Algorithmic Balancing of Familiarity, Similarity, & Discovery in Music Recommendations

Rishabh Mehrotra
Spotify
London, United Kingdom
rishabhm@spotify.com

ABSTRACT

Algorithmic recommendations shape music consumption at scale, and understanding the role different behavioral aspects play in how content is consumed, is a central question for music streaming platforms. Focusing on the notions of familiarity, similarity and discovery, we identify the need for explicit consideration and optimization of such objectives, and establish the need to efficiently balance them when generating algorithmic recommendations for users. We posit that while familiarity helps drive short term engagement, jointly optimizing for discovery enables the platform to influence and shape consumption across suppliers. We propose a multi-level ordered-weighted averaging based objective balancer to help maintain a healthy balance with respect to familiarity and discovery objectives, and conduct a series of offline evaluations and online AB tests, to demonstrate that despite the presence of strict trade-offs, we can achieve wins on both satisfaction and discover centric objectives. Our proposed methods and insights have implications for the design and deployment of practical approaches for music recommendations, and our findings demonstrate that they can lead to substantial improvements on recommendation quality on one of the world's largest music streaming platforms.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

objective balancing; multi-objective aggregation

ACM Reference Format:
Rishabh Mehrotra. 2021. Algorithmic Balancing of Familiarity, Similarity, & Discovery in Music Recommendations. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3481893

1 INTRODUCTION

Designing recommendation products for music streaming platforms necessitates understanding the diverse needs of users and assisting them in discovering and finding the music they want to listen. Recommender systems rely on their ability to model user's individual preferences using data about their past consumption. Specifically, once a user profile is built, such systems infer or predict relevance of a specific music track to a given user's profile. While this affords great modeling convenience to system designers, and enables development of increasingly sophisticated models to estimate relevance of content to user, it misses out on leveraging other related, and often more desired aspects and qualities of recommendations, e.g. familiarity, serendipity and discovery, which make recommendations good recommendations.

Aspects of Recommendations: Recent research into recommendations is starting to go beyond the notion of predicted relevance and consider a wider range of "*beyond relevance*" objectives, including qualities such as whether the user is familiar with the content, or whether the list of recommendations contains novel items, aspects which may have a significant impact on the overall quality of a recommender system. Specifically focusing on music streaming, we observe that users often have their favorite songs and artists they listen to, and therefore, music consumption is full of user's consuming the same content repeatedly over time. Such dynamics of repeat consumption has been studied in various hedonic settings [3, 10], including repeat web searches [23] and repeat website visits [1]. Prior and direct experiential engagement with specific music content, via repeat consumption, affords the notion of *familiarity* to users, thereby helping adoption of the served recommendations due to the effects of *perceived personalization* [12].

However, given the large repositories of music content available to users, only a fraction of such content is familiar to them, and over-exposing familiar content creates the issue of "*filter-bubbles*" [21]. This necessitates a focus on one particular need: music discovery, which we define as the experience of finding and listening to content that is previously unknown to the user. Discovery allows users to find fresh content, and driving discovery can help reduce staleness of recommendations, leading to greater user satisfaction and engagement, thereby resulting in increased user retention [4], and continued platform subscription [16]. Taken together, such notions of relevance, familiarity and discovery provide complimentary views on recommendation quality – while familiarity provides users with immediate, short term satisfaction, discoveries enables the streaming platform to influence and shape long term behavior on the platform.

Present Work. We identify and advocate for an explicit consideration of three different aspects of recommendations: (i) relevance, (i.e. estimated similarity), (ii) familiarity, and (iii) discovery, and study their impact on multiple user behavior metrics. We note that explicit balancing of such attributes is an understudied problem

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481893

specifically in the music context, and that it is challenging to incorporate such aspects in the typical predictive modeling setting. To address this, we present objective balancing approaches aimed at establishing a healthy balance of such aspects when surfacing algorithmic recommendations to users.

We investigate how the above mentioned aspects of recommendations affect user behavior, and conduct large-scale analyses and multiple live experiments on the music streaming platform Spotify for investigating such questions. We view user consumption on Spotify from the lens of the identified recommendation aspects, and present insights about user's preferences for familiar music, and the interplay between similarity, familiarity and discovery. We conduct a series of live A/B tests on a large user population on two distinct user-centric recommendation products, to test how the proposed objective balancing methods fare on key user engagement metrics. The proposed methods are able to obtain metric improvements on both user satisfaction and discovery centric objectives, despite the presence of strict trade-offs. Finally, we view discovery as an enabler for shifting consumption to non-popular or tail-artists, and present detailed results on how additionally optimizing for discovery helps in surfacing less popular artists.

2 RELATED WORK

Understanding different aspects of recommendations, and the role different objectives play in shaping recommendations is a long studied topic in the recommender systems community. Going beyond traditional notions of relevance and accuracy, recent research has investigated notions of diversity, serendipity, novelty, and coverage in designing and evaluating recommender systems [11]. Herlocker introduced novelty and serendipity (both related to the concept of discovery) as dimensions for evaluating recommendation quality [7]. Many researchers have proposed formalisms for defining and evaluating novelty, serendipity, & diversity.

Another well studied notion is that of re-consumption pattern of users, including search query re-finding [23], website revistations [1], modeling dynamics of reconsumption [3] and the interplay between re-consumption and variety in recommendations [22]. Our work builds on top of existing work on re-consumption and zooms in on familiarity as an important recommendation aspect. The role of familiarity and user trust has long been studied in the psychology community [12]. More recently, researchers have investigated item familiarity effects in user-centric evaluations of recommender systems [9]. Beyond familiarity, music discovery has been shown as some users' main motivation to continue platform subscription [16] and is considered an important need for music listeners [5, 14].

Another line of work has studied the different definitions of user satisfaction and ways of measuring them, using both qualitative means and implicit feedback signals obtained via logged user interactions [25]. Recent work has also started investigating the different trade-offs that exist in recommender domain. Hurley and Zhang discussed novelty and diversity and their trade-offs with system accuracy, casting this trade-off as a multi-objective optimization problem [8]. Beyond user goals, recent research has also investigated user-centric and supplier centric trade-offs in recommendations platforms [17, 19]. Other search and recommendation applications that need to meet multi-objective requirements include click shaping [2] and email volume optimisation [6].

Building on top of existing work, we investigate the direct impact jointly optimizing for familiarity, similarity and discovery has on key user engagement and supplier exposure metrics.

