Exploiting Sequential Music Preferences via Optimisation-Based Sequencing

Dmitrii Moor
dmitriim@spotify.com
Spotify, London, United Kingdom

Yi Yuan
yyuan@spotify.com
Spotify, New York, USA

Rishabh Mehrotra*
erishabh@gmail.com
ShareChat, London, United Kingdom

Zhenwen Dai
zhenwend@spotify.com
Spotify, London, United Kingdom

Mounia Lalmas
mounia@acm.org
Spotify, London, United Kingdom

ABSTRACT

Users in music streaming platforms typically consume tracks sequentially in *sessions* by interacting with personalised playlists. To satisfy users, music platforms usually rely on recommender systems that learn users' preferences over individual tracks and rank the tracks within each playlist according to the learned preferences. However, such rankings often do not fully exploit the sequential nature of the users' consumption, which may result in a lower within-a-session consumption. In this paper, we model the sequential *within-a-session* preferences of users and propose an optimisation-based sequencing approach that allows for *optimally* incorporating such preferences into the rankings. To this end, we rely on interaction data of a major music streaming service to identify two most common aspects of the users' sequential preferences: (1) Position-Aware preferences, and (2) Local-Sequential preferences. We propose a sequencing model that can leverage each of these aspects optimally to maximise the expected total consumption from the session. We further perform an extensive offline and off-policy evaluation of our model, and carry out a large scale online randomised control trial with 7M users across 80 countries. Our findings confirm that we can effectively incorporate sequential preferences of users into our sequencer to make users complete more and skip less tracks within their listening sessions.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Recommender Systems, Sequential Preferences, Optimisation-Based Sequencing

*This work was partially carried out while the author was at Spotify.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615476

ACM Reference Format:
Dmitrii Moor, Yi Yuan, Rishabh Mehrotra, Zhenwen Dai, and Mounia Lalmas. 2023. Exploiting Sequential Music Preferences via Optimisation-Based Sequencing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615476

1 INTRODUCTION

Music streaming platforms enable users to engage with millions of music tracks, with a bulk of consumption being shaped by algorithmically generated recommendations. Contrary to other recommendation domains (e.g., movies, books), in a music streaming setting tracks are consumed *sequentially* within listening sessions. Thus, an ideal recommender system (RS) would not only recommend music that matches users' preferences, but sequence tracks in a way to make the music "flow smoothly" from one track to the next.

To create an enjoyable listening experience, such RS has to go beyond individual tracks and generate a coherent sequence of tracks within a *playlist* so that users consume more tracks while skipping less. To this end, RS not only needs to learn user's preferences for the individual tracks but also to understand how these preferences change as the user navigates through their session.

Generally, users' sequential preferences can be sophisticated, and may require complex models to accurately predict consumption. In our work, we focus on studying two hypotheses that any (even the most complex) sequential preferences possess:

- First, we posit that there exists a relationship between users' consumption choices and *positions* of tracks in playlists. In particular, users may start listening to a playlist if they see a familiar track at the very first position of the playlist – but may or may not continue onward. This makes the choice of the track for the first position particularly important for the user to start consuming. At the same time, reaching the further positioned tracks might be predictive of the user staying in the session for even longer. We refer to such preferences as *position-aware preferences*.
- Second, we hypothesize that the user's decision to consume a track is influenced by the previous tracks within their session. Indeed, the quality of transitions between tracks may be fundamentally linked to the user's preferences making it difficult to enjoy a sequence of tracks that significantly differ in acoustic style or tempo. We refer to such preferences as *local sequential preferences* – preferences that the user relies on when consuming tracks *within-a-session*. We contrast these preferences to the

long-term sequential preferences that indicate the evolution of the user's taste over an expanded period of time [1–3].

The goal of our work is to validate the two aforementioned hypotheses and to quantify the impact of leveraging the users' sequential preferences on their within-a-session consumption in practice. To this end, we are not aiming at designing the most complex model that can capture all subtleties of the users' sequential preferences as such models are often infeasible to serve in the real-world low-latency applications and may be challenging to integrate into the practical RS infrastructure [4]. Instead, we retreat to the class of neural models that allows us to isolate the two aforementioned aspects of the users' preferences and to measure how much each of them contributes to the users' consumption *at most*, i.e., upon being exploited in an optimisation-based way. We then examine the practical implications of leveraging sequential preferences by performing an extensive offline counterfactual study as well as a *online* large-scale randomised control trial (A/B test).

To reach our goal, we propose an optimisation-based track sequencing model that leverages the two aforementioned aspects of the users' preferences. We first optimise a parametric model against session-level interaction data collected by Spotify to model users' preferences. We then design a optimisation-based sequencer that relies on this model to maximise a user's expected total consumption. This allows us to disentangle the effect of using sequential information (our two hypotheses) from simply using a more complex preference model that better exploits non-sequential information.

