Shifting Consumption towards Diverse Content on Music Streaming Platforms

Christian Hansen*
University of Copenhagen
chrh@di.ku.dk

Rishabh Mehrotra
Spotify
rishabhm@spotify.com

Casper Hansen*
University of Copenhagen
c.hansen@di.ku.dk

Brian Brost
Spotify
brianbrost@spotify.com

Lucas Maystre
Spotify
lucasm@spotify.com

Mounia Lalmas
Spotify
mounia@acm.org

ABSTRACT

Algorithmic recommendations shape music consumption at scale, and understanding the impact of various algorithmic models on how content is consumed is a central question for music streaming platforms. The ability to shift consumption towards less popular content and towards content different from user's typical historic tastes not only affords the platform ways of handling issues such as filter bubbles and popularity bias, but also contributes to maintaining a healthy and sustainable consumption patterns necessary for overall platform success.

In this work, we view diversity as an enabler for shifting consumption and consider two notions of music diversity, based on taste similarity and popularity, and investigate how four different recommendation approaches optimized for user satisfaction, fare on diversity metrics. To investigate how the ranker complexity influences diversity, we use two well-known rankers and propose two new models of increased complexity: a feedback aware neural ranker and a reinforcement learning (RL) based ranker. We demonstrate that our models lead to gains in satisfaction, but at the cost of diversity. Such trade-off between model complexity and diversity necessitates the need for explicitly encoding diversity in the modeling process, for which we consider four types of approaches: interleaving based, submodularity based, interpolation, and RL reward modeling based. We find that our reward modeling based RL approach achieves the best trade-off between optimizing the satisfaction metric and surfacing diverse content, thereby enabling consumption shifting at scale. Our findings have implications for the design and deployment of practical approaches for music diversification, which we discuss at length.

KEYWORDS

Recommender systems; Music; Shifting consumption; Diversity

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

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441775

ACM Reference Format:
Christian Hansen, Rishabh Mehrotra, Casper Hansen, Brian Brost, Lucas Maystre, and Mounia Lalmas. 2021. Shifting Consumption towards Diverse Content on Music Streaming Platforms. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441775

1 INTRODUCTION

Algorithmically generated recommendations power and shape the bulk of music consumption on music streaming platforms. Given the large role of streaming in the music industry, it has become important for music streaming platforms to consider the influence of their recommendations on music consumption in a manner benefiting not only the users, but also artists, and the long term goals of the platform itself. The ability to influence and shift consumption at scale enables system designers to maintain healthy consumption patterns needed for long term platform health and success.

A fundamental characteristic of a music recommendation system that helps platforms shape consumption is its *diversity*. What does diversity means in the context of music recommendation? First, it can facilitate exploration by helping users discover new content or inculcate new tastes [11, 31, 34]. Additionally, it can help the platform spread consumption across artists and facilitate consumption of less popular content. This, in turn, can help counteract rich-get-richer phenomena common throughout the music industry [24]. Finally, it has recently been shown that consumption of diverse music genres is strongly associated with important long-term business metrics, such as user conversion and retention [1].

We formalize our notion of diversity around two central factors that influence music consumption via recommender systems: 1) taste similarity, or how similar a piece of music is to the type of music the user has historically streamed, and 2) popularity, or how many users have recently streamed the piece of content [12]. Based on this, two notions of diversity naturally emerge, one based on the user bias of consumed content, and another based on the global bias of consumed content. From the former point of view, one can achieve diversity and shift consumption by avoiding recommending similar songs to what the user has historically streamed, while in the latter view of diversification, one can shift consumption towards the long tail of consumed music. Focusing on these two notions of diversity enables us to effectively and efficiently drive diversity and influence music consumption, both

at the user- and global- level.

Present Work. We focus on the case of sequential recommendations, and consider four different types of sequential recommenders, or *rankers*, of increasing complexity. We leverage two widely used types of rankers: similarity based and feed-forward neural rankers, and propose two additional rankers, a feedback aware neural attention ranker, and a reinforcement learning (RL) based ranker. This provides us with a wide spectrum of approaches, from simple similarity based rankers to sophisticated reward based RL ranker, and enables us to understand the interplay between model complexity and performance. In support of recent findings that highlight the drop in diversity metrics for models optimized for user satisfaction [15], we investigate how these rankers perform in terms of diversity. Further, we consider four different ways of incorporating diversity in recommendation models: (i) linear interpolation, (ii) submodular diversification, (iii) interleaving based and (iv) reward modeling based on reinforcement learning (RL) ranker.

Overall, our work considers three key questions around (i) the role of two classes of diversity for shifting consumption, and their interplay with user satisfaction, (ii) how four rankers of increasing complexity fare in terms of satisfaction and diversity objectives; and (iii) how four different techniques of incorporating diversity manage the trade-off against user satisfaction.

Overview of results. Looking at music consumption data, we find evidence that users can often be satisfied with recommendations that depart from their historic taste profiles and that are less popular. This underpins the scope for shifting consumption towards diverse content without dissatisfying users. Comparing different rankers, we find strong evidence suggesting that satisfaction centric rankers are heavily biased towards popular content, and content closely resembling user's historic listening activity, which should not be surprising. Interestingly, this bias increases with model complexity, with advanced rankers suffering from this bias to a greater extent.

Among the different diversification techniques, we see that the reward modeling approach for RL model obtains the best trade-off by obtaining a high satisfaction metric and succeeding in surfacing less popular content. For diversity with respect to a user's listening history, we observe that the RL approach performs comparably to the interpolation strategy, with the interpolation strategy offering a wider range of trade-off and subsequently more control over consumption. More interestingly, comparing these results with the ranker comparison on only satisfaction, we observe bigger differences in satisfaction metrics when rankers consider diversity, than when they are only focused on satisfaction.

Taken together, our work sheds light on a central tension between optimizing recommendation models for satisfaction centric objectives versus diversity goals. Developing better rankers results in increasing short term user satisfaction, albeit slightly. However, such models tend to serve less diverse recommendations.