3 ROLE OF RELEVANCE, FAMILIARITY & DISCOVERY

Our goal is to understand the nuances in user's preference towards familiarity and discovery at a global scale, and study its relationship with long-time user and platform outcomes. We begin by formally defining the different aspects of recommendations we consider in the present work, and describe ways of quantifying each.

3.1 Data context

We study Spotify, an online streaming platform where users can listen to a vast selection of music from around the world. We consider listening history of a random sample of over 100 million distinct users who cumulatively listened to millions of songs around 70 billion times during a one month period. We focus on two surfaces users use to get sequential recommendations:

Radio creates a collection of songs based on any artist, album, playlist, or song of user's choice. Users decide a seed artist, or track, and the radio service generates a list of tracks for a given seed.

Autoplay product comprises of the sequential recommendation scenario where a user reaches the end of an album, playlist, or selection of tracks. Upon reaching the end of the playlist, Autoplay automatically play similar songs, so as to continue user's streaming session.

3.2 Key Concepts

We define key concepts of similarity, familiarity, and discovery, which are used throughout the paper.

3.2.1 ***Similarity.*** An important aspect of personalization is the ability to recommend content tailored to users based on knowledge about their preferences and behavior. Personalized recommendations rely on making recommendations that are relevant to the user. We operationalize the notion of relevance based on estimated similarity between the user and content. A recommendation is identified as relevant, or similar if it closely resembles user's interest profile. We interchangeably use the term relevance for similarity throughout this text.

Quantifying Similarity. We quantify similarity between a user and music content by learning their embeddings and computing cosine similarities between the learnt embeddings. Embeddings are producted by training word2vec [20] on user-generated playlists, where the task is to predict the song in the middle of the context window given the surrounding songs. This naturally causes songs that frequently co-occur in playlists to have nearby embeddings in the space. We define a user's taste profile to be the average of the song embeddings that they have listened to within certain time window. Cosine similarity between embeddings of two songs provides the similarity estimate.

Similarity scores often miss out the relative ordering of tracks, i.e., often even the best available tracks might have a low similarity

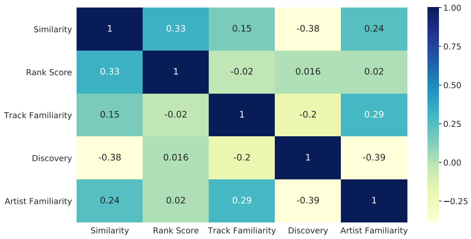

Figure 1: Correlation analysis across different attributes.

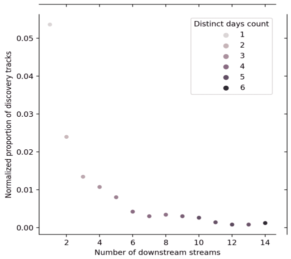

Figure 2: Discoveries enable downstream consumption.

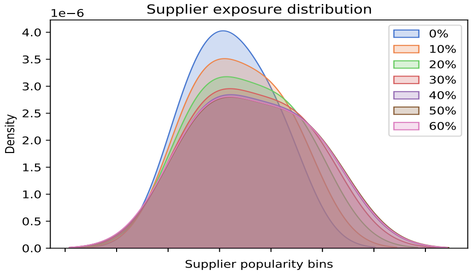

Figure 3: Impact on supplier distribution: simulating impact of varying proportions of discovery on supplier distribution.

score. To account for such aspects of relative ordering of tracks, we estimate a ***rank score*** for each track, which is derived from the rank obtained by each track when they sorted by their similarities score. Specifically, we take the cube-root of the inverse of rank to compute the rank score: $\gamma(r) = \sqrt[3]{\frac{1}{\alpha+r}}$, where r is the rank of track when ordered based on decreasing order of similarity scores, and α is a parameter to control the range of the rank score values. Tracks ranked higher would have a higher rank score regardless of their similarity score.

3.2.2 ***Familiarity****.* Familiarity refers to the state of having prior exposure to the content and knowing it well. Familiarity reduces the uncertainty of expectation through increased understanding of what has happened in the past, and allows users to accumulate trust-relevant knowledge from past interactions [12]. Repeated prior exposure to certain music content increases familiarity of users with that content, which in turn may inculcate affinity of the user towards that content: positive affinity if the user enjoys that content, and negative affinity if the user dislikes that content. Past research has also identified familiarity's role in fostering user trust in a recommendation system's competence, and helps adoption [12].

Quantifying Familiarity. Affinities help us quantify familiarity and approximates how much a user likes and is familiar with a piece of music content. It's calculated as a count of plays weighted with a few assumptions about how and where a user initiated those plays. For example, if a user plays track A from their library vs passively hearing track B as part of a long background session, the user's affinity from track A is assumed to be higher. The calculation is slightly down-weighted when a user skips a track. We compute two forms of affinities: (i) track affinity and (ii) artist affinity.

A user's affinity for a track is the sum of weights for all user plays of a track in a given time range. For each play event of a track being streamed, we compute the affinity weight for that play event as the product of 2 coefficients (c_1, c_2) plus a skip correction c_3, thus $p_i = c_1c_2 + c_3$, where c_1 is high for high intent play context; c_2 quantifies the reason this stream was started. Finally, c_3 is the skip correction term which adds a negative penalty for each time the track is skipped. The user's track affinity then becomes: $t_k = \sigma(\sum_i p_i)$ where σ is the sigmoid function, p_i is the affinity weight for play event i and t_k is the track affinity for the k-th track.

A user's affinity for an artist is equal to the product of that user's number of track-plays for an artist and the third highest track affinity weight from all tracks the user has listed to by that artist. Let P be all the user's track affinities for a given artist. Let $max_n(P)$ return the n-th largest value. Finally, let $M = |P|$ be the number of tracks played. The user's artist affinity is then: $a_i = \sigma_{a,b}(M.max_n(P))$ where $\sigma_{a,b} = 1/(1 + \exp(a - x/b))$ is the parameterized sigmoid function which scales the affinities back to 0-1. We intentionally refrain from disclosing specific parameter values to avoid disclosing sensitive internal information.

3.2.3 ***Discovery****.* Discovery is the experience of finding and listening to content that is previously unknown to the user. Discovery enables users to find fresh content, helps in reducing staleness of recommendations. Facilitating the process of discovery helps streaming platforms improve user retention [4], platform subscription [16] and satisfy important user needs and expectations from the platform [13]. Moreover, when done right, discovery can prove to be an enabler for shifting consumption to non-popular or tail-artists, thereby helping develop healthy and sustainable platforms.

