding value, to prevent the user from abandoning the process too soon.

We develop a framework for PE-based onboarding that addresses these challenges. We assume access to an embedding representation of items (music tracks) from which embeddings of attributes such as artists or genres are derived—we focus on artists here. PE consists of presenting artists to a user and asking them to select the artists they like (see Fig. 1). We propose algorithms that adaptively select the artists to ask about given the user's previous selections (and non-selections). To handle the diversity of user interests—given the limited query budget and multiple (sometimes

---

[*]Equal contribution [1]Google Research, Mountain View, CA USA [2]YouTube, New York, NY USA [3]Google Research, New York, NY USA. Correspondence to: Ofer Meshi <meshi@google.com>.

The Many Facets of Preference Learning Workshop at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

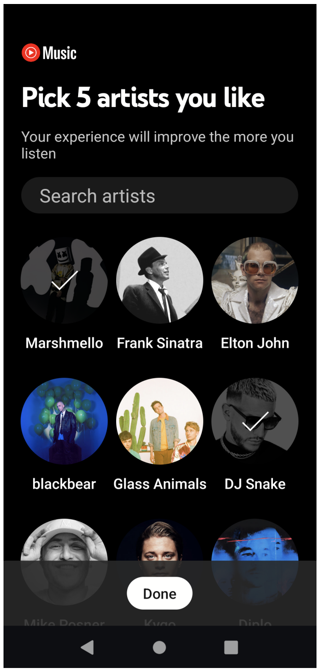

Figure 1. The artist selection interface. Users select artists they like, and skip those they don't. The scrollable interface allows for selection of as many artists as desired, with "Done" confirming the end of the onboarding session. The artists displayed as the user scrolls are selected dynamically given earlier selections and non-selections (or *skips*).

dynamic) downstream objectives—we develop a *coverage metric* that attempts to cover the user's entire range of preferences. By using artist preferences, PE covers more of the space than if track preferences were used. This coverage/precision tradeoff is essential when onboarding in domains with vast corpora like music.

We describe several experiments, conducted on the YouTube Music platform, showing the effectiveness of our method in providing domain coverage.

2. Problem Setting

We outline our problem setting and briefly describe a sampling of related work.

2.1. Recommender Domain and Assumptions

We assume an RS designed to recommend items from a large content corpus, using music recommendation as our domain of focus, where items are music tracks. Behavior-based *collaborative filtering (CF)* is a dominant approach to RS design in content domains, where methods such as matrix factorization (Salakhutdinov & Mnih, 2007) or neural CF (Beutel et al., 2018; Yang et al., 2020) are used to generate user and item embeddings in some latent space $\mathcal{X} \subseteq \mathbb{R}^d$. A user u's general affinity for item i is then given by the dot product or cosine similarity of embeddings $\phi(u)$ and $\phi(i)$.[1] Item embeddings also provide a measure of *item similarity*. We assume access to stable item embeddings.

For CF to generate a useful embedding $\phi(u)$ for user u, it requires access to behavioral data (e.g., listens, item ratings) for u. For new users with no behavior history, good recommendations are not generally possible—even a probabilistic prior will offer relatively weak recommendations given the broad variety of musical tastes across typical user populations. One way to address this *cold start problem* (Lam et al., 2008; Bobadilla et al., 2012) is to use *preference elicitation (PE)* (Keeney & Raiffa, 1993; Salo & Hamalainen, 2001; Chajewska et al., 2000; Boutilier, 2002). For new users, PE may be used as part of an *onboarding process* designed to make the service quickly usable and useful. PE methods require some class of *queries* designed to elicit information about a user's preferences, a specific semantics for user responses, and a procedure for selecting queries (which often adapts to a user's previous responses). For example, PE queries might ask users for individual item preferences (e.g., "do you like track X?"), to compare two items (e.g., "which track do you prefer, X or Y?"), or for general attribute preferences (e.g., "do you like (tracks by) artist A?" or "do you prefer genre G to genre H?"). We assume that attributes (artists) are embedded in the same latent space as tracks so a user's artist preferences informs track preferences. While attribute embeddings can be learned directly (e.g., using CF (Shi et al., 2014)), we derive them from item embeddings—an artist embedding is the listen-time-weighted average of the embeddings of their tracks (refinements of this scheme are left to future work).