2 RELATED WORK

Retrieving diverse documents has long been recognized as an important challenge in information retrieval [5–7] and for recommender systems [10]. The central problem is that in many applications, it is not sufficient simply to return relevant items, instead the system must account for multiple user intents and needs, in addition to possible redundancy in the content of the returned items. The term *diversity* was first used within information retrieval in [6]. Here a list was considered diverse if it contained items with low similarity to each other. The ranked list was built greedily, with the score of each item being an interpolation of the expected relevance to the user, and the dissimilarity of the item to all previously recommended items in the list. The problem of diversity in list recommendation has in later years received great interest in developing more advanced methods to ensure list diversity [2, 3, 22, 29]. A detailed survey of a variety of methods can be found in [10].

Closely related to diversity is the notion of fairness in recommendations, e.g. [4, 25]. Here we consider diversity from the point of view of the recommended items, e.g. in group fairness, where if the items can be considered to be part of a group, all groups must on average be represented in the final recommendation. This can be extended to marketplace settings, where multiple different stakeholders have requirements for the fairness of the recommendation [15]. Thus, whereas diversity is often considered to be a user centric concern, fairness is item centric, as a fair ranking needs to give equal opportunity for the recommended items.

Whereas existing work on diversity and fairness tends to focus on the ranked list setting, we consider the problem of sequential recommendations of single items. This is a substantially different problem setting, since the user is forced to consume each recommended item, and items introduced to satisfy diversity objectives cannot be as easily ignored by the user if they turn out irrelevant, as in the case of a ranked list.

The interplay between recommender systems and diversity has been popularized in [20], raising public awareness on the so-called "filter bubble" phenomenon. There has been a number of works looking at the effects of recommender systems on the diversity of consumption. A study of a movie recommender system used on a popular e-commerce web site found that the recommendations led to a decrease in sales diversity [8]. By contrast, a study on the effect of recommendations on the YouTube video platform was shown to lead to more diverse consumption [33]. Finally, in the context of music, a strong relationship between consumption diversity and long term platform metrics such as retention and conversion was shown [1]. These finding support the need for properly addressing diversity as part of the recommender system design.

Acting on diversity entails a *formal definition* of diversity. For example, in [1, 28] a setting where items are embedded in a Euclidean space is considered, and diversity is then defined as a function of pairwise distances in that space. Other definitions have been used in [7, 15, 23]. In this work, we consider two operationalizations of diversity, with a focus on simple and practical definitions that are easy to implement in real-world systems.

3 DIVERSITY FOR CONSUMPTION SHIFTING

Our goal is to understand how algorithmic recommendations can help shift consumption through diversity in music consumption. Given the sequential nature of music consumption wherein the user sequentially decides to stream or skip the recommended music, it is not straightforward to recommend a track solely for the purpose of increasing diversity, especially if the track has a low chance of being listened to. Given this complex interplay between relevance

Figure 1: Track level correlation.

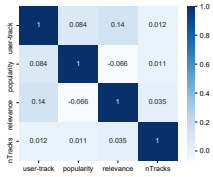

Figure 2: Session level correlation.

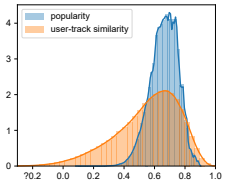

Figure 3: Diversity distributions across tracks.

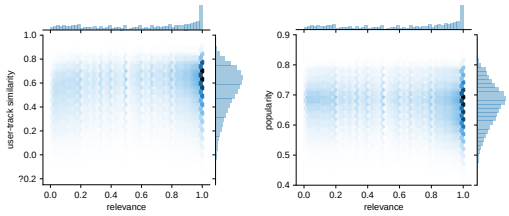

Figure 4: Density of popularity/user-track similarity in relation to relevance across session.

of music to the user, its popularity and the resulting success of diversification, it becomes important to carefully understand the relationship between such concepts. In this work we consider a track to be irrelevant if the user skipped it and otherwise relevant if the user listened to the track. We begin by looking at diversity through the lens of user-track similarity and popularity, and investigate how often are users satisfied with content that either departs from their historic listening habits, or is less popular. Understanding this enables us to underpin the scope of consumption shifting via diversification.

3.1 Quantifying diversity

While numerous ways of defining and quantifying diversity exist, in this work, we are interested in two notions of diversity: 1) diversity *on a global level*, where the diversity is defined as a property of a track t itself ($d(t)$); 2) diversity of a *track as depending on the user* u to which it is recommended ($d(t,u)$), such that the diversity would differ if the same track is recommended to two different users. $d(t)$ encompass a broad set of different notions of diversity, e.g., a high diversity score could be associated with new tracks on the platform, or for tracks of genres rarely listened to by the general user base. Similarly, $d(t,u)$ encompass different notions where the diversity is depending on the user, e.g., a high diversity score could be associated with tracks of artists rarely listened to by a user, or tracks from time periods less familiar to the user.

While the work in this paper can be used for different notions of diversity of the form $d(t)$ or $d(t,u)$, we choose to work on two specific notions of diversity of great importance for music recommendation. For $d(t)$, we consider the global popularity of the track, while for $d(t,u)$, we consider how similar the recommended track is to tracks previously encountered by the user. For ease of notation we always denote the diversity as $d(t,u)$, even though the track level diversity is independent of the user, i.e., $d(t,u') = d(t,u)$ for all users u, u'.

The similarity between a track and tracks previously encountered by the user (denoted as the user-track similarity) is computed as the cosine similarity between a user embedding and a track embedding (see Section 4.1), where the user embedding encodes information from all tracks the user has streamed in the past. The popularity of a track is determined by usage statistics on the platform.

3.2 Analysis of diversity and user satisfaction

We investigate how the notions of diversity, as defined in this work, are related to relevance, and overall engagement measured by session length (measured as the number of tracks within a session). We conduct our analysis on both track- and session- level, and consider a track to be relevant if the user did not skip it. For the track level analysis we use a dataset of 2 million randomly sampled recommended tracks, containing the popularity of the track, the user-track similarity, and the relevance. For the session level analysis we randomly sampled 1 million user sessions, where each session has at least 5 tracks to filter out short sessions. For each session, we log the number of tracks, average popularity, and average user-track similarity across the session, as well as the number of tracks relevant to the user.