Quantifying discovery. To quantify discovery, we consider past plays of the user with the artist or track, and identify a track as a discovery track (or artist) if the user has not streamed this track (or artist) in the last 6 months. We mandate that to be identified as a discovery track, the user should not have streamed neither the track nor the artist in the past 6 months. Further, to better differentiate between all discovery tracks, we multiply the estimated similarity between user and the track by the 0/1 discovery indicator, to obtain the discovery score we use in the paper:

$$d_S(u,t) = \mathbb{1}[d] * \zeta(u,t) \tag{1}$$

where d_S is the discovery-score for the user u and track t, and $\zeta(u,t)$ is the estimated similarity between the user and track.

3.3 Objective Interplay & Need for Discovery

Figure 1 shows the correlation across these attributes, and indicates that these attributes all carry heterogenous information. We observe a positive correlation between similarity and both track and artist familiarity, with artist familiarity having a relatively higher correlation with similarity than track familiarity. Indeed, a higher familiarity with an artist would stem from the fact that users have enjoyed multiple tracks from that artist, thereby it represents a larger set of user tracks than any specific individual track. Further, we observe that all of similarity, and track and artist familiarities are negatively correlated with discovery, which is indeed expected.

Focusing on discovery content, we posit that facilitating interaction on discovery tracks is useful for long term engagement of users, and also for platform health. We identify two specific advantages of explicitly considering discovery as an objective in recommendation design:

Discoveries enable downstream listens: Discoveries provide a means for artists to broaden their audience. Often when users discover a new track or an artist, they save the track and come back to re-stream the track later on. We call such re-streaming events as downstreams. Based on our analysis of follow-up streaming on discovered tracks, we find evidence that "*good*" discoveries often lead to downstream listens from the user, as shown in Figure 2. Such downstream effects are important drivers of creating connections between a user and an artist.

Discoveries enable shifting consumption to tail-artists: To better understand the impact of discoveries on the consumption distribution of artists, we perform a simulation experiment wherein we intentionally increase the amount of discoveries in user sessions by ranking discovery tracks from less popular artists higher than familiar tracks. Specifically, we vary the proportion of discovery tracks surfaced to users in their sessions from 0% to 60%, and plot the corresponding exposure distribution of artists across the different popularity bins. Figure 3 plots the artist exposure distributions across the artist popularity spectrum as we increase the amount of discoveries surfaced. We observe that upon intentionally increasing discoveries, we are able to significantly shift the distribution towards right, i.e., towards less popular artists.

Taken together, our analysis highlights that trade-offs exists between these different recommendation aspects, and that there is value in driving discoveries on the platform. How best can we balance these aspects is an understudied question. We next present few specific algorithmic ways of balancing these objectives in a typical recommendation setting.

4 ALGORITHMIC BALANCING APPROACHES

We posit that recommender systems need to explicitly optimize for and balance discovery with relevance and familiarity, so as to provide short term as well as long term happiness to users. In this section we formulate the problem as a multi-criterion objective balancing problem, and present fuzzy aggregation function based aggregators which jointly optimize for multiple criterion.

4.1 Notation

For a given track x, we consider multiple criterion $(x_1, x_2, ..., x_k)$ based on which each candidate track is scored.

Definition 1. We denote by $x_{\nearrow}$ the vector obtained from x by arranging its components in non-decreasing order, that is, $x_{\nearrow} = x_P$ where P is the permutation such that $x_{P(1)} \leq x_{P(2)} \leq ... \leq x_{P(n)}$. Similarly, we denote by $x_{\searrow}$ the vector obtained from x by arranging its components in non-increasing order.

Definition 2. *Weighting vector*: A vector $w = (w_1, ..., w_n)$ is called a weighting vector if $w_i \in [0, 1]$ and $\sum_{i=1}^{n} w_i = 1$.

4.2 Objective Balancing Problem Formulation

We cast our problem of balancing between relevance, familiarity and discovery as an objective balancing problem, wherein the recommender system has to consider a number of different and sometimes conflicting objectives when scoring candidate items for ranking. We note that such objective balancing problems are not just applicable in our current setting, but are often masked by multi-stage ranking setups, which involve post-processing step wherein different criterions are satisfied by post-processing the final ranking.

Typical recommendation systems follow a two-stage design with a candidate generation and a ranking. In the ranking stage, the recommender has a few hundred candidates retrieved from the candidate generation, and applies sophisticated large-capacity models to score the candidates. Different from typical setting, instead of working with one final score, we have multiple scores and criterions based on which we wish to score and rank tracks. Consequently, the final sorting requires consideration of multiple different criterion, with each candidate getting a score for each criterion, and some combination of such scores describing the final aggregated score of each candidate based on which recommendations are served. The aggregated score is seen as some sort of representative value of the different scoring criterion for each candidate.

Given a set of T candidate tracks, $x_1, ..., x_T$, our goal is to score each of these tracks based on some aggregation function $G(.)$, such that $G(.)$ considers and respects multiple, often conflicting criterion. Given the function $G(.)$, the final ranking is obtained by sorting the candidates tracks based on their final aggregation scores. We next present three specific instantiations of the aggregation function for objective balancing, which enables us to balance the different recommendations aspects when ranking tracks for recommendations.

4.3 Weighted Sum Aggregation

A natural extension to simple average is the weighted average function, wherein weights $w_i \in [0, 1]$ are associated with each criterion, which reflects the relative contribution of the i-th score to the aggregated value. Given a weighting vector $\mathbf{w}$, the weighted arithmetic mean is the aggregation function: $M_{\mathbf{w}}(\mathbf{x}) = w_1 x_1 + w_2 x_2 + ... + w_n x_n = \sum_{i=1}^{n} w_i x_i = < \mathbf{w}, \mathbf{x} >$, where $M_{\mathbf{w}}$ is a symmetric, Lipschitz continuous function, and is strictly increasing if all $w_i > 0$. Weighted arithmetic means are good for averaging scores that can be added together.

While weighted averaging provides a simple and easy way to aggregate multiple scores, their behavior remains fairly static over time given specific weight assignments. Also, since the weights are associated with the specific objectives, the importance of the criterion is fixed regardless of the exact value the criterion might score. We next present extensions of weighted averaging aggregators which address such issues.

4.4 OWA balancing

Ordered weighted averaging functions (OWA) are aggregation functions, that associate weights not with a particular criterion, but rather with its value [24]. They differ to the weighted sum aggregators in that the weights are associated not with the particular inputs, but with their magnitude. In some applications, all input criterion are equivalent, and the importance of a criterion is determined by its value. For example, when one wishes to serve recommendations using several satisfaction criterion (e.g. relevance, affinity), the largest criterion score is the most important, regardless of whichever specific one it is. OWA provide us a way to specify aggregation functions in such scenarios.