Item-based elicitation is often effective with small item corpora, or when making fine-grained distinctions when the RS already has a good understanding of a user's preferences. However, it is unsuitable for onboarding in RSs with vast corpora like those in music recommendation—the granularity of item-based PE, while allowing for precision, renders sufficient coverage of the music domain infeasible during a time-constrained onboarding process. For this reason, we focus on attribute-based elicitation, specifically, asking the user for their preferences over a selection of music artists.[2] The interface we use to elicit artist preferences is shown in Fig. 1. The user can scroll to select as many artists as desired and select "Done" to terminate the onboarding process. Since this delays actual music consumption, and a user can terminate at any point, it is important that onboarding quickly and effectively gleans preferences that support

[1] Other predictive signals, history and context are often used to make more refined predictions.

[2] Other attributes could include genres, sub-genres, albums, eras, etc., and combinations of these. We leave exploration of other query and attribute types to future work.

downstream recommendations.[3]

With music RSs, a given user u will request many recommendations over time, often across a variety of contexts (e.g., engaged in different activities, in different moods or locations, with different companions, etc.). Moreover, the range of downstream recommendation tasks induces additional variability in user preferences. This, together with the inherent diversity of tastes exhibited by any given user, means that a user's preferences generally vary considerably.[4] This variety suggests that obtaining broad (albeit coarse-grained) coverage of a user's range of preferences is more important during onboarding than obtaining a precise, but narrow view, and influences the design of our algorithms below. Of course, even the space of *artists* is voluminous, so the artist queries themselves must be chosen judiciously. Just as importantly, the PE method must keep users engaged sufficiently to prevent early termination, before the desired coverage is attained.

2.2. Related Work

Given the rich history of preference elicitation in the broader decision analysis, marketing science and AI literatures, and in RSs specifically, we provide only a brief discussion of some of the more relevant themes. Both item-based (Chajewska et al., 2000; Boutilier, 2002; Viappiani & Boutilier, 2010) and attribute-based (Viappiani et al., 2006; Chen & Pu, 2012) elicitation have been widely studied in AI, decision analysis and non-content-based RSs. In content RSs, the cold-start problem has been addressed using various PE techniques, though primarily in an item-based fashion using notions such as value of information, information gain, or simpler heuristics (Boutilier et al., 2003; Rashid et al., 2008; Zhao et al., 2013).

McNee et al. (2003) use a non-adaptive, item-based popularity-based heuristic for onboarding (and also consider direct user item-specification), but do not consider attribute-based methods. Work on diversity in recommendations (e.g., Meymandpour & Davis (2020)) and multi-interest representations (e.g., Weston et al. (2013)) bear on our motivation for covering a broad set of user interests during onboarding, though little work explores covering this diversity during elicitation. One recent exception is bandit-style (hence item-based) approach of Parapar & Radlinski (2021) which explicitly elicits in preferences diverse fashion (though not necessarily for onboarding). The question of eliciting preferences to cover a range of downstream tasks, a key motivation for our approach, does not appear to have been explicitly studied in the literature.

3. Formulation and Algorithms

We outline our problem formulation in this section along with algorithms to implement effective PE-based onboarding. As discussed above, the range (and possibly dynamic nature) of downstream tasks/objectives, together with per-user preference diversity, suggests that domain *coverage* should serve as a key criterion for assessing our understanding of a user's preferences (see Secs. 3.2 & 3.3). The limited time "budget" associated with onboarding means that keeping the user engaged is also important (see Sec. 3.1). It also points to the use of coarse preferences using attributes—in our case, artists (see Fig. 1). Our PE algorithm for onboarding (Sec. 3.4) brings these considerations together.