Track level: The distribution of popularity and user-track similarity can be seen in Figure 3, and the correlation between the notions of diversity and relevance can be seen in Figure 1. The distribution plots show that users engage with tracks of varying popularity and user-track similarity, but with a large tendency to engage with tracks of both high popularity and user-track similarity. For both distribution plots, the density drops rapidly as popularity/user-track similarity decreases. The correlation plots between the notions of diversity and relevance show that user-track similarity is positively correlated with relevance, which indicates that reducing user-track similarity potentially can harm the user experience. In contrast, the popularity of the tracks is not correlated with relevance, and could likely be reduced without harming the user experience.

Session level: Figure 2 shows the correlation between session popularity, user-track similarity, average relevance of recommended tracks, and number of tracks in the session (session length). We observe that the average popularity is not correlated to either the session length or the average relevance. As seen in the track level analysis, user-track similarity is correlated with relevance, but interestingly it is not correlated with the session length. Figure 4 shows the distribution of both notions of diversity with regards to the average relevance of the session. The highest density is at high popularity/user-track similarity and at fully relevant sessions, but there is considerable density outside this area. Indeed, sessions exist where users are not satisfied with the most popular tracks (upper left side), and there are sessions where they are satisfied with low popularity tracks (lower right side). The same can be observed for user-track similarity.

This analysis motivates that it is possible to shift consumption towards more diverse recommendations without harming user satisfaction, and the typical focus on high popularity/user-track similarity is detrimental for some sessions.

4 RANKERS & DIVERSITY METHODS

We consider the problem of sequential recommendation in a session, where a user consumes a series of recommended music tracks. In this setting, users can either skip or listen to a track. We consider

Table 1: Description of user, track, and user-track combination features used in the neural rankers.

Feature Type	Feature	Description
User	embedding	40 dimensional learnt word2vec vector of user
	country	country of registration for user
Track	embedding	40 dimensional learnt word2vec vector of track
	popularity	normalized popularity of the track
	genres	genres relevant to the track
	acoustic	16 derived acoustic features
	track length	track duration in seconds
User-Track	similarity	cosine similarity between user and track embeddings
	distance	Euclidean distance between user and track embeddings
	genre affinity	affinity for highest overlapping genre between user & track
Playlist	playlist ID	a unique playlist identifier used for learning embeddings

a skipped track as irrelevant, and a listened track as relevant. A session starts with a user selecting a playlist, which consists of tracks with some thematic overlap (e.g., Jazz songs), and is recommended a series of tracks from the playlist, until the user chooses to end the session. We consider two different recommendation scenarios. In the first one, we aim to recommend the tracks a user is most likely to enjoy, and consider four different *rankers* of increasing complexity for this purpose. In the second one, we aim to recommend tracks the user is likely to enjoy, but with the secondary objective that the tracks should also be diverse, where the definition of diversity is detailed in Section 3. To include *diversity* in the recommendation, we explore four different methods to optimize the trade-off between making both relevant and diverse recommendations.

4.1 Preliminaries

We describe the features available to the rankers. This is important as part of the difference between the rankers is to what extent they can make use of the feature space. An overview of the features can be found in Table 1.

Each track is represented as a concatenation of three distinct feature vectors: a **contextual** vector, an **acoustic** vector, and a **statistic** vector. The **contextual** vector is a 40 dimensional real valued vector, which is trained such that two tracks that occur in the same context, will lie close to each other in the vector space [15]. The **acoustic** vector consists of 16 derived features that reflect different acoustic features of the track, e.g., loudness. Lastly, the **statistics** vector contains information on the track length and popularity of the track on the platform. Each user is represented as a weighted average of the **contextual** vectors of the tracks the user has played in the past as described in [15]. The similarity between a track and a user are computed by taking the cosine similarity between the user vector and the track **contextual** vector, as they reside in the same space.

For each user and track pair, there are a number of derived features capturing their relations. The cosine similarity and Euclidean distance between the user and track is computed and used as a feature. Additionally, each user has an affinity for all genres, which is used as a feature by taking the maximum affinity within the track's genres. Lastly, each playlist is represented with a unique identifier, which is used by some of the ranking models for learning playlist specific embeddings during model training. In the next sections, the features are grouped into either: T, which is the combination of the track and user-track features (track level features); or M, which is the combination of the playlist embedding and user features (session level meta features).

4.2 Rankers

We present four different rankers of increasing complexity. The first is based on the cosine similarity between user and track, while the remaining three are learned neural models. An overview of the latter three is provided in Figure 5.

4.2.1 Cosine ranker. This ranker uses the cosine between a track's *contextual* embedding, $e_{track} \in \mathbf{R}^{40}$, and a user's *contextual* embedding $e_{user} \in \mathbf{R}^{40}$: $score_{\text{cosine}} = \frac{e_{track} \cdot e_{user}}{||e_{track}||_2 ||e_{user}||_2}$. A high cosine score indicates that the track is similar to tracks the user has previously consumed on the platform. While being simple, this type of ranker has been used for music recommendation in previous work [1, 9, 15].

4.2.2 Feed forward ranker. This is a neural feed forward network, which takes as input the track level features (T) and session level meta features (M). All the features are concatenated, and the network gives a score for a single track:

$$score_{\text{FF}} = \sigma(W_3 \,\text{ReLU}(W_2 \text{ReLU}(W_1 \,[T \oplus M] + b_1) + b_2) + b_3) \quad (1)$$

where *FF* stands for feed forward, $\oplus$ is vector concatenation, ReLU is the rectified linear unit, and σ is the sigmoid function. The weight matrices (W) and bias vectors (b) have input-suitable sizes and are learned during training. The embedding for the playlist is learned by the network during training. The feed forward network consists of 2 hidden layers with relu activation functions, and a prediction layer using a sigmoid activation function. This prediction corresponds to the probability of a user skipping a track, which is optimized using the cross entropy loss. This model is relatively simple, and computationally efficient. We include it to show how well the score can be computed without considering the user's history directly. Note that the network is indirectly aware of the user's history through the user embedding and the user-track features.