OWA are symmetric aggregation functions that allocate weights according to the input value, i.e. OWA can emphasize the largest, the smallest or mid-range inputs. Thus in the OWA aggregation the weights are not associated with a particular argument but with the ordered position of the arguments. The form of the aggregation is very strongly dependent upon the weighting vector used. Given a weighting vector w, the OWA function is

$$OWA_{\mathbf{w}}(x) = \Sigma_{i=1}^{n} w_i x_{(i)} =< \mathbf{w}, x_{\searrow} > \tag{2}$$

where $x_{\searrow}$ is the vector obtained from x by arranging its components in non-increasing order, that is, $x_{\searrow} = x_P$ where P is the permutation such that $x_{P(n)} \leq x_{P(n-1)} \leq ... \leq x_{P(1)}$. If all weights are equal, OWA becomes the arithmetic mean. To use OWA for objective balancing, we need to be able to generate the weights to use with OWA. We next describe the function used to generate the OWA weights.

4.4.1 Weight Quantifier Function. The process of weight assignment in OWA is of fundamental importance, since it controls how the different criterion are weighed when scoring candidate tracks. One popular way to specify such weights is via linguistic quantifiers, that are able to express the concept of fuzzy majority: "*most*", "*some*", "*at least one*", "*as many as possible*". An example of such quantifiers is the Regular Increasing Monotone (RIM) quantifiers [15]. These functions generate OWA weights for any n using:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) \tag{3}$$

where i is the index of the i-th largest value among the n criterion, and Q is the quantifer function that assigns weights to the criterion ranked i. To identify the quantifier we employ one of the simplest and the most often used methods for defining a parameterized subset on the unit interval [24]. Specifically,

$$Q(p) = p^{\alpha} \quad (\alpha > 0) \tag{4}$$

Q(p) is represented as a fuzzy set in interval [0, 1]. By changing the parameter, α, one can generate different types of quantifiers and associated operators between the two extreme cases of the all and at least one quantifiers. For $\alpha = 1$, Q(p) is proportional to α and therefore it is referred to as the identity quantifier, which is also the arithmetic mean. As α tends to zero, the quantifier Q(p) approaches its extreme case of at least one, which corresponds to the MAX operator. As α tends to infinity, the quantifier Q(p) approaches its extreme case of all, which corresponds to the MIN operator.

By choosing appropriate α values, we govern the functional form of the aggregation function, and can vary it to score items higher when all criterion scores are high (e.g. $\alpha = 2.5$), or score items which have atleast one criterion score as high (e.g. $\alpha = 0.1$). The use of a RIM quantifier to guide the aggregation essentially implies that the more criteria satisfied the better the solution. This condition seems to be one that is naturally desired in criteria aggregation.

Overall ranking:
To generate the final ranking of tracks, we follow a three step procedure:

(1) **Step 1:** Use Q to generate a set of OWA weights, $w_l, w_2, ..., w_n$
(2) **Step 2:** For each candidate track $x \in X$, using $x_{\searrow}$ calculate the overall score using the OWA function
(3) **Step 3:** Sort the candidate tracks by using the output of the individual track's OWA scores

4.5 Hierarchical OWA balancing

We further extend the OWA weighting to multiple levels. Specifically, for a given set of criterion scores $(x_1, x_2, ..., x_n)$ for a track x, we consider applying OWA function to specific subset of criterions, and then using the output as an additional intermediate criterion, and applying OWA function on top of this intermediate criterion. For example, consider a 2-Level OWA function and set of candidate tracks, with each track scored using 5 criterion $(x_1, x_2, x_3, x_4, x_5)$. We assume the first level OWA is applied to first three criterion (x_1, x_2, x_3), and its output is used as part of the second OWA, with the remaining criterion, i.e. x_4, x_5. Specifically:

$$s_1 = OWA(x_1, x_2, x_3) \tag{5}$$

$$s2 = OWA(x_4, x_5, s_1) \tag{6}$$

Such splitting of criterion across different OWA functions provides us with great modeling flexibility, which we can use to combine different user satisfaction attributes using different logic across many abstractions.

5 EXPERIMENTAL EVALUATION

To understand how different recommendations fare in terms of engagement and artist exposure metrics, we conduct extensive offline evaluations on a large scale real world user interaction data from Spotify and also multiple live AB test.

5.1 Track Sequencing Experiment

We evaluate the different rankers on the task of track re-ranking: we consider a subset of user sessions, and extract information on all the tracks users interacted with in those sessions. For each track, we record the user interaction information, i.e. whether the user streamed the track (above a particular time threshold) or skipped a track. Track streams are considered as positive, satisfying interaction, while skips are tagged as negative interactions. We then use the different ranking techniques to re-rank all interacted tracks, and estimate six metrics measured at top-5:

(1) Satisfaction metric: average number of tracks which the user streamed. Higher number indicate more user satisfaction.
(2) Discovery: the proportion of tracks served which were discovery tracks. Higher numbers indicate more discoveries.
(3) Popularity: average popularity of tracks served to users. We consider normalized global popularity of tracks to compute this metric. Higher number indicates more popular track.
(4) Skip Rate: the proportion of tracks users skipped by users. Lower number indicate more user satisfaction.
(5) Similarity: this metric computes the average of similarity scores across all served tracks. Higher number indicates higher similarity, i.e. estimated relevance of track to the user.
(6) Familiarity: this metric estimates the average user-track familiarity of the served tracks. Higher numbers indicate more familiar tracks, with higher positive track affinity scores.

To avoid revealing sensitive metrics, we introduce a multiplicative factor to the base metrics reported.

5.2 Methods Compared

We compare a number of different recommendation policies, from single attribute scoring techniques to objective balancing methods.
Single attribute rankers:
A simple way of ranking tracks is to consider each attribute as a scoring criterion and rank the tracks based on a decreasing order of these scores. We consider the following five single score rankers:

1. **Similarity ranker:** we compute estimated relevance between user and track using the learnt user and track embeddings from the word2vec model (as described in Section 3.2.1), and compute cosine similarities score, and rank the candidate tracks based on a decreasing order of the similarity scores.
2. **Track Familiarity ranker:** this ranker considers user's affinity towards each candidate tracks, and orders tracks based on the estimated track affinity scores. Tracks which the users have historically streamed, and liked are ranked higher.
3. **Artist Familiarity ranker:** this ranker considers user's affinity towards the main artist of each candidate track, and orders tracks based on the estimated artist affinity scores. Tracks from artists which the users have historically streamed, and liked are ranked higher.
4. **Discovery ranker:** this ranker optimizes for discovery, and ranks discovery tracks higher than similar or familiar tracks. Specifically, we order tracks using the discovery scores, which are the track similarities scores, multiplied by a binary 0/1 indicator identifying discovery tracks.
5. **Rank score ranker:** this ranker sorts tracks based on the rank scores, computed using the rank obtained by the track when sorted by similarity scores.