3.1. PCTR Modeling

Let $\mathcal{X}$ denote the set of all artists in the domain, whose size may be in the millions. From among these, some process is used to select a smaller query set $\mathcal{Q} \subset \mathcal{X}$ of artists eligible to ask the user about during PE. This set is intended to reflect the varied musical tastes of a wide range of users across many countries; but it also contains primarily, conditional on their preference subsegments or micro-genres, more popular artists. This is to increase the odds of user familiarity when asked about them. Still, the sheer size of $\mathcal{X}$ ensures that most users will be *unfamiliar* with most artists in $\mathcal{Q}$. If a user is presented with a large number of unfamiliar artists, they are likely to abandon the session providing little, or no, preference information. Thus the queried artists must be selected with care.

To address this problem, we train a *pCTR model* to predict an artist's *click-through rate (CTR)*, or probability that a user will select that artist. The pCTR model is then used to prioritize the presentation of artists that the user is likely to select, increasing the odds that the user sees a significant numbers of artists they know and they like. Such pCTR models are common in RSs (e.g., McMahan et al. (2013)), and typically depend on artist features (genre, popularity, etc.), certain user features, and the user's past selections/non-selections in the onboarding process. The latter is one reason that our PE queries (i.e., artists presented during onboarding) are selected dynamically and adaptively. In this work we use a Generalized Additive Model for the CTR prediction (Hastie et al., 2009). Letting h_u denote user u's history of selections/non-selections during onboarding at any point in time, the pCTR for artist i is denoted $r_i(h_u)$.

If pCTR is the sole criterion used to rank candidate artists, the PE process tends to ask the user about artists that are

[3] How different downstream RS components incorporate elicited artist preferences into their recommendations, specifically, how they are combined with other signals—and how they are mapped into track preference predictions—is typically quite intricate and beyond the scope of this article.

[4] Indeed, multi-interest (Weston et al., 2013) and context-dependent (Hansen et al., 2020) user representations should generally be better-suited to music domains than standard CF models.

very similar to those previously selected—unsurprisingly, similarity to previously selected artists is a strong predictor of artist selection, and is an input feature to the pCTR model. This tends to lead to "deep dives" into narrow genres or groups of very similar artists. While this induces more artist selections in the near-term, it does little to enhance our understanding of a user's diverse tastes. Moreover, it can be perceived by the user as lacking value, thus increasing the odds of early termination (hence decreasing selections in the long run). Since the information contained in pCTR-based artist selections quickly deteriorates as onboarding progresses, we augment this objective next.

3.2. Coverage

Another important consideration is the coverage of a user's (potentially diverse) tastes, across contexts and downstream tasks. This requires that the user be shown a set of artists that spans the track/embedding space. To this end, we first assume access to a set of *target points* $\mathcal{T}$ in embedding space, points which are well-spread throughout the space. In principal, this could be the entire set $\mathcal{X}$, but there are reasons to make this set smaller as we discuss below. There could be many strategies for constructing $\mathcal{T}$, we describe one such construction in Section 3.3. Given the target set, we define the *coverage* of a set of artists $\mathcal{A} \subseteq \mathcal{X}$ w.r.t. $\mathcal{T}$ as follows. For any artist i, let $x_i \in \mathbb{R}^d$ denote its embedding. Standard *(cosine) similarity* of two artists is defined as $S(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\|\|x_j\|} \in [-1, 1]$. We then define the *coverage score* of $\mathcal{A}$ (w.r.t. $\mathcal{T}$) as:

$$C(\mathcal{A}; \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max_{x \in \mathcal{A}} S(t, x). \tag{1}$$

This reflects that the coverage of a target $t \in \mathcal{T}$ is given by the artist in $\mathcal{A}$ most similar to it, and that set coverage is the average over all targets. A higher coverage score implies that $\mathcal{A}$ contains artists that are more similar (closer) to more targets, with maximum coverage $C(\mathcal{A}; \mathcal{T}) = 1$ attained by any $\mathcal{A} \mid \mathcal{T} \subseteq \mathcal{A}$. To simplify notation we sometimes write $C(\mathcal{A})$, when $\mathcal{T}$ is fixed.