To evaluate our approach we perform large-scale experiments on real world music streaming data of over 200K Spotify users, and present qualitative and quantitative insights about users' streaming behaviour. Additionally, we carry out a large-scale online experiment with 7M users across 80 countries to confirm our hypothesis that sequential preference models can help to increase music consumption within-a-session. To the best of our knowledge, our work is the first to quantify the *online* effect of leveraging sequential preferences in the music domain. Our findings demonstrate:

- Users' behaviour is inherently sequential: (1) Users consumption behaviour changes across different positions; (2) Whether a user completes a track depends not only on the user and the track but also on the previous tracks within the user's session.
- Our optimisation-based approach allows to efficiently leverage the position-aware and the local-sequential aspects of the users' preferences. This results in an increased consumption of tracks.
- Our large-scale online experiment confirms that exploiting the users' sequential preferences helps to increase track completion and decrease skips within sessions.

2 RELATED WORK

Sequential Music Listening Behaviour. There is extensive literature on the sequential nature of music consumption, e.g. [5, 6]. In [6], the authors notice that sequential user embeddings help to better predict consumption. Contrary to our work they model sequential consumption across sessions rather than within a session.

The sequential music consumption is also examined in [7] where the authors proposed a convolutional RNN for genre prediction. In contrast to our work, the authors focus on shorter track-level sequences rather than session-level ones. A similar approach for analysing track-level sequences was adopted in [8] to learn audio embeddings. Finally, when discussing music *preferences* we refer to the general preference models that given the sequence of tracks predict which tracks the user would consume, see [9].

Sequential Recommendations. There are multiple RS built around the assumption of sequential consumption. Wu et al. [1] suggest a RNN-based approach to model the dynamics of users' preferences for movie rating prediction. A similar approach is taken by Yu et al. [10] for the next basket recommendation. In contrast to our work, they study the longer-term dynamics rather than on local within-session dynamics. Some aspects of long-term recommendations have been studied with dynamic Poisson factorisation [2], Kalman filters [3] and cross-domain collaborative filtering [11]. Hidasi et al. [12] used RNN for session-based recommendations of single-item consumption. This is different to our case when the user consumes the entire sequences of tracks. A number of papers consider sequence-aware RS in ways that are not directly relevant to the present approach [13–20].

3 TEMPORAL SEQUENCING MODELS

Users typically consume tracks in *sessions* rather than individually, with *playlists* forming the backbone of in-session streaming. Such playlists typically contain selected tracks from a massive pool of songs. To generate playlists users would enjoy RS need to know how users' consumption evolves as they progress through the playlist. This motivates the study of *local*, or within-a-session, preferences. We examine two hypotheses regarding such preferences: (1) *influence of track position*: users' consumption signals at different positions of their sessions may carry slightly different information; (2) *influence of previous track*: preferences of the user may be influenced by their recent consumption within the session.

Interplay between User behavior and Track position. We investigate how track performance varies across positions. To tease apart the effect of position from other interfering effects (e.g. relevance), we first carry out an online experiment wherein users are presented with the playlists in which tracks are ordered uniformly at random (rather than by relevance). Users then interact with their playlists by consuming some tracks. For each position we quantify the global *success rate* as the average track completion rate at that position. Figure 1 (left) illustrates that the position of a track heavily influences its success rate, which monotonically decreases.

To disentangle the effect arising from track-level heterogeneity, we perform additional analysis by clustering tracks based on their success rates, and analyse the resulting trends in Figure 1 (right). Here, each cluster corresponds to a group of tracks similar in their success rates across the different positions. As we can see some of these curves have an intersection point at certain positions (e.g., curves of clusters 0 and 1 intersect at positions 15, 50, etc). This suggests that some tracks have a higher success rate than others if they are exposed to users at the earlier positions. This finding hints at a strong interplay between track position and user behaviour.

Influence of the previous track. Beyond track positions, we argue that user behaviour is influenced by relationships between the subsequent tracks. We posit that certain aspects of transitions between tracks is linked to how users perceive the subsequent tracks.

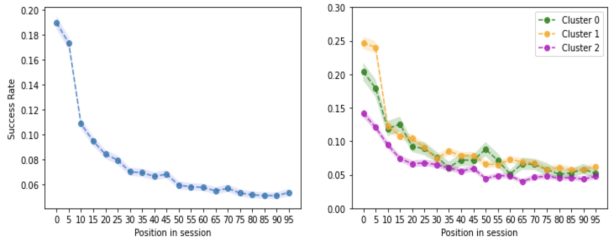

Figure 1: **Left: Track success rate across positions within sessions (error bars indicate the standard errors). Right: Performance across track clusters. We rely on k-means clustering and use BIC criterion to choose the number of clusters.**