Objective Balancing Rankers:
We compare the single attribute ranker by objective balancers, using the objective balancing techniques presented in Section 4. Specifically, we compare the following seven rankers:

1. **Weighted Sum rankers:** these rankers interpolate the scores using a weighted sum approach. We consider six variants of such rankers, comprised of two groups of three rankers each. We first consider a weighted sum between the similarity and track familiarity attribute, and pick three different weight profiles for these rankers. The first weight profile (i.e. Weighted Sum (S+F) - I) gives the lowest weight to similarity, while *Weighted Sum (S+F) - II* gives equal weight to similarity and familiarity. Finally, Weighted Sum (S+F) - III gives the most weight to similarity. Similarly, Weighted Sum (F + D) balance between familiarity and discovery attribute scores.
2. **OWA-SAT (AND):** we consider a single OWA function to score tracks, and use the following criterion scores as part of the OWA function inputs: (i) similarity score ($\zeta(u,x)$), (ii) track affinity score (t_x), (iii) artist affinity score (a_x), and (iv) rank score (γ_x). Specifically, each candidate track is scored by:
$$s_x = OWA(\zeta(u,x), t_x, a_x, \gamma_x) \tag{7}$$
The α parameter for this function is obtained via grid search, and to cater to the AND-ness of the OWA function, we select $\alpha > 1$.
3. **OWA-SAT (OR):** similar to above, we consider a single OWA function to score tracks in a single level OWA function $s_x = OWA(\zeta(u,x), t_x, a_x, \gamma_x)$ and in order to use OR-ness of the OWA function, we select $\alpha < 1$.
4. **OWA-SAT-Discovery (AND):** we additionally consider discovery attribute in the single level OWA function, and use the following criterion scores as part of the OWA function inputs: (i) similarity score ($\zeta(u,x)$), (ii) track affinity score (t_x), (iii) rank score (γ_x), and (iv) discovery score ($d_S(u,x)$). Specifically, each candidate track is scored by:
$$s_x = OWA(\zeta(u,x), t_x, \gamma_x, d_S(x)) \tag{8}$$
with α selected via grid search with the constraint $\alpha > 1$.
5. **OWA-SAT-Discovery (OR):** similar to above, we additionally consider discovery attribute in the single level OWA function: $s_x = OWA(\zeta(u,x), t_x, \gamma_x, d_S(x))$ and selectα via grid search with the constraint $\alpha < 1$.
6. **Hierarchical OWA (SAT):** for this ranker, we employ the 2-level OWA formulation of objective balancing, with the first level OWA consisting of similarity score, and artist familiarity, and the second level OWA with track familiarity and rank score. Specifically:
$$s_1 = OWA(\zeta(i,x), a_x) \quad \alpha = 0.3 \tag{9}$$
$$s_2 = OWA(s_1, \gamma_x, t_x) \quad \alpha = 3.0 \tag{10}$$
the specific configuration and parameters were selected using performance on a separate held-out validation set.
7. **Hierarchical OWA (SAT + Discovery):** for this ranker, we employ the 2-level OWA formulation of objective balancing and additionally consider the discovery attribute along with other satisfaction attributes. Specifically:
$$s_1 = OWA(\zeta(i,x), a_x, d_S) \quad \alpha = \alpha_1 \tag{11}$$
$$s_2 = OWA(s_1, t_x) \quad \alpha = \alpha_2 \tag{12}$$
with the specific configuration and parameters selected using performance on a separate held-out validation set.

5.3 Performance across approaches

Table 1 presents the detailed results across the 17 rankers compared across six different metrics. We first perform sanity checks, and observe that the single attribute rankers does indeed give best metrics for their corresponding attributes, with the similarity ranker obtaining the maximum similarity metric (0.848), and discovery and track familiarity ranker obtaining the maximum discovery (0.634) and familiarity metric (0.070) respectively.

Looking at user engagement metrics, we observe that the track familiarity ranker outperforms all other methods inn terms of satisfaction metric. This suggests and supports the hypothesis that users indeed like familiar music content. However, as expected, this ranker tanks the discovery metric severely. Further, it suffers from popularity bias, and achieves the maximum popularity metric across all methods compared. We further observe that both the track and familiarity rankers perform worst on discovery metric, which is expected since familiarity is conceptually opposite to discovery. However, we observe that artist familiarity ranker performs worse than track familiarity on discovery metric. Indeed, a familiar artist invalidates a large number of tracks from the discovery

Ranker	Satisfaction	Discovery	Popularity	Skip Rate	Similarity	Familiarity
Similarity ranker	0.7441	0.232	0.608	0.478	0.848	0.041
Track Familiarity ranker	0.7703	0.160	0.625	0.456	0.808	0.070
Artist Familiarity ranker	0.7542	0.150	0.611	0.467	0.811	0.044
Discovery ranker	0.7402	0.634	0.602	0.482	0.799	0.028
Rank score ranker	0.7422	0.316	0.601	0.487	0.822	0.033
Weighted Sum (S+F) - I	0.744	0.232	0.608	0.478	0.848	0.041
Weighted Sum (S+F) - II	0.754	0.213	0.612	0.470	0.843	0.066
Weighted Sum (S+F) - III	0.770	0.160	0.625	0.456	0.808	0.070
Weighted Sum (F+D) - I	0.740	0.634	0.602	0.482	0.799	0.028
Weighted Sum (F+D) - II	0.744	0.630	0.605	0.481	0.802	0.048
Weighted Sum (F+D) - III	0.770	0.160	0.625	0.456	0.808	0.070
OWA-SAT (AND)	0.7486	0.255	0.606	0.489	0.842	0.055
OWA-SAT (OR)	0.7521	0.273	0.606	0.477	0.833	0.059
OWA-SAT-Discovery (AND)	0.7378	0.568	0.603	0.489	0.822	0.046
OWA-SAT-Discovery (OR)	0.7384	0.620	0.602	0.491	0.807	0.046
Hierarchical OWA (SAT)	0.7626	0.193	0.616	0.464	0.832	0.069
Hierarchical OWA (S + D)	0.7588	0.359	0.613	0.470	0.827	0.069

Table 1: Offline evaluation of the different rankers on track re-ranking task.