4.2.3 Feedback aware ranker. This ranker is our proposed extension of the feed forward ranker, and incorporates the user's previous sessions to compute a dynamic user embedding. While the two previous models gave a score based on a single track, this model needs to be provided the user's history as input. We first cover how the dynamic user embedding is computed, which consists of two parts: 1) summarising a single session, and 2) summarising all sessions to a final dynamic user embedding.

Summarising a single session. Each session, s, consists of session level meta features, M, and a sequence of tracks (T,R) $\in s$, where T is the track level features and R is a indicator whether the user found the track relevant. The session is summarised using a long short-term memory (LSTM) followed by an attention softmax layer:

$$o_i, h_i = \text{LSTM}(s_i | o_{i-1}, h_{i-1}), \quad \hat{o}_i = \text{ReLU}(W_4 \, o_i + b_4) \quad (2)$$

$$S = \sum_{i=1}^{|s|} \hat{o}_i \frac{e^{W_5 \, \hat{o}_i + b_5}}{\sum_{j=0}^{|s|} e^{W_5 \, \hat{o}_j + b_5}} \quad (3)$$

where LSTM denotes an LSTM cell which update the hidden state h and output o. The LSTM cell is initialised by a linear projection of the session meta information such that the session representation can be user and playlist dependent.

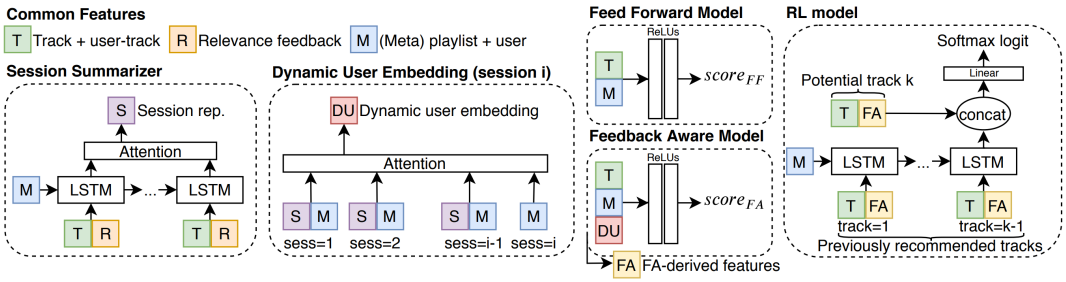

Figure 5: Overview of the neural rankers seen from the perspective of a single track.

Dynamic user embedding. At timepoint i, the users have a set of previous session embeddings, $S_j \in \mathcal{S},\ j \in [1, i-1]$, each having associated meta information M_j. The dynamic user embedding is a summary of all previous session embeddings, conditioned on the current sessions Meta information, M_i. The summarisation of previous sessions is done by attention weighting, where the weighting is based on an interaction vector between the current session meta information, and the historic sessions meta information. The interaction vector [17] is the concatenation, subtraction, and multiplication of the past session and current session meta representations, to represent the representational changes between the sessions. The dynamic user embedding, DU, is then given by

$$\text{interact}(M_i, M_j) = [M_i - M_j \oplus M_i \cdot M_j \oplus M_j \oplus M_i] \tag{4}$$

$$DU = \sum_{j=1}^{i-1} S_j \frac{e^{W_6\,\text{interact}(M_i,M_j)+b_6}}{\sum_{k=1}^{i-1} e^{W_6\,\text{interact}(M_i,M_k)+b_6}} \tag{5}$$

The feedback aware track score is then computed similarly to the feed forward ranker, with the dynamic user embedding (DU) as an additional input:

$$score_{\text{FA}} = \sigma(W_9\,\text{ReLU}(W_8\,\text{ReLU}(W_7\,[T \oplus M_i \oplus DU] + b_7) + b_8) + b_9) \tag{6}$$

where FA stands for feedback aware. Similar to the feed forward ranker, this model is also optimized using the cross entropy loss. This model is still relatively computationally efficient, assuming the dynamic user embedding is pre-computed, which is possible assuming we know the playlists a user is most likely to listen to.

4.2.4 Reinforcement learning ranker. This ranker (RL) is our proposed sampling-based ranker that samples a single track from a set of tracks as the recommendation, which depends on the previous recommended tracks. This process is repeated on the remaining set of possible tracks to produce a ranked list. We formulate the problem of ranking as a standard reinforcement learning problem. We want to find a policy $\pi(t|s)$ that gives the probability of sampling track t given state s. The policy π is learned so it maximises some notion of reward $R(t, s)$, which gives some reward for recommending track t at state s. We therefore have to define the sampling probability $\pi(t|s)$ and the reward $R(t, s)$.

Sampler. Before we cover how $\pi(t|s)$ is computed, we first define how t and s are represented for the RL ranker. t is the track level features (denoted as T previously), but also concatenated with derived features from the feedback aware ranker as explained next. The derived features are the second and last layer of the feedback aware ranker for each track, and we denote this set of derived features as FA. These features are included to provide a richer representation to the RL model, which incorporates the user's past feedback. The state s is a sequence of tracks the user previously has been recommended in the session, in addition to the session meta representation (M). The state is encoded using a stacked LSTM with 2 layers, and initialised based on a linear projection of the session meta information:

$$o_i, h_i = LSTM_{stacked}(s_i|o_{i-1}, h_{i-1}), \quad s_{enc} = o_{|s|} \tag{7}$$

where $\text{LSTM}_{stacked}$ is a stacked LSTM with 2 layers, and s_{enc} is the last output of the stacked LSTM. The logit for each track t in the set of possible tracks, $\mathcal{T}$, is then computed as:

$$\begin{aligned}\text{logit}_t = W_{13}\,\text{ReLU}(W_{12}\,[(W_{10}\,s_{enc} + b_{10}) \\ \oplus\,(W_{11}\,[T \oplus FA] + b_{11})] + b_{12}) + b_{13}\end{aligned} \tag{8}$$

where both session encoding and track representation are passed through a linear feed forward layer, then concatenated and run through a feed forward layer using a relu activation function, followed by a linear output that gives the unnormalised logit for the track. The unnormalised logit is computed for all tracks in the set of possible tracks, and the sample probability is found by applying a softmax: $\pi(t|s) = \frac{e^{\text{logit}_t}}{\sum_{t' \in \mathcal{T}} e^{\text{logit}_{t'}}}$.