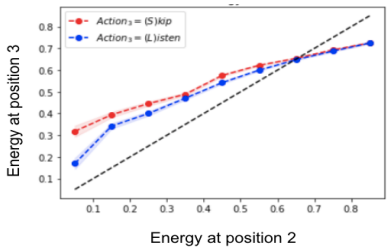

Figure 2: **Mean energy of two adjacent tracks. Track on position 2 is completed by the user. If energy of the tracks is similar, it is more likely that the track on position 3 is completed (blue); otherwise, it is likely that track 3 is skipped (red). Shaded areas indicate the standard errors.**

To illustrate this point, we extract audio characteristics/features of the tracks [21] (e.g., acoustic energy, instrumentalness). We consider acoustic energy of tracks, and compare mean acoustic energies between the adjacent tracks in the users' sessions. Figure 2 illustrates the mean acoustic energy of tracks on the 2nd and 3rd positions across 250K sessions. We see that if the energy of these tracks is more similar (i.e., closer to the dashed diagonal line), then it is more likely that both tracks are completed (blue line). Otherwise, it is more likely that the latter track is skipped (red line). This underpins the point that consumption of a certain track may depend on features of the previously interacted tracks.

4 METHOD

We now describe how the aforementioned assumptions of users' preferences can be effectively modelled by simple parametric models optimised against session-level interaction data. We start with introducing the formal model that helps us to state the general optimal sequencing problem. We then illustrate how the respective sequencer can employ our preference models to construct playlists to maximise users' expected within-a-session consumption.

4.1 Optimal Sequencing Problem

We start this section with some notations. Let u be a user and t_j be a track, for $j = 1, ..., n$. We let $\pi : \{1, ..., n\}^n \rightarrow \{1, ..., n\}^n$ be the permutation of the indices of the tracks computed for the user u; here, π defines the allocation of the n tracks to the n positions in the user's playlist. In particular, we let $t_{\pi(\ell)}$ be the track that takes position $\ell \in \{1, ..., n\}$ in u's playlist.

Let $r(u, t_j, \pi) \in \{0, 1\}$ be the random variable that corresponds to the reward generated when user u interacts with track t_j. We can let, for example, the reward take the unit value if the user completes the track (and zero otherwise). Importantly, we assume that the reward depends not only on the user u and the interacted track t_j but also on the whole permutation π. This allows us to model the sequential preferences of the user. We let $\tilde{r}(u, t_j, \pi)$ be the *realised* value of this random variable that is observed by RS after the actual interaction has happened. We also assume access to logged data, which we use to construct the training dataset $\mathcal{D} = \{(x_i, y_i)\}$ where x_i corresponds to the user and track features and y_i is the label.

Let $f(.|\mathcal{D}) \in \mathcal{F}$ be a parametric *preference model* from some class of models $\mathcal{F}$ optimised with respect to the chosen loss function on data $\mathcal{D}$. The model maps features x_i into the predicted scores $f(x_i|\mathcal{D})$ that correspond to the probabilities that a certain user completes a certain track, i.e., the probability of observing a positive reward $\tilde{r}(.)$. For example, $\mathcal{F}$ could correspond to the class of feed-forward DNNs with the cross-entropy loss and L_2 regularisation. We assume that the scores are calibrated to represent the true probabilities of the tracks to be completed.

Finally, we define the *sequencer* as the mapping that for each user u and set of tracks $\{t_1, ..., t_n\}$ outputs the permutation of the indices of these tracks π^* that constitutes the order of the tracks in u's playlist. To compute this *optimal* permutation the sequencer relies on the scores produced by model $f(.|\mathcal{D})$. The goal of the sequencer is to arrange the tracks in the playlist in a way that maximises the total expected discounted reward. Formally, the sequencer computes the optimal permutation π^* for user u as follows:

$$\pi^* \in \arg\max_{\pi} \mathbb{E}\Big[\sum_{\ell=1}^{n} \gamma^{\ell-1} r\big(u, t_{\pi(\ell)}, \pi\big)\Big], \tag{1}$$

where the expectation is taken w.r.t. the uncertain rewards. Here, $\gamma \in (0, 1)$ is a constant discount factor that is motivated by our first hypothesis (see Section 3). As shown in Figure 1, users may quit their session after each track and, consequently, they may interact more with the first tracks in the playlist than with the last ones. To reflect the risk of a user quitting their listening session, we introduce the discount factor to encourage the high reward tracks to be placed at the beginning of the playlist. Practically, the discount factor can be identified, for example, with maximum likelihood estimation of the geometric trend illustrated in Figure 1.

In the following sub-sections, we illustrate how each of the hypotheses discussed in Section 3 can be incorporated into the respective preference model.

4.2 Myopic (non-Sequential) Model

We first introduce a baseline model that does not consider any of our hypotheses on the user's sequential preferences. Instead, it assumes that the user consumes the tracks independently of each other and of their respective positions in session.