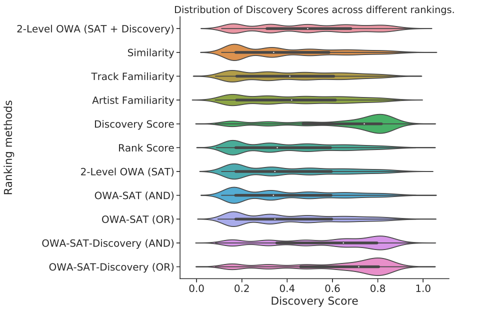

Table 2: Spread of discovery scores for the different rankers compared.

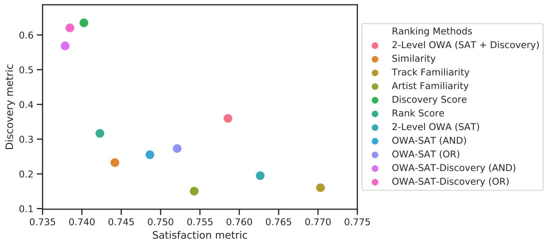

Figure 4: Scatter plot comparing different rankers on satisfaction and discovery metrics.

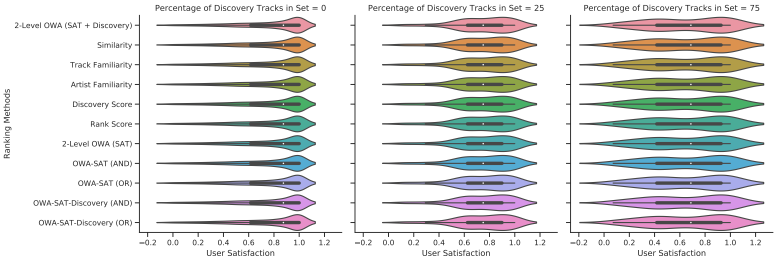

Figure 5: Spread of satisfaction metrics, for different levels of discovery tracks.

pool, whereas the impact track affinity has is only valid on one specific track. The weighted sum rankers give decent trade-offs, with slight satisfaction metric gains compared with similarity ranker, and slight discovery gains compared with familiarity rankers.

The OWA-SAT (OR) method gives almost as good a satisfaction metric win as the best performing weighted sum ranker, and while also giving a higher uplift in discovery metric. Finally, we observe that the 2-Level OWA functions both give a significant increase in satisfaction metric when compared with similarity ranker, with the 2-Level OWA (SAT + Discovery) giving the best trade-off: significant increase in satisfaction metric, and significant increase in discovery.

5.3.1 Spread of Discovery Scores. We next compare the spread of discovery scores across different rankers in Table 2. As expected, we observe that the discovery ranker has the most right-skewed distribution, which is expected since it is directly optimizing for discovery. However, we further observe that both the OWA-SAT-Discovery rankers have enough mass towards the right, indicating that a significant number of sessions were high discovery sessions for these rankers. Relating it to the above mentioned results, we note that such drastic shift towards the right of discovery distribution is often accompanied by steep drop in user satisfaction metrics, so these might not be viable solutions.

However, we observe that the 2-Level OWA (SAT + Discovery) presents a well distributed spread of discovery scores across different sessions, which decent number of session having high, medium and low discovery scores. This highlights that this model is able to identify sessions wherein boosting discoveries would give gains in discovery metric without severely hurting satisfaction metrics. Indeed, a key functionality which OWA functions bring to the table is their flexibility in assigning weights to different objectives. Only when there are tracks for which boosting the weight to discovery attribute make sense, does this model boost those weights.

5.3.2 Satisfaction across different levels of Discoveries. We next compare how sessions with varying extent of discoveries fare on user satisfaction. Figure 5 plots satisfaction metric on the x-axis and the different factor plots represent sessions with 0% discovery tracks, 25% discovery tracks and 75% discovery tracks. We observe that as we go from left to right, we see the mean satisfaction indicator reducing, and the spread of satisfaction metric increasing getting spread towards lower satisfaction values. Sessions with 0% discovery tracks have satisfaction scores skewed towards right, with a higher satisfaction mean. Further, interesting enough, not all sessions with 75% discovery tracks have low satisfaction. This indicates that there are certain sessions wherein high rates of discoveries are acceptable and satisfying. This hints at user and session level heterogeneities for discovery acceptability within sessions.

5.4 Trade-off Analysis

To better understand how different rankers situate themselves on the satisfaction-discovery trade-off, we plot a 2D scatter plot, comparing the satisfaction metric on x-axis and discovery metric on y-axis. Figure 4 presents the results, with circles towards the right having higher satisfaction, and circles towards the top having higher discoveries. First, we observe that there exist no ranker which is

	Autoplay				Radio			
Ranker	**Satisfaction**	**Skip Rate**	**Discovery**	**Return Rate**	**Satisfaction**	**Skip Rate**	**Unfamiliar**	**Return Rate**
Similarity ranker	-	-	-	-	-	-	-	-
Track Familiarity ranker	+13.6%	-11.3%	-13.9%	+7.9%	+5.3%	-5.95%	-8.04%	+2.14%
Artist Familiarity ranker	+3.5%	-2.23%	-38.6%	+2.02%	+2.6%	-2.20%	-10.22%	+0.86%
WeghtdSum Ranker	+5.1%	-1.54%	-25.32%	+2.44%	+0.79%$^+$	-0.41%	-7.47%	0.05%$^+$
OWA-SAT (AND)	-1.28%$^+$	+2.02%	+0.79%	-0.61%	-1.28%$^+$	+1.36%	+1.23%	-1.94%
Hierarchical OWA (SAT)	+3.99%	-4.8%	+1.43%	-5.48%	+1.38%$^+$	+1.36%	+1.01%	+0.1%
OWA-SAT-Discovery (AND)	-8.75%$^+$	+5.18%	+36.19%	+1.88%	-6.8%$^+$	+6.52%	+21.88%	-6.7%
Hierarchical OWA (SAT + Discovery)	+8.34%	-5.08%	+2.90%	+3.55%	+0.55%	+1.05%	+9.92%	+0.05$^+$

Table 3: Online AB test results: performance of rankers relative to similarity control. All results are statistically significant except for those marked by $^+$.