Reward. The reward associated with a sampled track, $t \sim \pi(\cdot|s)$ is defined based on whether the user found the track relevant:

$$R(t, s) = r(t, u) - c \tag{9}$$

where r is a binary relevance function, which is 0 if the user skipped the track and otherwise 1. c is a small constant that ensures that a negative reward is assigned to non relevant tracks. For all experiments c was fixed at 0.1. The model is trained using the REINFORCE algorithm [30].

4.3 Methods for diversity

We describe four methods used to obtain diversity in the recommended tracks. We assume the diversity score can be computed as a function between the track and user, $d(u, t)$, as detailed in Section 3.

4.3.1 Linear interpolation. Given the diversity function $d(u, t)$ and score function $s(u, t)$, the linear interpolation is defined as an α weighted combination of score and diversity:

$$s(u, t)_{diversify} = s(u, t) + \alpha d(u, t) \tag{10}$$

4.3.2 Submodular. Diversity can be introduced by formulating the diversity problem as a submodular set function. Submodular set

functions must uphold the following condition:

$$f(X \cup x) - f(X) \geq f(Y \cup x) - f(Y), \quad X \in Y \tag{11}$$

where X and Y are set of items, x is a single item, and f is a real valued function that takes as argument a set. This condition states that a submodular function should have some diminishing return when adding new items to the set. Submodular functions have been used extensively to provide diversity in recommendations [18, 26, 27], as they fit naturally when the set of recommended items should be diverse in regards to some similarity metric between the items. Our notion of diversity (see Section 3) is not naturally submodular, as diversity is a property of either the track itself or the user-track interaction, and thus do not have diminishing returns. To make our notion of diversity submodular, we change the task to recommend tracks of varying diversity. Given a set of recommended tracks τ for user u, we define f:

$$f(\tau, u) = \sum_{t \in \tau} s(u, t) + \frac{\alpha}{|\tau|} \sum_{t' \in \tau \setminus t} \text{abs}(d(u, t) - d(u, t')) \tag{12}$$

where $|\tau|$ is the size of the set τ, and abs is the absolute value. In this setting, we want to recommend the tracks with the highest relevance scores for a given user and that have as different diversity scores as possible, as this maximises the distance between these. This is a NP-hard problem, but can be solved greedily obtaining a near optimal solution [19].

4.3.3 Interleaving. Diversity can be introduced by alternatively recommending tracks with high diversity and high relevance scores. To do this we sort the tracks into two lists, l_{score} and $l_{diversity}$, and sample with probability $1-\alpha$ from the score list and otherwise from the diversity list at each time step, where α controls the trade-off between relevance and diversity. After each recommendation, the recommended track is removed from both lists.

4.3.4 Reinforcement learning. RL allows us to optimize multiple objectives directly by modifying the reward function. Thus, for the RL ranker we introduce diversity by including a diversity term in the reward function:

$$R(t, s) = r(t, u) - c + \alpha d(t, u) r(t, u) \tag{13}$$

where α is a trade-off parameter between diversity and relevance. Diversity is multiplied with relevance, such that it is only beneficial to recommend diverse tracks when they are relevant to the user.

5 EXPERIMENTAL EVALUATION

We observed strong associations between diversity, relevance and extent of user satisfaction based on the analysis presented in Section 3. The natural follow up question is how the different rankers and diversity methods presented in Section 4 fare, in terms of key satisfaction and diversity metrics, which we investigate next.

5.1 Dataset, metrics and evaluation

We use a dataset from Spotify, a large online music streaming service. The dataset consists of the listening history over a 2 month period of a sample of 1 million of users across 20 million sessions. All users in our sample dataset have at least 5 listening sessions, whereas all sessions have at least 5 tracks. We split users randomly into a training, validation, and testing set (85%, 7.5%, and 7.5%).

Table 2: Performance of rankers relative to the cosine ranker while only optimizing relevance. To avoid revealing sensitive metrics, we introduce a multiplicative factor to the base metrics (Hitrate, NDCG and User-track similarity) reported.

Ranker	Hitrate	NDCG	Popularity	User-track similarity
Cosine	56.006	0.632	1.741	0.584
Feed forward	+2.037%	+2.057%	+4.365%	-10.959%
Feedback aware	+2.553%	+2.848%	+4.078%	-9.247%
RL	+2.703%	+3.165%	+4.538%	-8.048%

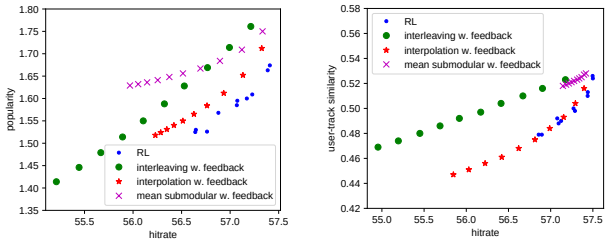

Figure 6: Popularity (left) and Average user-track similarity (right) vs hitrate using the feedback aware ranker.

We measure user satisfaction with the served recommendations using *Hitrate* – the percentage of recommendations relevant to the user (recommendations that the user fully listens to without skipping), as well as Normalised Discounted Cumulative Gain (*NDCG*). For diversity centric experiments, we use as metrics the average popularity of the recommended content (*Popularity*) and average user-track similarity for recommended tracks (*User-track similarity*). To avoid revealing sensitive metrics, we introduce a multiplicative factor to the base metrics (Hitrate, NDCG and User-track similarity) reported.