4.2.1 Model. In this scenario, we assume that whether user u completes track t_j depends only on the user/track features but not on any temporal information and the positions. To build the model that estimates the completion probability of each track we construct a dataset $\mathcal{D}^{(0)} = \{(x_i, y_i)\}$ where the feature vector $x_i = \big(u_i, t_j\big)$ corresponds to the user and track features, and the label $y_i = \tilde{r}(u_i, t_j, \pi)$ corresponds to the observed reward of the respective interaction. We then optimize the parametric function $f(.|\mathcal{D}^{(0)})$ with respect to the cross-entropy loss on the dataset $\mathcal{D}^{(0)}$.

4.2.2 Sequencer. Since the tracks and rewards are independent, then Equation (1) can be equivalently rewritten as

$$\pi^* \in \arg\max_{\pi} \sum_{\ell=1}^{n} \gamma^{\ell-1} f(u, t_{\pi(\ell)} | \mathcal{D}^{(0)}), \tag{2}$$

where $f(u, t_{\pi(\ell)}|\mathcal{D}^{(0)})$ indicates the probability of each track to be completed by the user. The optimal permutation π^* that solves Equation (2) can be found by ranking tracks according to the decreasing scores $f(u, t_j|\mathcal{D}^{(0)})$. The computational complexity of such a sequencer is $O(n \log(n))$, where n is the number of tracks in the set. Thus, *the myopic preferences can be efficiently (i.e., in polynomial time) incorporated into the optimisation-based sequencer.*

4.3 Position-Aware Model

To incorporate the positional aspect into our model, we assume that once the user reaches a certain position ℓ in the listening session, the completion of the track $t_{\pi(\ell)}$ depends not only on the user and track features but also on the position ℓ itself. Practically, we can augment the user/track features with the position ℓ in session where the respective interaction has happened. The resulting model trained on such data outputs the probability of user u to complete track $t_{\pi(\ell)}$ if the track takes the specified position ℓ in the session.

4.3.1 Model. To design such a position-aware model, we first construct a dataset $\mathcal{D}^{(p)} = \{(x_i, y_i)\}$ by augmenting $\mathcal{D}^{(0)}$ with the positions ℓ of each user/track interaction. The new feature vector is defined as $x_i = (u_i, t_j, \ell)$ while the label $y_i = \tilde{r}(u_i, t_j, \pi)$ still corresponds to the reward of the respective interaction. We then optimise the parametric model $f(.|\mathcal{D}^{(p)})$ against $\mathcal{D}^{(p)}$. Contrary to Myopic model, the predicted scores $f(u, t_j, \ell|\mathcal{D}^{(p)})$ for the same user and track interaction may be different for different positions ℓ.

4.3.2 Sequencer. To predict probabilities of each track to be completed, the model $f(.|\mathcal{D}^{(p)})$ defined above relies on the positions ℓ of tracks in sessions. However, these positions are not known at the sequencing time. Indeed, the goal of the sequencer is precisely to compute these positions based on the scores output by $f(.|\mathcal{D}^{(p)})$. Thus, the sequencing Problem (1) can be equivalently re-formulated as an Integer Programming (IP) problem as follows:

$$\max_{a_{j\ell} \in \{0,1\}} \sum_{j=1}^{n} \sum_{\ell=1}^{n} a_{j\ell} \gamma^{\ell-1} f(u, t_j, \ell|\mathcal{D}^{(p)}) \tag{3}$$

$$\text{s.t.} \sum_{j=1}^{n} a_{j\ell} \leq 1 \quad \forall \ell = 1, ..., n \tag{4}$$

$$\sum_{\ell=1}^{n} a_{j\ell} \leq 1 \quad \forall j = 1, ..., n \tag{5}$$

Here, $a_{j\ell}$ are the binary optimisation variables indicating whether track t_j should be allocated to the position ℓ in the playlist. The Objective (3) represents the total expected allocated discounted reward for user u defined in Equation (1). The Constraint (4) says that no more than one track can be allocated to each position of the playlist. The Constraint (5) says that each track can be allocated at most once. While generally IP problems are NP-complete, the following Proposition 4.1 shows that the exact solution of this particular IP (3)-(5) can be obtained in polynomial time.

Proposition 4.1. *(3)-(5) can be efficiently solved in* $O(n^{4+\frac{1}{9}})$.

Proof. We provide a sketch of the proof due to the space constraints. First, observe that the matrix of constraints (4)-(5) is totally unimodular as it coincides with the incidence matrix of the bipartite graph $G = (V \cup U, E)$ with vertex set V corresponding to the tracks, U corresponding to the positions and E being all edges between V and U. Consequently, the optimal solution can found with linear programming relaxation which is polynomial (see [22]). Q.E.D. □

Thus, *position-aware preferences can be efficiently (in polynomial time) incorporated into the sequencer.* However, while being tractable, the problem has a higher complexity than myopic ranking.

4.4 Modeling Local Sequential Preferences

Now, let us assume that completion of the track $t_{\pi(\ell)}$ at the position ℓ in u's session depends not only on the track itself but also on the tracks allocated at the previous positions $\ell - 1$, $\ell - 2$ etc.