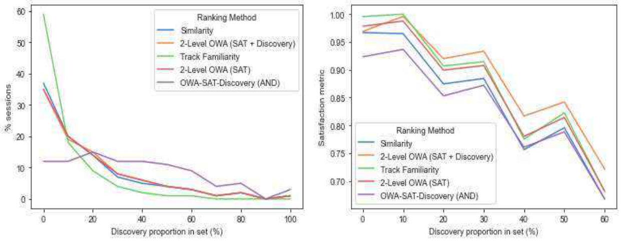

Figure 6: Left: Distribution of sessions with different amounts of discoveries across different rankers. Right: Normalized satisfaction metrics for different sessions with varying degrees of discoveries.

at the top-right, i.e., no ranker gives us higher satisfaction metrics while also giving higher discovery metric, highlighting the intricate trade-off that exists between the two goals. Further, the methods which maximize user satisfaction metric are much lower on the y-axis, indicating lower discovery metric performance. However, we can now visually confirm the findings we reported earlier that the Hierarchical OWA (SAT + Discovery) method is able to give us better trade-offs, by offering intermediate wins in satisfaction and at the same time, offering intermediate wins on discovery metrics.

6 LIVE AB TESTS

We next presents insights from the live AB tests we conducted with the most promising rankers based on offline evaluation.

6.1 Online Experiment Setup

We design and deploy four online randomized experiments, two each on Radio and Autoplay products:

Test 1: SAT only Experiment: First, we focus only on user satisfaction attributes, and deploy the following six rankers in production, on Radio and Autoplay products: (i) Similarity ranker (which acts as our Control), (ii) Track Familiarity ranker, (iii) Artist Familiarity ranker, (iv) Weighted Sum ranker (balancing familiarity and similarity with equal weightage), (v) OWA-SAT (AND) and (vi) Hierarchical OWA (SAT). We conduct a 7 day long randomized experiment wherein users are randomly assigned either to Control cell or one of the six treatment cells. Each cell was randomly allotted 4 million users, totaling 28 million users, and over 150 million user-track interactions.

Test 2: SAT + Discovery experiment: For the next online test, we additionally consider a discovery attribute, which required implementation of a discovery data provider that provided real time discovery score for each track, for both track and artist discovery. We deployed the following two additional rankers: (i) OWA-SAT-Discovery (AND), (ii) Hierarchical OWA (SAT+Discovery). We used similarity ranker as the control and conducted a 6 day long randomized experiment, across a total of 15 million users.

6.2 Online ranker performance

Table 3 presents the online metric results for the different rankers deployed in production. We observe a trend similar to the one from our offline evaluation experiments: familiarity centric rankers increase satisfaction while tanking discovery metrics. We also observe that the 7-day return rate increases for all rankers optimizing for familiarity in Autoplay. Importantly, we observe that the Hierarchical OWA (SAT + Discovery) is the only ranker which gives us a win-win across both metrics, compared to the similarity control. Indeed, multi-level OWA functions offer more control over balancing, and are less prescriptive about which attributes get what weights. Their flexibility in the weight assignment, which happens in real time upon observing the different scores for each track, allows them to make more informed, contextualized decisions of boosting discoveries when its useful.

6.3 Satisfaction ↔ levels of discoveries

Figure 6 (left) presents the extent to which different rankers served sessions with different proportions of discovery tracks. On one hand we observe that the track familiarity ranker had almost 60% sessions with 0 discovery tracks, while on the other hand, the OWA-SAT-Discovery ranker maintained a steady distribution of sessions across different discovery proportions. The best performing Hierarchical OWA (SAT + Discovery) ranker has a significantly higher number of sessions with 20% discoveries, thereby prioritizing traces of discoveries across most of the sessions. Building on top of these distributions, we next investigate how satisfaction metrics fare for sessions as we increase the amount of discovery tracks in them. Figure 6 (right) highlights that as we increase the amount of discoveries in a session, satisfaction metrics steadily decrease. However, interestingly most rankers either maintain or even slightly increase on satisfaction, when going from 0% discoveries to 10% discoveries. This promising result hints at the fact that users do expect some level of discoveries in their sessions, and 10% of discoveries is almost always preferred over no discoveries.

6.4 User Level Heterogeneity

We next present a user level view on the impact of injecting discoveries into sessions in Figure 7. The metrics we considered so far are aggregate metrics, and hence, not useful to investigate user level insights. Instead, for each user, we compute the average stream rate

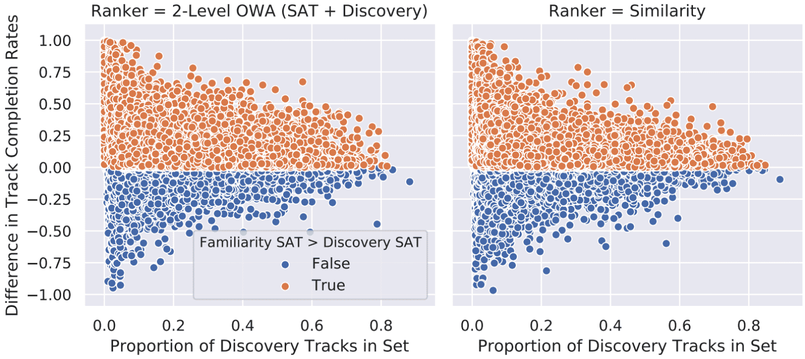

Figure 7: User level heterogeneity. Per user difference in track completion rates for discovery and non-discovery tracks.

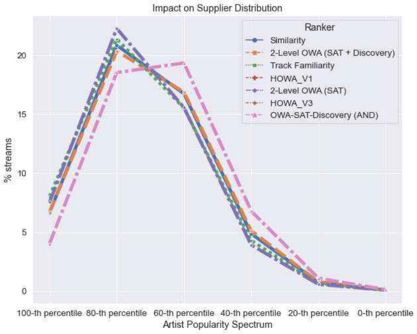

Figure 8: Impact on Suppliers: stream share across different popularity percentiles, across different rankers.

for discovery and non-discovery tracks, and plot this difference in track stream rates against the proportion of discovery tracks in those sessions. Items above 0 (marked as orange) indicate user sessions which preferred non-discovery tracks over discovery tracks, and hence had a higher track stream rate for non-discovery tracks.

We observe a strong user/session level heterogeneity, and find a large number of sessions where discovery tracks had a higher track stream rate than non-discovery tracks, across different levels of discovery proportions in sessions, and across different rankers. These results further establish the user/session level characteristics of discovery receptivity of users, and provides empirical support to prior work highlighting the presence of strong user-level receptivity to specific objectives [18]. Further, we observe that the differences converge to smaller values as session length increases, which indicates that longer sessions are either generally more successful, or that some of these user sessions are laid back sessions, wherein the user is not in an interactive enough state to skip tracks. Further investigation is needed to understand this in detail.