To keep users engaged in the session from the start, it is important to provide highly relevant initial recommendations. Therefore, given the sequential nature of our problem, we employ a *seed* song based approach, wherein the first track is selected based on relevance, and the diversity metrics are computed on the subsequent recommended tracks. Higher values of *Hitrate* and *NDCG* indicate greater satisfaction, while lower values of *Popularity* and *User-track similarity* indicate more diversity in the served recommendations. We evaluate the rankers on their top 10 recommendations. To have a large and potentially diverse pool of tracks to recommend, we base the evaluation only on sessions with at least 25 tracks.

5.2 Training details

The neural rankers are tuned by choosing the batch sizes within $\{128, 256, 512\}$, and learning rate from $\{0.001, 0.0005, 0.0001\}$. We kept all hidden layers fixed to 50 neurons, and used LSTM sizes of 50 as well. For the feed forward and feedback aware rankers, a batch size of 256 and learning rate of 0.0005 was optimal. For the RL ranker, a batch size of 512 and learning rate of 0.0001 was optimal. To train the RL ranker, as we only have access to logged data, which does not have any propensity scores to allow for off-policy techniques, we use the logged data as a simulator similar to [13, 32]. In our simulation setup, the pool of available tracks is limited to what was originally recommended to the user in a session, and that the user's relevance feedback is the same no matter the order the RL ranker presents the tracks in.

Table 3: Change (Δ) in hitrate, popularity and user-track similarity in comparison to reinforcement learning optimized only for relevance Table 2.

	Optimizing popularity		Optimizing user-track sim.	
Method (α)	**Δhit**	**Δpopularity**	**Δhit**	**Δuser-track sim.**
RL (0.1)	-0.212%	-8.324%	-0.037%	-2.235%
RL (0.3)	-0.780%	-12.637%	-0.418%	-7.076%
RL (0.5)	-1.523%	-16.071%	-1.119%	-10.801%
Interpolation (0.1)	-0.675%	-9.231%	-0.395%	-6.145%
Interpolation (0.3)	-1.751%	-14.835%	-1.563%	-12.849%
Interpolation (0.5)	-2.243%	-16.593%	-2.909%	-16.760%
Submodular (0.1)	-0.694%	-6.099%	-0.217%	-1.862%
Submodular (0.3)	-1.992%	-9.451%	-0.421%	-2.793%
Submodular (0.5)	-2.698%	-10.495%	-0.652%	-3.538%
Interleaving (0.1)	-0.916%	-5.824%	-1.073%	-3.911%
Interleaving (0.3)	-2.460%	-14.835%	-2.785%	-8.380%
Interleaving (0.5)	-4.016%	-22.308%	-4.461%	-12.663%

5.3 Comparison of ranking approaches

We begin by investigating the trade-off between model complexity and performance, and investigate how the different rankers fare on diversity metrics when not optimized explicitly for diversity. Table 2 shows the performance of the four rankers on satisfaction and diversity metrics. We observe that the hitrate and NDCG for the rankers follows their computational complexity. The proposed RL ranker have the highest user satisfaction, although we observe a relative small difference in hitrate and NDCG for all neural rankers.

As the increasingly complex rankers lead to higher user satisfaction, they also result in recommendations with a higher average popularity. Most notably, the largest popularity increase occurs when going from the cosine ranker to any of the neural rankers, whereas the popularity difference between the neural rankers is negligible in comparison. For the user-track similarity diversity metric, the cosine ranker will by definition have the largest user-track similarity. However, among the three neural rankers, we observe that the more complex models lead to recommendations that are more similar to what the user has previously encountered. These results suggest that while increased model complexity gives better user consumption predictability, it comes at a cost of decreased diversity. As the hitrate and NDCG both show the same trends, we will focus on only the hitrate for the remaining results.

Note: While seemingly small, a 2-3% gain in offline metrics (e.g. NDCG) has resulted in over 10-15% gain in important online measures of user satisfaction in past A/B tests. This is further supported by prior research that suggests that small changes in NDCG might result in significant changes in online user behevaior [21].

5.4 Comparison of diversity methods

To evaluate the four diversity methods, we compare their performance for introducing diversity against each other, keeping the ranker fixed. For the three methods requiring a track relevance score (interpolation, submodular, and interleaving) we use the feedback aware ranker as the base ranker. These three methods are compared directly against the RL ranker, which is optimized for both relevance and diversity through its reward definition. As optimizing for both relevance and diversity is a trade-off, the results are presented using scatter plots. For the non-RL methods, the trade-off parameter α was chosen as $\alpha \in \{0.05, ..., 0.5\}$ with increments of 0.05. For the RL ranker, we choose $\alpha \in \{0.1, ..., 0.5\}$ with increments of 0.1, and train each configuration twice to explore the variance.

Figure 6 shows the trade-off between hitrate and the diversity metric for the diversity methods, while Table 3 shows the relative values of the hitrate and diversity metrics in comparison to the RL method not optimized for diversity.

Popularity. We observe that the RL method obtains the best trade-off between high hitrate while reducing the average popularity. Linear interpolation obtains the second best trade-off, and interleaving obtains low average popularity at the cost of large reductions in hitrate. Submodular is unable to obtain any large decrease in the average popularity, as larger α values only leads to marginal drops in average popularity. Overall, these results shows a small benefit of using RL to reduce the average popularity, but at the cost of higher computational complexity and training time compared to the simple linear interpolation.

User-track similarity. We observe that the RL method and linear interpolation obtain very similar trade-offs, but that the linear interpolation cover a wider range of trade-offs than the RL method. Diversity by the submodular method results in the worst trade-offs, as the effective user-track similarity reduction is very limited. Similar to the popularity diversity metric, we observe that the interleaving method perform significantly worse than linear interpolation.