4.4.1 Model. The assumption above can be incorporated into the model by introducing lagged track features. We construct a dataset $\mathcal{D}^{(\tau)} = \{(x_i, y_i)\}$ with $\tau \in \mathbb{N}$ lagged track features where $x_i = (u_i, t_{\pi(\ell)}, ..., t_{\pi(\ell-\tau)})$. We can then construct a model $f(.|\mathcal{D}^{(\tau)})$ optimizing it against such a dataset.

4.4.2 Sequencer. For simplicity, we illustrate the design of the sequencer in case when $\tau = 1$. For larger values of τ the design is analogous. First, observe that Problem (1) can be equivalently formulated as the following Integer Program:

$$\max_{a_{jk} \in \{0,1\}} \sum_{j=1}^{n} f(u, t_j|\mathcal{D}^{(0)}) a_{j1} + \sum_{\ell=2}^{n} \sum_{\substack{j,s=1 \\ j \neq s}}^{n} \gamma^{\ell-1} f(u, t_s, t_j|\mathcal{D}^{(1)}) \underbrace{a_{s\ell} a_{j,\ell-1}}_{\text{non-linearity}} \tag{6}$$

$$\text{s.t.} \sum_{j=1}^{n} a_{j\ell} \leq 1 \quad \forall \ell = 1, ..., n \tag{7}$$

$$\sum_{\ell=1}^{n} a_{j\ell} \leq 1 \quad \forall j = 1, ..., n \tag{8}$$

Here, the first term in the objective function corresponds to the first track in the playlist. As for this track there does not exist previous tracks, we rely on the non-sequential model $f(.|\mathcal{D}^{(0)})$ to compute the probability of this track to be completed. For positions $\ell \geq 2$ we compute the respective probabilities relying on model $f(.|\mathcal{D}^{(1)})$. Constraints (7-8) are feasibility constraints.

In contrast to Problem (3)-(5), Problem (6)-(8) is NP-hard.[1] This makes *incorporation of the local sequential preferences into the sequencer inherently more complex than the problem of incorporating the position-aware preferences.* Practically, we can still rely on heuristics to find a solution to (6)-(8) that is "good enough". For example, at any position ℓ we can greedily pick the next track that optimises the immediate reward given the previously allocated track.

[1]The formal statement about NP-hardness follows from the reduction of Problem (6)-(8) to the problem of finding the maximal Hamiltonian path in a complete directed graph $G = (V \cup \{w\}, E)$. Here, vertices $v_i \in V$ correspond to different tracks, w is the starting vertex, an edge $(v_i, v_j) \in E$ is weighted with $f(u, v_i, v_j|\mathcal{D}_{\ell}^{(t)})$ for all $i, j = 1, ..., n$, $i \neq j$ and an edge $(w, v_j) \in E$ is weighted with $f(u, v_j|\mathcal{D}^{(0)})$ for all $j = 1, ..., n$. We omit the full proof due to the space constraints.

5 EXPERIMENTS

We start this section with presenting our experimental setup. We then perform an offline counterfactual analysis to assess the quality of the sequencers. Finally, we provide a qualitative analysis followed by a large-scale online experiment with the best performing model.

5.1 Setup

Dataset 1. We train and evaluate our models on a sample of 20M user/track interactions, extracted from 250K sessions across 2K personalised playlists. We restrict ourselves to sessions that contain between 5 and 20 tracks. We rely on three categories of features: user features (user latent 80-dimensional vectors, demographics, etc.), track features (track latent 80-dimensional vectors, acoustic features, genres), and joint user-track features (time of the day, positions of the tracks in the sessions). The 80-dimensional latent track representations are computed relying on the upstream *word2vec* model [23] optimised against the users-generated playlists [24]. Shortly, the resulting vectors of any two tracks are close to each other if the respective tracks are likely to co-occur in the user-generated playlists. The respective users' vectors are computed as the average of all tracks vectors the users have interacted with within the last 3 months. The track acoustic features include 12 different features such as acoustic tempo, energy, etc. Our dataset contains the binary rewards $\tilde{r}(.)$ indicating completions of tracks.

Dataset 2. To perform offline off-policy evaluation we conduct an auxiliary randomised A/B experiment wherein we deployed a *random sequencer* that presents users with sequences generated by a uniform random ranking *logging policy*. For every position in the playlist this policy chooses a random track out of a pool of N tracks with probability $1/N$. We then let 200K users interact with such playlists over the course of one day. This produced a dataset with approximately 22M of interaction records. We augmented this dataset with the respective re-ranked sequences produced by *target policies* that corresponds to the sequencers described in Sections 4.2-4.4. We further elaborate on our off-policy evaluation setup that leverages this dataset in Section 5.2.