Notice that C is monotone, since $C(\mathcal{A} \cup \{x\}) \geq C(\mathcal{A})$ (defining $C(\emptyset) = -1$), and is similar to the notion of facility location coverage (Bateni et al., 2018). C is also submodular. Letting $x(t) = \operatorname{argmax}_{x \in \mathcal{A}} S(t, x)$, this can be shown by observing:

$$C(\mathcal{A}) + C(\mathcal{A} \cup \{x'\} \cup \{x''\}) \leq C(\mathcal{A} \cup \{x'\}) + C(\mathcal{A} \cup \{x''\})$$

$$\begin{aligned}
\Leftrightarrow & \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} S(t, x(t)) \\
& \quad + \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max\{S(t, x'), S(t, x''), S(t, x(t))\} \\
& \leq \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max\{S(t, x'), S(t, x(t))\} \\
& \quad + \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \max\{S(t, x''), S(t, x(t))\} \\
\Leftrightarrow & \sum_{t \in \mathcal{T}} S(t, x(t)) + \max\{S(t, x'), S(t, x''), S(t, x(t))\} \\
& \leq \sum_{t \in \mathcal{T}} \max\{S(t, x'), S(t, x(t))\} + \max\{S(t, x''), S(t, x(t))\}
\end{aligned}$$

Now, define:

$$\begin{aligned}
\mathcal{T}^0 &= \{t \mid S(t, x(t)) \geq \max\{S(t, x'), S(t, x'')\}\} \\
\mathcal{T}' &= \{t \mid S(t, x') \geq \max\{S(t, x(t)), S(t, x'')\}\} \\
\mathcal{T}'' &= \{t \mid S(t, x'') \geq \max\{S(t, x(t)), S(t, x')\}\}
\end{aligned}$$

In case of ties, we assign each target to one of the groups, so $\mathcal{T}^0$, $\mathcal{T}'$, and $\mathcal{T}''$ are disjoint. We then have that

$$\begin{aligned}
\Leftrightarrow & \sum_{t \in \mathcal{T}^0} 2S(t, x(t)) + \sum_{t \in \mathcal{T}'} S(t, x(t)) + S(t, x') \\
& \quad + \sum_{t \in \mathcal{T}''} S(t, x(t)) + S(t, x'') \\
& \leq \sum_{t \in \mathcal{T}^0} 2S(t, x(t)) \\
& \quad + \sum_{t \in \mathcal{T}'} S(t, x') + \max\{S(t, x''), S(t, x(t))\} \\
& \quad + \sum_{t \in \mathcal{T}''} S(t, x'') + \max\{S(t, x'), S(t, x(t))\} \\
\Leftrightarrow & \sum_{t \in \mathcal{T}'} S(t, x(t)) + \sum_{t \in \mathcal{T}''} S(t, x(t)) \\
& \leq \sum_{t \in \mathcal{T}'} \max\{S(t, x''), S(t, x(t))\} \\
& \quad + \sum_{t \in \mathcal{T}''} \max\{S(t, x'), S(t, x(t))\},
\end{aligned}$$

which holds element-wise.

Though submodular function maximization is NP-hard, the greedy algorithm, which iteratively adds the best item to the set given those already in it, has a $(1 - 1/e)$-approximation guarantee for monotone functions (Nemhauser et al., 1978). In fact, no polynomial-time algorithm may achieve a better approximation guarantee unless P = NP (Feige, 1998; Nemhauser & Wolsey, 1978). To this end, we define the *coverage gain* of an artist x w.r.t. some $\mathcal{A}'$:

$$G(x, \mathcal{A}'; \mathcal{T}) = C(\mathcal{A}' \cup \{x\}; \mathcal{T}) - C(\mathcal{A}'; \mathcal{T}).$$

3.3. Target Set Construction

We now describe our approach to computing the target set $\mathcal{T} \subset \mathcal{X}$. Recall that $\mathcal{T}$ should spans the embedding space well, i.e., each point in the space $x \in \mathcal{X}$ should have some $t \in \mathcal{T}$ close to it. Thus, if, during onboarding, we ask a user about a set of artists that cover the target set well, we will have gained sufficient information to offer reasonable initial recommendations.

