Exploring Exploration in Recommender Systems: Where? How Much? For Whom?

Mitchell J.C. Dingjan

TU Delft

muziekweb

(This page has been intentionally left blank.)

Exploring Exploration in Recommender Systems: Where? How Much? For Whom?

by

Mitchell J.C. Dingjan

to obtain the degree of Master of Science

in Computer Science

at the Delft University of Technology,

to be defended publicly on Monday March 9, 2020 at 11:00 AM.

Program track:	Data Science & Technology	
Project duration:	July 8, 2019 – March 9, 2020	
Thesis committee:	Prof. dr. A. Hanjalic,	TU Delft, chair
	Dr. ir. C.C.S. Liem,	TU Delft, supervisor
	Dr. N. Tintarev,	TU Delft

This thesis is confidential and cannot be made public until March 9, 2020.

An electronic version of this thesis is available at https://repository.tudelft.nl/.

(This page has been intentionally left blank.)

Abstract

Recommender systems focus on automatically surfacing suitable items for users from digital collections that are too large for the user to oversee themselves. A considerable body of work exists on surfacing items that match what a user liked in the past; this way, the recommender system will exploit its knowledge of a user's comfort zone. However, application scenarios exist in which it is explicitly important to offer the user opportunities to explore items beyond their existing comfort zones. In such cases, the recommender should include items for which there is less existing evidence that the user will like them. This calls for the recommender to explore to what extent a user would be tolerant to exploration.

In this thesis, we consider that different users will likely have different preferences with respect to item exploration. We propose personalized item filtering techniques for this, modeled under a multi-armed bandit framework, that consider (1) how and where exploration vs. exploitation items should be distributed in a result overview and (2) how adventurous exploration items can be, considering a user's general willingness to explore and existing familiarity with item categories in the collection. We present the results of a survey and an online quantitative experiment with 43 users of Muziekweb, a public music library that encourages taste broadening in the Netherlands, demonstrating the effectiveness of both our proposed filtering techniques that aim for personalized exploration.

Keywords
Recommender systems; Exploration; Personalization; Filtering; User preferences; Taste broadening; Multi-armed bandits

(This page has been intentionally left blank.)

Preface

This report concludes my thesis project, the last compulsory course needed to obtain the Master's degree in Computer Science at the Delft University of Technology. This project, done in the course of nine months, was carried out with the help of Muziekweb, the biggest source of music that has been released in the Netherlands.

I would first like to express my gratitude to my supervisor, Dr. ir. Cynthia Liem, for her professional guidance, encouragement to think outside of the box, and continuous support throughout the project. Also, I would like to thank the people at Muziekweb, especially Dr. Ingmar Vroomen and Casper Karreman, for their enthusiasm and support. Furthermore, I would like to thank Prof. dr. Alan Hanjalic and Dr. Nava Tintarev for being part of my thesis committee and their reviews. I would like to thank the staff and students present at the student meetings of the Multimedia Computing Group for their feedback on my presentations as well.

Additionally, in special, I would like to thank two of my best friends, Maarten Sijm and Mathias Meuleman, for being there from our very first day of university. We have learned a lot together over the last five years and I hope we will continue to do so for a very long time.

Last but not least, I would like to express my gratitude to my family, friends, and girlfriend. My deepest thanks go out to my parents and sister for always encouraging and supporting me; you deserve my eternal gratitude.

Mitchell J.C. Dingjan
Delft, March 2020

(This page has been intentionally left blank.)

Contents

List of Algorithms

(This page has been intentionally left blank.)

List of Figures

List of Tables

(This page has been intentionally left blank.)

Nomenclature

Acronyms

BPR Bayesian Personalized Ranking. 10

CDB Counterfactual Dueling Bandits. 11

CDR Centrale Discotheek Rotterdam. 6

DOM Document Object Model. 35, 61

DSP Document Space Projection. 11

EA Evolutionary Algorithm. 5

Exp3 Exponential-weight algorithm for Exploration and Exploitation. 16, 44

Exp4 Exponential-weight algorithm for Exploration and Exploitation using Expert advice. 17, 44

FP Familiarity Profile. 22, 23, 30, 32, 34, 36, 41, 43–45

HREC Human Research Ethics Committee. 7, 37, 75

iid independent and identically distributed. 14

IR Information Retrieval. 1, 2, 44

JS JavaScript. 61

MAB Multi-Armed Bandit. 2, 5–7, 9, 13, 30, 31, 44

ML Machine Learning. xvi, 13

npm Node package manager. 61

OL2R online learning to rank. 11

PDF Probability Density Function. 21

RL Reinforcement Learning. 2, 3, 5, 13, 19, 21

SD Standard Deviation. 39

SIGIR Special Interest Group on Information Retrieval. 5, 7, 47

TS Thompson Sampling. 18, 44

UCB Upper Confidence Bound. 17, 44

Jargon

Diversity The internal difference within parts of an experience. 1, 2

Exploitation The probing of items that users like with high certainty through recommendation, based on the user's derived preferences. To us, for horizon broadening, exploitation consists of recommending items that are inside the user's comfort zone. 1, 2