Baselines. We compare all our models to these three baselines:

- *Relevance Sequencer* [25]. This sequencer ranks tracks according to the decreasing cosine similarity between the user and the track vectors. This is a weak baseline that uses minimal information to produce rankings.
- *Myopic Sequencer* [24]. The sequencer defined in Section 4.2, relies on the non-sequential model $f(.|\mathcal{D}^{(0)})$ to rank tracks w.r.t. the decreasing completion scores. We use this sequencer as our primary baseline as it relies on the same model class $\mathcal{F}$ and exploits same data as is available to our local sequential models. This allows us to attribute the gains in performance to sequential modelling rather that to a more complex model.
- *Local-Sequential with Short Memory*. To avoid an unfair comparison of sequential to non-sequential models we use the most basic of our sequential models $f(.|\mathcal{D}^{(1)})$ as a baseline for the rest of all our models. This baseline relies only on one-step lagged track features, and thus has a short *memory*. In what follows, we refer to this model as Local-Sequential (Short Mem.).

As the goal of our work is not to model all the subtleties of the sequential users' preferences but rather to quantify the impact of the two specific sequential hypotheses on the users' within-a-session consumption, *we do not compare our models to the complex neural architectures such as Transformers [26] or CVA*E. Instead, we rely on a class of simple yet universal models to capture preferences.

Class of Models. We restrict the class of models $\mathcal{F}$ to feed-forward DNNs trained with the cross-entropy loss and L_2 and dropout regularisation as these models are universal approximators [27] while being relatively simple. We further embed categorical features (such as the country or genre) into a lower ten-dimensional embedding space. Our experiments relied on the DNN with four hidden layers and 256, 128, 64 and 32 units in each of the layers, respectively. We further used Linear Programming (LP) solver [28] to solve (3)-(5) and implemented a greedy strategy (Section 4.4) to solve (6)-(8).

5.2 Quantitative Analysis

To understand how well the model improvements discussed above translate into higher consumption we perform an offline evaluation of the proposed sequencers. As standard information retrieval (IR) metrics such as NDCG@k do not take into account the local past history of user/track interactions we instead rely on Inverse Propensity Scoring (IPS) technique [29] and the randomised Dataset 2 (see Section 5.1) to perform a counterfactual evaluation.

As our dataset was produced with the uniform random logging policy the computation of propensities required by IPS become straightforward. We utilise Reward Interaction IPS (RIPS) [30] that provides an asymptotically unbiased estimator in the domain with sequential reward interactions. RIPS allows to account for the fact that the track and the respective reward on position ℓ can impact the reward of the following track on position $\ell + 1$. Using RIPS we estimate the expected number of completed tracks across the first K positions, K=3,5,10. We further compare these results with the ones obtained with the standard IPS estimator (IIPS) [31].

Table 1 contains our results. As expected, Relevance Sequencer performs worse than Myopic (as this is a weak baseline). Looking at the performance of Position-Aware model, we see that it w.r.t. IIPS@10 and RIPS@3. Notice that low values of RIPS@5 and RIPS@10 are not negative results as the respective variance is large (hence, RIPS@5 and RIPS@10 measurements may be less reliable). Overall, this suggests that the Position-Aware sequencer can indeed improve the number of completed tracks within a session.

We now look into Local-Sequential sequencers. Interestingly, RIPS@3 suggests that the performance of Local-Sequential (Short Mem.) is significantly lower compared to Myopic. This result is not surprising. As we see in Section 4.4 local-sequential preferences are much harder to *optimally* incorporate into the sequencer. Contrary, Local-Sequential (Medium Mem.) and Local-Sequential (Long Mem.) demonstrate much more promising results w.r.t. RIPS@3 (the rest of the IPS estimates are slightly smaller but the difference is not significant given the variance). This can be explained by the fact that the respective preference models have more memory which helps the sequencer to avoid a number of local optima. This suggests that *the high complexity of ranking with sequential models can be partially mitigated by increasing the memory of the preference model.*

	IIPS@3	IIPS@5	IIPS@10	RIPS@3	RIPS@5	RIPS@10	AUC
Relevance Sequencer	0.5041 (0.0001)	0.8502 (0.0002)	1.6408 (0.0004)	0.4550 (0.0012)	0.7613(0.0035)	1.4778 (0.0193)	-
Myopic Sequencer	0.5558 (0.0002)	0.9218 (0.0003)	1.7372 (0.0005)	0.5063 (0.0049)	0.9723 (0.0192)	1.7600 (0.0379)	0.722
Position-Aware	0.5597 (0.0002)	0.9236 (0.0003)	**1.7723 (0.0006)**	**0.5810 (0.0072)**	0.8757 (0.0139)	1.6826 (0.0328)	0.725
Local-Sequential (Short Mem.)	0.5522 (0.0001)	0.8977 (0.0002)	1.7102 (0.0005)	0.4277 (0.0028)	0.7404 (0.0070)	1.5322 (0.0309)	0.728
Local-Sequential (Medium Mem.)	0.5455 (0.0001)	0.9016 (0.0002)	1.7055 (0.0005)	**0.5731 (0.0070)**	0.8861 (0.0136)	1.7554 (0.0325)	0.729
Local-Sequential (Long Mem.)	0.5468 (0.0001)	0.8849 (0.0003)	1.7128 (0.0003)	**0.5777 (0.0084)**	0.9819 (0.0168)	1.9370 (0.0429)	0.730

Table 1: Off-policy evaluation. Expected rewards (track completions) and variances estimated with Independent IPS and Reward Interaction IPS for K=3,5,10.