We begin the process with a large set of points, in this case, the set $\mathcal{X} \subset \mathbb{R}^d$ of all artists. A point $x \in \mathcal{X}$ is *τ-covered* by an artist $t \in \mathcal{T}$ if $S(t, x) \geq \tau$ for some threshold τ. Given a budget b bounding the size of $\mathcal{T}$, our goal is to find a set $\mathcal{T} \subset \mathcal{X}$ that maximizes the fraction of τ-covered points in $\mathcal{X}$. That is, we aim to maximize

$$f(\mathcal{T}) = \frac{|\{x \in \mathcal{X} \mid \max_{t \in \mathcal{T}} S(t, x) \geq \tau\}|}{|\mathcal{X}|},$$

subject to $|\mathcal{T}| = b$. This problem is an instance of the *set cover optimization problem* (or *maximum k-coverage*). This problem is also NP-hard and, unless P = NP, does not admit a polynomial-time approximation better than $1 - 1/e$ (Feige, 1998). As a special case of submodular function maximization, a greedy algorithm to target selection ensures a $1 - 1/e$ approximation, matching the above hardness result. In practice, the approximation ratio is often within few percentage points of the optimum.

There is a direct trade-off between the number of targets b, and the fraction of $\mathcal{X}$ τ-covered by $\mathcal{T}$. While we want to cover as many points in $\mathcal{X}$ as possible, b cannot be too large for computational reasons (see Section 3.4). In our music domain, we are more likely to attain good coverage for a user u if we are able to query artists with which u is familiar, similar to our motivation for using pCTR above. However, since the target set is user-independent, we use the population-level playtime (or popularity) w.r.t. existing users as a prior over familiarity. Rather than maximize coverage $f(\mathcal{T})$, we use the playtime score as a weight and instead maximize the *weighted coverage* of $\mathcal{T}$:

$$w(\mathcal{T}) = \frac{\sum_{x \in \mathcal{X} \mid \max_{t \in \mathcal{T}} S(t,x) \geq \tau} w(x)}{\sum_{x \in \mathcal{X}} w(x)},$$

subject to $|\mathcal{T}| = b$, where $w(x)$ is the *weight* of artist x, i.e., their total historical playtime. (Note that other metrics can be used as weights.) This is an instance of the *weighted set cover problem* with weights $w(x)$ and is also submodular, thus is well-approximated by the greedy algorithm.

Using playtime weights, in Figure 2 (top) we show the fraction of playtime covered $w(\mathcal{T})$ as a function of the size of the target set in our YouTube Music domain. With $b = 4000$ targets, we cover more than 90% of playtime at $\tau = 0.9$. By contrast, Figure 2 (bottom) shows that with the same $b = 4000$, we only cover roughly 25% of all (unweighted) artists $\mathcal{X}$ at $\tau = 0.9$. This may be undesirable for users with niche tastes. For example, a small number of extremely popular artists skew the playtime distribution significantly. To compensate for this effect, we set $w(x)$ to playtime raised to some power less than one—the square root of playtime provides a suitable trade-off in practice.