Exploration The probing of items that users like with relatively low certainty through recommendation, to gain information on the user's preferences. To us, for horizon broadening, exploration consists of recommending items that are outside the user's comfort zone. 1, 2

Exploration–exploitation paradigm A reinforcement learning idea that aims for balancing content that the user likes with high certainty (exploitation) versus content with less certainty (exploration). 1, 2

Explore–exploit item placement strategies Strategies to merge exploration and exploitation item candidates deduced by the recommender system in a single result overview such that the explore–exploit item placement suits the user. 20

Filter bubble A self-reinforcing pattern of narrowing exposure that reduces user creativity, learning, and connection. 2, 5

Novelty The difference between present and past experience. 1, 2

Personal value Something that is important for a person in life at the most abstract level. 9

Personality A combination of characteristics or qualities that form an individual's style of thinking, feeling and behaving in different situations. 9

Recommender system Software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. 1

Reinforcement Learning (RL) The branch of Machine Learning (ML) that learns a mapping from situations to actions maximizing a numerical reward. xv, 2

Taste broadening The result of the act to broaden/expand someone's taste which involves trying items that the person is yet unfamiliar with. 1

1

Introduction

In many digital multimedia consumption scenarios, the size of digital collections has grown so large that automatic filtering methods have become indispensable to users. In particular, personalized recommendations are considered to be good means to get the right content to the right users. Many commercial recommender applications may most obviously benefit from getting users the items they explicitly want, in a way that is as frictionless as possible. This criterion is reflected in common success metrics, such as engagement and click-through rates. At the same time, other scenarios exist in which a more serendipitous perspective is important, and the recommender system should assist users in broadening their horizons and making them discover items they did not yet know they liked. Such a perspective is increasingly becoming important to the positioning of public libraries. While online search engines may offer easy at-home access to a broader body of information, libraries can offer users trusted, curated, and locally approachable ways to also engage with more specialized content in the long tail.

One domain in which discovering new content (and possibly, new tastes and interests) is natural, is music. However, a system that solely would provide items outside of a user's known comfort zone will likely hamper the user experience. Therefore, proposals have been presented to balance recommendations that exploit known user preferences with recommendations that explore what acceptable recommendations may exist beyond these known preferences [42, 54]. The exploration-exploitation paradigm has more broadly been proposed as a means to increase utility beyond relevance, both in the recommender systems and Information Retrieval (IR) communities [9, 36, 82].

In this thesis, we consider that different users will likely have different preferences with respect to item exploration. Therefore, we *explore exploration.* That is, we try to include the user through item exploration preferences in the algorithmic exploration process to personalize it. Also, we try to capture the preferences through an interactive user interface. Our aims are to deal with the questions of *"Where?"*, *"How much?"*, and *"For whom?"* to let the recommender explore. To this end, we will propose methodology that aims for user-aware exploration by the recommender. The answers to these questions will benefit any recommendation type – especially taste broadening recommendation. This chapter introduces our research in detail.

1.1. A High-Level Introduction to the Problem

Recommendation is in mankind's nature; each conversation can lead to a recommendation. For example, if you meet someone for the first time, during the conversation, you typically try to gather information about the person. Based on what the person tells you, if relevant, you can do a recommendation. You could derive familiarity and relevancy of subjects, as well as of any of their categories. Besides, to do so, you could combine this information with the person's appearance and body language. Conversations with someone you have met before, are based on the same principle, but during these you can use information that you have gathered during earlier conversations. Also, for the sake of completeness, experiences obtained during conversations with other people can influence the recommendations you do.

A method to improve the recommendations you do in terms of diversity and novelty, is to introduce items, during the conversation, of which you have less evidence that your conversation partner will like them. As a result, even if the person does not like the recommended item, you learn something. Where, how much, for whom, and when to apply this technique is not always trivial; you apply it based on your gut feeling. 'Where'

here refers to the question that, if you are recommending some set of items, where do you introduce an item that the conversation partner will like with less certainty in your enumeration? If you rely on the method too much, your recommendations may become unreliable, if not irrelevant. To determine the way in which you want to apply this technique, you could *inter alia* take into account the person's familiarity on the subject. For example, imagine that you want to recommend some new movies to a movie expert; indeed, this is difficult. For the expert you probably need to come up with a smarter way of applying this technique, than in the case of someone with less expertise in the movie domain.