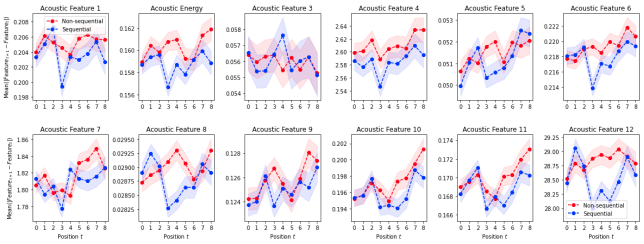

Figure 3: Mean distances between the acoustic features of consecutive tracks on different positions of the ranked playlists. The red and blue trends correspond to Myopic and Local-Sequential (Short Mem.) Sequencers respectively. Shaded areas indicate the standard errors.

5.3 Impact across content & session types

To illustrate the impact of our model on the acoustic coherence of the generated playlists we perform additional qualitative analysis. Specifically, Figure 2 suggests that if two tracks on consecutive positions ℓ and $\ell + 1$ are similar w.r.t. the acoustic energy, it is more likely that both tracks are completed. Figure 3 illustrates the difference between 12 acoustic features of two consecutive tracks at different positions of the sequences ranked with our models. The red trend corresponds to the rankings produced by Myopic model while the blue one corresponds to Local-Sequential (Short Mem.).[2] We see that whenever there is significant difference between the two trends, the consecutive tracks ranked with Local-Sequential (Short Mem.) are more similar to each other than the tracks ranked with Myopic (i.e., the blue curve is typically below the red one). Thus, our models could learn and exploit the users' acoustic preferences.

5.4 Online A/B Experiment

To validate our hypothesis that exploiting local-sequential preferences increases within-session consumption we carried out an online A/B test with the *treatment* group exposed to our best performing local-sequential model. We optimised this model against session level dataset to predict whether the user completes tracks taking into account the previous tracks (see Section 4.4). We compared our treatment against the *control* Myopic sequencer on a sample of 7M users across more than 80 markets who were registered with the service for at least one month. We segmented users into Premium users (6M) who can enjoy unlimited interactions with the playlist and Free users (1M) who can only skip a limited number of tracks. Table 2 illustrates our findings.

First, we see that relying on the sequential model we can indeed increase the number of completed tracks per session by 2.52% relative to Myopic. This confirms our main hypothesis that capturing the users' sequential preferences helps to increase music consumption within a session. As the completion rate increases, the skip rate decreases (the 2nd row in Table 2). This holds for both Premium and Free users. In fact, Free users benefit from our model more than Premium (+3.7% in completion rate and -4.27% in skip rate against +2.5% and -2.11% for Premium respectively). The decrease in the number of skips for Free users is important as it significantly improves their experience given their limited number of skips.

[2]Results for Local-Sequential (Med. Mem.), Local-Sequential (Long Mem.) are similar.

	Relative diff (%) to Control		
	All	Premium	Free
Completions	2.52*** (2.50, 2.54)	2.50*** (2.47, 2.52)	3.7*** (3.66, 3.74)
Skips	-2.73*** (-2.75, -2.71)	-2.11*** (-2.13,-2.09)	-4.27*** (-4.29,-4.26)
Observations	7M	6M	1M
Note	*** p<.001		

Table 2: Online Experiment: relative gains and confidence intervals for the average number of completions/skips within a session relative to the myopic non-sequential sequencer. 95% confidence intervals are in the parentheses.

5.5 Discussion

Our experiments demonstrate that models that make use of users' sequential preferences lead to better predictions of users' consumption. These models can be incorporated into practical sequencing algorithms to improve within-a-session consumption. By performing an offline study we argued that even the complex sequential preferences can be effectively exploited by the sequencer by transferring the complexity from the sequencer to the preference model itself. Furthermore, while both position-aware and local-sequential hypotheses help to improve the expected number of completed tracks, the gains of the local-sequential model overweight those of the position-aware one. Finally, by means of an A/B test we illustrated that leveraging the users' sequential preferences helps to significantly improve within-session consumption. The effect is even more pronounced for the Free users.