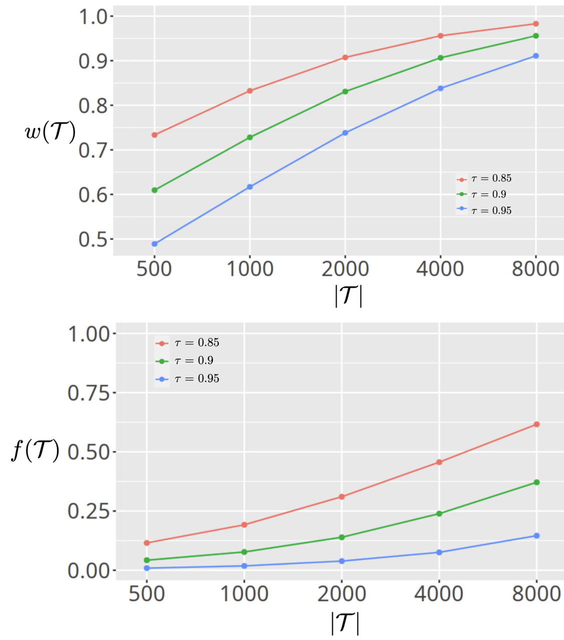

Figure 2. Tradeoff between the number of targets and the fraction covered. Top: play-time cover, bottom: artist cover.

3.4. Algorithm

We combine the considerations above into an algorithm for the dynamic selection of artists to present to a user during onboarding. Our algorithm is a simple score-and-sort procedure which scores artists dynamically, using a weighted sum of pCTR and coverage. Specifically, the score of artist $x_i \in \mathcal{Q}$ given history h_u of user u is

$$Score(x_i; h_u) = r_i(h_u) + \lambda G(x_i, \mathcal{A}(h_u); \mathcal{T}), \quad (2)$$

where $\mathcal{A}(h_u)$ is the set of artists shown to the user (either selected or unselected), and λ is a trade-off parameter.

Notice that r_i is a complex set function as pCTR input features depend on previously displayed artists, as well as the (non-deterministic) user responses (selection/non-selection). Hence, the combined score Equation (2) is not generally submodular. If the pCTR score $r_i(h_u)$ is significantly smaller than the coverage term $\lambda G(x_i, \mathcal{A}(h_u); \mathcal{T})$ for all x_i, then $Score(x_i; h_u)$ is approximately submodular. In this case, Horel & Singer (2016) show that the greedy algorithm can achieve approximation guarantees under some assumptions.

In practice, we send a *batch* of artists to the user's device, as sending artists one-by-one induces unacceptable latency. We do not address the batch optimization problem here. Instead we compute the scores for all candidate artists in $\mathcal{Q}$ and construct a batch of size k by collecting the k top-scoring artists. When the user selects an artist or scrolls past

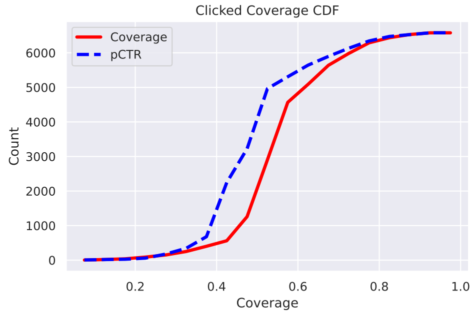

Figure 3. Clicked coverage scores.

the current batch, we construct the next batch (conditioned on the updated user history) and send it to their device. We leave batch optimization to future work.

Choosing the next artist for presentation requires computing $Score(x_i; h_u)$ for all (remaining) artists $x_i \in \mathcal{Q}$ and sorting these artists by their scores.[5] The complexity of the latter step is $O(|\mathcal{Q}| \ln |\mathcal{Q}|)$ per iteration, which can be reduced to $O(k|\mathcal{Q}|)$ for batch size k. To compute $Score(x_i; h_u)$, we must compute the pCTR score $r_i(h_u)$ and the coverage score $C(\mathcal{A}(h_u) \cup \{x_i\}; \mathcal{T})$. pCTR $r_i(h_u)$ is obtained by a single call to the pCTR model. For coverage, we check which targets x_i is most similar to, i.e., $S(t, x_i) > \max_{x \in \mathcal{A}(h_u)} S(t, x)$. Therefore, we need to compute $|\mathcal{T}|$ similarity scores $S(t, x_i)$ for each candidate artist x_i, so overall computational cost is $O(|\mathcal{Q}| \cdot |\mathcal{T}|)$ per iteration (or batch). Overall, $O(|\mathcal{Q}| \cdot |\mathcal{T}|)$ is the dominant term, which means that keeping $|\mathcal{T}|$ small significantly reduces complexity.