Overall, these findings suggest that leveraging RL reward modeling for diversification gives slightly better performance, but interpolation based methods offer a wider range of trade-offs, which provides more flexibility and control to system designers. For submodularity, we observed limited ability to reduce both the popularity and user-track similarity. As described in Section 4.3.2, neither of the diversities are naturally submodular, hence we formulated a submodular function for recommending a sequence of items with varying diversity (as opposed to simply increasing diversity). However, based on the results in our setting, this submodular formulation is less suited to the problem compared to other traditional approaches like interleaving and interpolation.

5.5 Interplay between ranker and diversity methods

We have compared rankers on satisfaction metric, and investigated the effect of the four diversity methods when the ranker was fixed. A natural question to answer is whether the observed trends in diversity methods generalize across all rankers, or does specific diversity methods work with specific rankers. We next investigate this interplay of rankers and diversity methods. For all experiments we use the same choice of α values as done previously.

Popularity. Figure 7 shows the trade-off between average popularity and hitrate for all combinations of rankers and methods introducing diversity. In all cases, the RL ranker is the same and is used as a reference between the plots. We observe that the difference in hitrate from the rankers carries almost directly over for the interpolation and interleaving, while the difference is smaller between the hitrate for the submodular method. Independently of the ranker, the span of average popularity for each of the three diversity methods is approximately the same, showing that the ranker almost entirely influences hitrate. As the average popularity decreases, we observe that the hitrate differences get comparatively smaller than for larger average popularity values. Independent of

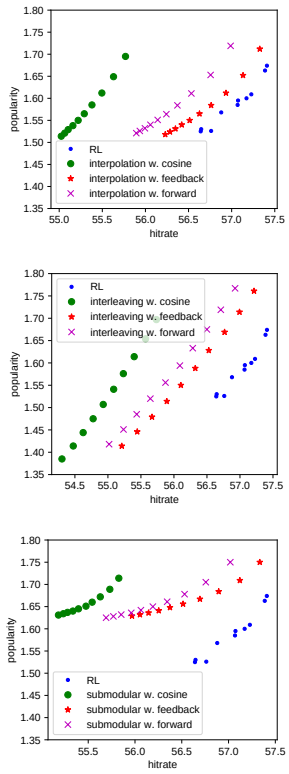

Figure 7: Popularity vs hitrate when varying the ranker across diversity methods.

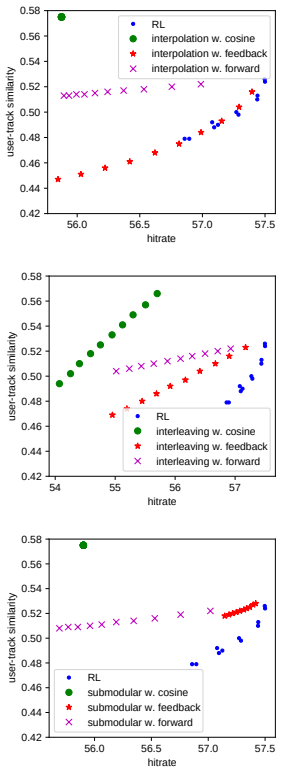

Figure 8: Average user-track similarity vs hitrate when varying the ranker across diversity methods.

the ranker choice, we observe that linear interpolation obtains the best trade-off among the non-RL diversity methods, while RL obtains the best overall trade-off.

User-track similarity. Figure 8 shows the trade-off between average user-track similarity and hitrate for all combinations of rankers and diversity methods. Due to how linear interpolation and submodular both use the diversity metric to subtract from the rank score, they do not work when the diversity metric is the same as the relevance score (as is the case for the cosine), and all values of α therefore leads to the same ranking.

The submodular method again provides the worst trade-offs out of all the diversity methods. When the feed forward ranker is used, the hitrate decrease is notably larger than for the feedback aware ranker, but the effective span of average user-track similarity values is very small for both rankers. For both linear interpolation and interleaving, we observe the difference in hitrate between the feed forward ranker and feedback aware ranker is much greater than the difference observed when only optimizing relevance. While the difference in hitrate between the feed forward and feedback aware ranker is only 0.29 when diversity is not considered (see Table 2), the difference in hitrate can be over 1 depending on the average user-track similarity. This is even though the feedback aware ranker has a slightly higher average user-track similarity when diversity is not considered. Thus, we observe that the choice of ranker can interact with the choice of diversity method non-trivially.

Overall, we observe that RL and linear interpolation work better than interleaving and submodular diversity methods, with both RL

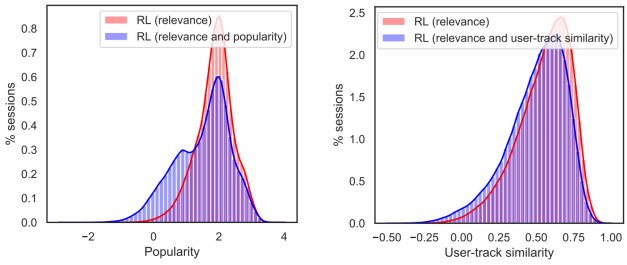

Figure 9: Shift in average session diversity for the RL method ($\alpha = 0.3$) compared to optimizing only relevance

and interleaving with feedback aware ranker obtaining approximately the same trade-offs, while the linear interpolation covering a larger span of average user-track similarities. More interestingly, comparing these results with the ranker comparison on only satisfaction (Section 5.3), we observe bigger differences in hitrate when rankers consider diversity, than when they are only focused on satisfaction. This suggests that when one cares only about satisfaction, there exist little difference between the rankers; however when one cares additionally about diversity, the difference between rankers becomes more pronounced.

Given the varying complexity of development and deployment of these rankers, this result has big ramifications on the choice of rankers for system designers based on the task at hand.