6.5 Impact on Suppliers

We hypothesized that discovery can act as an enabler of shifting consumption towards less popular artists (Section 3.3). We investigate to what extent this is true. In Figure 8, we consider a random sample of streamed content, and plot the stream share that went to artists of different popularity buckets. We observe that models which over-emphasize on discovery, are able to significantly shift the consumption towards right, i.e. transfer streams to less popular artists, as is evident by the right-shifted distribution of methods like OWA-SAT-Discovery (AND). Even rankers which provide a healthy balance between satisfaction gains and discovery gains are able to shift stream share towards less popular artists, and decrease stream share for more popular artists.

Indeed, optimizing for discovery enables platforms to control consumption patterns, and divert consumption towards less popular or niche artists, who might otherwise not get exposed enough. Such departure from relevant, popular and familiar content allow platforms to broaden the scope of music listening and shift consumption towards the tail and less familiar content.

7 CONCLUSION

Looking at music consumption data, and the presented results, we found evidence that beyond relevance, familiarity plays a key role in driving user engagement on music streaming platforms. However, we additionally highlight that blindly optimizing for familiarity results in potential long term harms, which are detrimental not only for user, but also for artists and the overall health of the streaming platform. Our findings demonstrate the need for efficient balancing of relevance, familiarity and discovery objectives when serving recommendations to users, and demonstrates that the proposed objective balancing methods are able to obtain wins on both aspects of user satisfaction and discovery metrics, in offline as well as online live experiments.

Specifically in the context of music streaming, we posit our findings relates to and builds upon insights on how users consume music. Our work underlines the importance of explicitly considering and optimizing for familiarity and discovery, and underpins the improvements in user engagements that are on offer when rightly done. Further, we highlight that music streaming applications are essentially multi-stakeholder platforms which connect users and artists. Such platforms need to maintain a healthy balance between user satisfaction and artist exposure goals [17, 19]. We advocate for discovery as another enabling tool which equips system developers to influence and control consumption patterns on the platform, and shift them towards less popular, and potentially niche artists.

On the system design perspective, our findings give system designers practical approaches which are easy to extend and deploy in large scale industrial setting. We contend that the proposed framework is generic enough to work with more sophisticated ML models for various predicted attributes.

Finally, our findings motivate future work on (i) quantifying user propensities for discovery, (ii) leveraging discovery for targeted supplier exposure optimization, and (iii) identifying good discovery candidates by developing personalized discovery models.

ACKNOWLEDGMENTS

I am grateful to various teams in the Mixer product area, specifically the Radio team, including Maddie Kirwin and Harpreet Singh for their help in experimentation, and to Edward Lee for the insightful discussions on impact of balancing on users.

REFERENCES

[1] Eytan Adar, Jaime Teevan, and Susan T Dumais. [n.d.]. Large scale analysis of web revisitation patterns. In *CHI 2008*.

[2] D Agarwal, B-C Chen, P Elango, and X Wang. 2012. Personalized click shaping through lagrangian duality for online recommendation. In *SIGIR*.

[3] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. [n.d.]. The dynamics of repeat consumption. In *WWW 2014*.

[4] Steven Caldwell Brown and Amanda Krause. 2016. A psychological approach to understanding the varied functions that different music formats serve. In *14th Biennial International Conference on Music Perception and Cognition*.

[5] Jean Garcia-Gathright, Brian St. Thomas, Christine Hosey, Zahra Nazari, and Fernando Diaz. 2018. Understanding and evaluating user satisfaction with music discovery. In *SIGIR*.

[6] Rupesh Gupta, Guanfeng Liang, Ravi Kiran Tseng, Xiaoyu Chen, and Romer Rosales. 2016. Email volume optimization at LinkedIn. In *KDD*.

[7] J L Herlocker, J A Konstan, L G Terveen, and J T Riedl. [n.d.]. Evaluating collaborative filtering recommender systems. *ACM TOIS 2004* ([n. d.]).

[8] Neil Hurley and Mi Zhang. [n.d.]. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM TOIT 2011* ([n. d.]).

[9] D Jannach, L Lerche, and M Jugovac. 2015. Item Familiarity Effects in User-Centric Evaluations of Recommender Systems.. In *RecSys Posters*.

[10] B Kahnx, R Ratner, and D Kahneman. [n.d.]. Patterns of hedonic consumption over time. *Marketing Letters 1997* ([n. d.]).

[11] Marius Kaminskas and Derek Bridge. [n.d.]. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM TiiS 2016* ([n. d.]).

[12] Sherrie YX Komiak and Izak Benbasat. 2006. The effects of personalization and familiarity on trust and adoption of recommendation agents. *MIS quarterly* (2006).

[13] Audrey Laplante and J Stephen Downie. [n.d.]. The utilitarian and hedonic outcomes of music information-seeking in everyday life. *Library & Information Science Research 2011* ([n. d.]).

[14] Jin Ha Lee, Hyerim Cho, and Yea-Seul Kim. 2016. Users' music information needs and behaviors: Design implications for music information retrieval systems. *Journal of the association for information science and technology* 67, 6 (2016).

[15] Xinwang Liu and Qingli Da. [n.d.]. On the properties of regular increasing monotone (RIM) quantifiers with maximum entropy. *International Journal of General Systems 2008* ([n. d.]).

[16] Matti Mäntymäki and AKM Islam. 2015. Gratifications from using freemium music streaming services: Differences between basic and premium users. (2015).

[17] R Mehrotra, J McInerney, H Bouchard, M Lalmas, and F Diaz. [n.d.]. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM 2018*.

[18] Rishabh Mehrotra, Chirag Shah, and Benjamin Carterette. 2020. Investigating listeners' responses to divergent recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 692–696.

[19] 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.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

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

[22] Rebecca K Ratner, Barbara E Kahn, and Daniel Kahneman. 1999. Choosing less-preferred experiences for the sake of variety. *Journal of consumer research* (1999).

[23] Sarah K Tyler and Jaime Teevan. [n.d.]. Large scale query log analysis of re-finding. In *WSDM 2010*.

[24] Ronald R Yager. [n.d.]. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics 1998* ([n. d.]).

[25] X Yi, L Hong, E Zhong, Nanthan N Liu, and S Rajan. [n.d.]. Beyond clicks: dwell time for personalization. In *RecSys 2014*.