4. Results

In this section we describe experimental results from preliminary A/B tests onboarding new YouTube Music users with one of two algorithms: the first is `pCTR`, which uses only the pCTR score $r_i(h_u)$ to dynamically rank artists during onbaording PE (i.e., Equation (2) with $\lambda = 0$); the second is `Coverage`, which uses the combined score for ranking (Equation (2) with $\lambda > 0$).

We first assess whether `Coverage` achieves better coverage than `pCTR`. To this end, we examine the preferences provided by users with each algorithm and compute their coverage score (Equation (1)), where $\mathcal{A}$ is taken to be the set of artists *selected* by the user. We find that `Coverage` indeed achieves a higher average coverage score of 0.57 compared to 0.51 for `pCTR`, i.e., inducing +12.3% improvement in coverage, with 95% confidence interval [11.5%, 13.2%]. To give a sense of the distribution, Figure 3 shows CDFs of coverage scores for both algorithms, demonstrating that `Coverage` shifts to higher coverage values.

Apart from coverage, we examine various metrics that assess preference elicitation effectiveness of the two methods, specifically, differences in the average number of artists selected per user, the average number of artists viewed, and the number of users with at least one artist selection. We observe a slightly greater number of selections for `Coverage` compared to `pCTR`— $+1.24\%$ selections (95% CI $[-0.32\%, 2.80\%]$)—and more users with at least one selection— $+0.49\%$ (95% CI $[-0.23\%, 1.21\%]$). At the same time, the number of *shown* artists is actually slightly lower for `Coverage`, -0.08% (95% CI $[-0.97\%, 0.80\%]$), which means that average CTR is higher using `Coverage` than `pCTR`, despite the fact that `pCTR` optimizes only for (immediate) selection. We note, however, that none of these differences is statistically significant (0 is inside the CIs); we therefore conclude that the increased coverage of `Coverage` is obtained without additional effort from the user in terms of time spent or number of selections.

Without going into details on how user preferences are used downstream, we note that we do see evidence of improved initial recommendations with `Coverage` compared to `pCTR`. For example, the number of active users on the first day after onboarding is higher $+0.3\%$ (95% CI $[0.04\%, 0.55\%]$).

5. Conclusion

We have presented a framework for eliciting user preferences over domain-specific attributes—artists in the music domain examined—during the onboarding of new users of an RS. We argued that one should explicitly trade-off immediate user engagement with the quality of information collected, and to this end proposed an objective that combines pCTR with embedding space coverage. We presented efficient algorithms for choosing artists to display to the user, and showed that some of them enjoy favorable approximation guarantees. Finally, we showed that our approach leads to gains in primary performance metrics in the real-world onboarding of new users of YouTube Music.

There are a number of interesting future directions. It will be interesting to tackle the batch optimization problem described in Section 3.4. We plan to test alternative artist selection strategies and compare them to the one explored here. One such method maintains a probability distribution over a user's preferences, selecting artists using expected reduction in entropy. Our coverage objective is more tractable since we do not have to model the distribution, but the entropy-based objective should perform better w.r.t. some metrics, inducing an interesting tradeoff. Additional probabilistic objectives include expected value of information w.r.t. post-onboarding recommendation tasks.

[5]We do not consider the cost of computing $\mathcal{T}$, since it is a one-time computation that only needs to be updated periodically.

References

Abel, F., Gao, Q., Houben, G.-J., and Tao, K. Analyzing user modeling on twitter for personalized news recommendations. In Konstan, J. A., Conejo, R., Marzo, J. L., and Oliver, N. (eds.), *User Modeling, Adaption and Personalization*, 2011.