6 DISCUSSION

Looking at music consumption data, and presented results, we found evidence that not only are users satisfied with relevant recommendations, but also often with recommendations that depart from their historic tastes, or are less popular. Such departure from relevant and popular content allow platforms to broaden the scope of music listening and shift consumption towards the tail and less familiar content. Figure 9 visually depicts this shift in consumption, wherein we observe a significant shift in popularity distribution from unimodal to bimodal when additionally optimizing for popularity diversity, and a slight shift towards lower user-track similarity when optimizing for similarity diversity.

Specifically in the context of music streaming, we posit our findings relates to and builds upon insights on how users consume music. First, recent results suggest that users often have flexible and broad intents when they interact with the music streaming apps [14]. Indeed, the broader the intent, the more we expect the user to be open about music recommendations, which enables the system to shift consumption while still serving satisfying content. Another line of recent research has characterized users as "*specialists*" vs "*generalists*" based on their consumption diversity [1], with generalists preferring diverse sets of music. This highlights the strong preference of some users to prefer diversity, which in turn makes shifting of consumption to less similar or less popular content more amenable. Finally, music streaming applications are essentially multi-stakeholder platforms which connect users and artists [16]. Such platforms need to maintain a healthy balance between user satisfaction and artist exposure goals [15]. A recommender model equipped with consumption shifting ability enables the platform to surface under-served artists, thereby maintaining a healthy balance between consumer and supplier objectives.

On the system design perspective, our findings give system designers practical considerations on the choice of rankers, ways of diversification and serving infrastructure. We argue that the cosine rankers are a good first solution to the recommendation problem – being greedy algorithms, they are quick to deploy, and offer comparable performance to neural and RL rankers if satisfaction is the only goal. However, if diversity is important to consider, and as systems mature and the need for improved models arises, switching to neural ranker makes sense. On the choice between different ways of diversification, the reward modeling based RL method performs better than interpolation for swaying consumption away from popular content, though such methods are non-trivial to productionize at scale. We advocate system designers make this choice based on the underlying infrastructure in place.

Future Work The limitations of this work lead to several next steps. First, while we used logged data to train our RL model, the benefits RL has to offer are more prominent when trained via off-policy training, or a live deployment. Second, there is increasing evidence that propensity for diverse content is an innate user trait, with some users preferring diverse content more than others [1, 15]. This motivates the need for developing user-aware diversification models that personalize the extent to which served recommendations are diverse. Finally, while we explicitly focused on trivial reward combinations, there exist ways to account for richer interactions between objectives. Future work will involve considering richer reward structures to improve performance gains offered by RL approaches.

REFERENCES

[1] Ashton Anderson, Lucas Maystre, Rishabh Mehrotra, Ian Anderson, and Mounia Lalmas. 2020. Algorithmic Effects on the Diversity of Consumption on Spotify. In *The World Wide Web Conference*.

[2] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal greedy diversity for recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[3] Punam Bedi, Shikha Agarwa, Archana Singhal, Ena Jain, and Gunjan Gupta. 2015. A novel semantic clustering approach for reasonable diversity in news recommendations. In *Computational Intelligence in Data Mining-Volume 1*.

[4] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 405–414.

[5] Bert Boyce. 1982. Beyond topicality: A two stage view of relevance and the retrieval process. *Information Processing & Management* 18, 3 (1982), 105–109.

[6] Jaime G Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries.. In *SIGIR*, Vol. 98. 335–336.

[7] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and Diversity in Information Retrieval Evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 659–666. https://doi.org/10.1145/1390334.1390446

[8] Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science* 55, 5 (2009), 697–712.

[9] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and Sequential User Embeddings for Large-Scale Music Recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 53–62.

[10] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-Based Systems* 123 (2017), 154–162.

[11] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. 2010. Temporal diversity in recommender systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 210–217.

[12] Mark Levy and Klaas Bosteels. 2010. Music recommendation and the long tail. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*. Citeseer.

[13] Elad Liebman, Maytal Saar-Tsechansky, and Peter Stone. 2015. Dj-mc: A reinforcement-learning agent for music playlist recommendation. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 591–599.

[14] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *The World Wide Web Conference*. 1256–1267.

[15] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2243–2251.

[16] Rishabh Mehrotra, Niannan Xue, and Mounia Lalmas. 2020. Bandit based Optimization of Multiple Objectives on a Music Streaming Platform. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3224–3233.

[17] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural Language Inference by Tree-Based Convolution and Heuristic Matching. In *ACL*. 130–136.

[18] Houssam Nassif, Kemal Oral Cansizlar, Mitchell Goodman, and SVN Vishwanathan. 2018. Diversifying music recommendations. *arXiv preprint arXiv:1810.01482* (2018).

[19] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14, 1 (1978), 265–294.

[20] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

[21] Filip Radlinski and Nick Craswell. 2010. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 667–674.

[22] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. 2015. Multiobjective pareto-efficient approaches for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2015), 53.

[23] Tetsuya Sakai and Zhaohao Zeng. 2019. Which Diversity Evaluation Measures Are "Good"? 595–604. https://doi.org/10.1145/3331184.3331215

[24] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311, 5762 (2006), 854–856.

[25] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2219–2228.

[26] Choon Hui Teo, Houssam Nassif, Daniel Hill, Sriram Srinivasan, Mitchell Goodman, Vijai Mohan, and SVN Vishwanathan. 2016. Adaptive, personalized diversity for visual discovery. In *Proceedings of the 10th ACM conference on recommender systems*. 35–38.

[27] Sebastian Tschiatschek, Adish Singla, and Andreas Krause. 2017. Selecting sequences of items via submodular maximization. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[28] Isaac Waller and Ashton Anderson. 2019. Generalists and Specialists: Using Community Embeddings to Quantify Activity Diversity in Online Platforms. In *The World Wide Web Conference*. ACM, 1954–1964.

[29] Jacek Wasilewski and Neil Hurley. 2016. Incorporating diversity in a learning to rank recommender system. In *The Twenty-Ninth International Flairs Conference*.

[30] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[31] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 13–22.

[32] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1040–1048.

[33] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. 2010. The impact of YouTube recommendation system on video views. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 404–410.

[34] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.