6 CONCLUSION

We studied the problem of modelling users' local *within-a-session* preferences and incorporating these preferences into sequencing algorithms in an optimisation-based way. Specifically, we identified and focused on two main hypotheses: the role of the track position and the role of the previous tracks on the users' consumption. We illustrated that incorporating such preferences into the sequencer is non-trivial: The more expressive the preference model is, the harder it is to use it efficiently. Therefore, optimisation-driven techniques are required. To this end, we illustrated the design of an optimisation-based sequencer that is able to leverage such preferences to improve the users' consumption within-a-session. Importantly, despite of the higher complexity of exploiting sequential preferences we could incorporate our models into the industrial low-latency production system to serve millions of users at scale within our A/B test. This further illustrates practicality of our approach. All the results above suggest that leveraging sequential preferences for music sequencing is a promising research direction.

REFERENCES

[1] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, (New York, NY, USA), p. 495–503, Association for Computing Machinery, 2017.

[2] L. Charlin, R. Ranganath, J. McInerney, and D. M. Blei, "Dynamic poisson factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, (New York, NY, USA), p. 155–162, Association for Computing Machinery, 2015.

[3] S. Gultekin and J. Paisley, "A collaborative kalman filter for time-evolving dyadic processes," in *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, (USA), p. 140–149, IEEE Computer Society, 2014.

[4] R. Mehrotra, B. Carterette, Y. Li, Q. Yao, C. Gao, J. Kwok, Q. Yang, and I. Guyon, "Advances in recommender systems: From multi-stakeholder marketplaces to automated recsys," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, (New York, NY, USA), p. 3533–3534, Association for Computing Machinery, 2020.

[5] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challenges," *Computer Science Review*, vol. 6, no. 2, pp. 89–119, 2012.

[6] 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 *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pp. 53–62, ACM, 2020.

[7] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2392–2396, 2017.

[8] K. Chen, B. Liang, X. Ma, and M. Gu, "Learning audio embeddings with user listening data for content-based music recommendation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10 2020.

[9] J. D. Levin and P. R. Milgrom, "Introduction to choice theory," 2004.

[10] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, (New York, NY, USA), p. 729–732, Association for Computing Machinery, 2016.

[11] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu, "Cross-domain collaborative filtering over time," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three*, IJCAI'11, p. 2293–2298, AAAI Press, 2011.

[12] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *International Conference on Learning Representations (ICLR)*, (San Juan, Puerto Rico), 05 2016.

[13] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Comput. Surv.*, vol. 51, July 2018.

[14] Y. Cao, W. Zhang, B. Song, W. Pan, and C. Xu, "Position-aware context attention for session-based recommendation," *Neurocomputing*, vol. 376, pp. 65–72, 2020.

[15] L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack, "Sse-pt: Sequential recommendation via personalized transformer," in *Fourteenth ACM Conference on Recommender Systems*, RecSys '20, (New York, NY, USA), p. 328–337, Association for Computing Machinery, 2020.

[16] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, (New York, NY, USA), p. 1441–1450, Association for Computing Machinery, 2019.

[17] D. Jannach, B. Mobasher, and S. Berkovsky, "Research directions in session-based and sequential recommendation: A preface to the special issue," *User Modeling and User-Adapted Interaction*, vol. 30, 08 2020.

[18] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, Jul 2019.

[19] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, (New York, NY, USA), p. 729–732, Association for Computing Machinery, 2016.

[20] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, and E. H. Chi, "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, (New York, NY, USA), p. 46–54, Association for Computing Machinery, 2018.

[21] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[22] S. Jiang, Z. Song, O. Weinstein, and H. Zhang, "Faster dynamic matrix inverse for faster lps," *ArXiv*, vol. abs/2004.07470, 2020.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[24] C. Hansen, R. Mehrotra, C. Hansen, B. Brost, L. Maystre, and M. Lalmas, "Shifting consumption towards diverse content on music streaming platforms," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, (New York, NY, USA), p. 238–246, Association for Computing Machinery, 2021.

[25] A. Anderson, L. Maystre, I. Anderson, R. Mehrotra, and M. Lalmas, "Algorithmic effects on the diversity of consumption on spotify," in *Proceedings of The Web Conference 2020*, WWW '20, (New York, NY, USA), p. 2155–2165, Association for Computing Machinery, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[27] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[28] "Google or-tools." https://developers.google.com/optimization. Accessed: 2021-02-07.

[29] M. Dudík, D. Erhan, J. Langford, and L. Li, "Doubly robust policy evaluation and optimization," *Statistical Science*, vol. 29, no. 4, pp. 485–511, 2014.

[30] J. McInerney, B. Brost, P. Chandar, R. Mehrotra, and B. Carterette, "Counterfactual evaluation of slate recommendations with sequential reward interactions," pp. 1779–1788, 08 2020.

[31] S. Li, Y. Abbasi-Yadkori, B. Kveton, S. Muthukrishnan, V. Vinay, and Z. Wen, "Offline evaluation of ranking policies with click models," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, (New York, NY, USA), p. 1685–1694, Association for Computing Machinery, 2018.