Bateni, M., Esfandiari, H., and Mirrokni, V. S. Optimal distributed submodular optimization via sketching. In Guo, Y. and Farooq, F. (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2018)*, pp. 1138–1147. ACM, 2018.

Beutel, A., Covington, P., Jain, S., Xu, C., Li, J., Gatto, V., and Chi, E. H. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM-18)*, pp. 46–54, Marina Del Rey, CA, 2018.

Bobadilla, J., Ortega, F., Hernando, A., and Bernal, J. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 26:225–238, 2012.

Boutilier, C. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 239–246, 2002.

Boutilier, C., Zemel, R., and Marlin, B. Active collaborative filtering. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pp. 98–106, 2003.

Chajewska, U., Koller, D., and Parr, R. Making rational decisions using adaptive utility elicitation. pp. 363–369, Austin, TX, 2000.

Chen, L. and Pu, P. Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1):125–150, 2012.

Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, 2016.

Feige, U. A threshold of ln *n* for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi: 10.1145/285055.285059.

Hallinan, B. and Striphas, T. Recommended for you: The netflix prize and the production of algorithmic culture. *New Media & Society*, 18(1):117–137, 2016.

Hansen, C., Hansen, C., Maystre, L., Mehrotra, R., Brost, B., Tomasi, F., and Lalmas, M. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 53–62, 2020.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Horel, T. and Singer, Y. Maximization of approximately submodular functions. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, 2016.

Keeney, R. L. and Raiffa, H. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.

Lam, X. N., Vu, T., Le, T. D., and Duong, A. D. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, 2008.

Linden, G., Smith, B., and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Distributed Syst. Online*, 4, 2003.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, 2013.

McNee, S. M., Lam, S. K., Konstan, J. A., and Riedl, J. Interfaces for eliciting new user preferences in recommender systems. In *User Modeling 2003: 9th International Conference on User Modeling (UM-03)*, pp. 178–187, Johnstown, PA, 2003.

Meymandpour, R. and Davis, J. G. Measuring the diversity of recommendations: A preference-aware approach for evaluating and adjusting diversity. *Knowledge and Information Systems*, 62(2):787–811, 2020.

Nemhauser, G. L. and Wolsey, L. A. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.*, 3(3):177–188, 1978. doi: 10.1287/moor.3.3.177. URL https://doi.org/10.1287/moor.3.3.177.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978. doi: 10.1007/BF01588971.

Pal, A., Eksombatchai, C., Zhou, Y., Zhao, B., Rosenberg, C., and Leskovec, J. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, 2020.

Parapar, J. and Radlinski, F. Diverse user preference elicitation with multi-armed bandits. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM-21)*, pp. 130–138, 2021.

Rashid, A. M., Karypis, G., and Riedl, J. Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Explor. Newsl.*, 10 (2):90–100, dec 2008.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20 (NIPS-07)*, pp. 1257–1264, Vancouver, 2007.

Salo, A. A. and Hamalainen, R. P. Preference ratios in multiattribute evaluation (prime)-elicitation and decision procedures under incomplete information. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(6):533–545, 2001.

Shi, Y., Larson, M., and Hanjalic, A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1), 2014.

Viappiani, P. and Boutilier, C. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems 23*, pp. 2352–2360, 2010.

Viappiani, P., Faltings, B., and Pu, P. Preference-based search using example-critiquing with suggestions. 27: 465–503, 2006.

Weston, J., Weiss, R. J., and Yee, H. Nonlinear latent factorization by embedding multiple user interests. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pp. 65–68. ACM, 2013.

Yang, J., Yi, X., Zhiyuan Cheng, D., Hong, L., Li, Y., Xiaoming Wang, S., Xu, T., and Chi, E. H. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Proceedings of the Web Conference (WWW-20)*, pp. 441–447, Taipei, 2020.

Zhao, X., Zhang, W., and Wang, J. Interactive collaborative filtering. In *Proceedings of the Twenty-Second ACM International Conference on Information and Knowledge Management*, pp. 1411–1420, 2013.