Because of the utility of recommendations and the rise of automation, recommender systems have been introduced, motivated by the fact that they can be very powerful if fed with lots of data. Of course, building the 'perfect' recommender system is at least very difficult, if not impossible. The most important characteristics of recommendations through the 'old-fashioned' way of conversations are still applicable for automated recommendation. During sessions with the user, recommender systems are programmed to try to gather as much as possible user preference data to optimize their recommendations for the user on request. Gathering information from verbal communication, body language, and appearance has been replaced by activity tracking during sessions. Meeting a person for the first time, is for recommender systems referred to as the 'cold start' problem. Whereas you can use, for example, the person's appearance and experience from previous conversations, a recommender system also requires some tricks to build an information foundation. Solving the problem is, however, not trivial; it is tightly connected to the context of the recommender system.

Similarly, the technique of introducing items of which you are less certain that the conversation partner will like them can also be applied by recommenders. Within the recommender system and IR domains, it is actually a well-known and important method termed 'exploration', as mentioned at the start of this chapter. Since where, how much, for whom, and when to apply this technique is not trivial within conversations, as you apply it based on your gut feeling, this is at least as difficult for recommenders.

1.2. The Exploration-Exploitation Paradigm

Nowadays, it no longer comes as a surprise that the use of recommender systems cannot only be judged based on their accuracy. In modern recommendation and IR, diversity and novelty have made their entrance for assessing usefulness and are becoming increasingly important [20, 33, 41, 44, 69, 83]. The difference between these two properties is subtle. Whereas diversity is the internal difference within parts of an experience, novelty is the difference between present and past experience [69]. In IR, diversity is the need to resolve ambiguity and novelty is the need to avoid redundancy [20]. A lack of diversity and novelty could result in a filter bubble, which is caused by too much content personalization [61, 63]. Aside from preventing the system to overfit, together these concepts enrich user experience, which can help to broaden the user's horizon. However, modelling them is not trivial, as different users perceive diversity and novelty differently. Thus, this problem heavily depends on human-oriented involvement.

A well-established concept in recommender systems and IR that allows for increasing the use of systems beyond relevance is the exploration-exploitation paradigm [9, 36, 54, 82]. This Reinforcement Learning (RL) idea aims for balancing content that the user likes with high certainty (exploitation) versus content with less certainty (exploration). Whereas exploration focuses on recommendation diversification, exploitation leads to recommendation intensification. Without exploration, users will miss out on diverse and novel content. On the other hand, without exploitation, users will miss out on relevant content. This balancing dilemma is known as the exploration-exploitation trade-off [77].

In this thesis, we focus on horizon broadening recommendation, while guarding accuracy. Therefore, we work from the exploration-exploitation paradigm. From this perspective, we interpret exploration to be recommendation of items outside the user's comfort zone. Similarly, we interpret exploitation to be recommendation of items inside the user's comfort zone. These interpretations will help us to set up our methods. Moreover, we work from Multi-Armed Bandits (MABs), as these are a classical formalism for studying the explore-exploit dilemma [77]. We elaborate on this in Section 1.5. Since exploration focuses on diverse and novel content, exploring exploration will be worth it to improve the overall use of recommender systems.

1.3. Diving a Little Deeper

We explore exploration to make recommendation more user-aware. More specifically, we deal with the questions of where, how much, and for whom to let the recommender explore. Let's dive a little deeper. In any recommendation setting, we deal with a problem in which we have to generate a recommendation item result overview (e.g., a list) of size $k \in \mathbb{N}$. The usual procedure is to first select a candidate pool, from which items are chosen. As explained in the previous section, we work from the exploration-exploitation paradigm. To illustrate, using a rather basic RL random exploration algorithm termed ϵ-greedy, the following procedure is applied: given a fixed $\epsilon \in [0, 1]$, generate some value p between 0 and 1 randomly. If $p > \epsilon$, exploit by selecting the best candidate from the candidate pool. Else, explore by choosing a random item from the pool (or a totally random item). RL methods are all about finding a balance between exploitation and exploration to get to an optimal solution.

In practice and literature, this ϵ is usually considered to be *fixed* over the result overview. In other words, one could state that a uniform probability distribution is assumed over the result overview. In this thesis, we research whether we can tweak the probability distribution to make it suitable to the user. This means that we aim for personalizing the explore-exploit item placement of the output. For a list, for example, would it be preferable to explore at the start, the end, or somewhere in between? Another example is a mosaic. We extend this to any human-interpretable result overview. This preference could depend on the setting of the problem. For instance, it could differ for different types of items (e.g., a song versus a movie). Figure 1.1 illustrates this, where two users prefer different exploration values for the third position in a list.

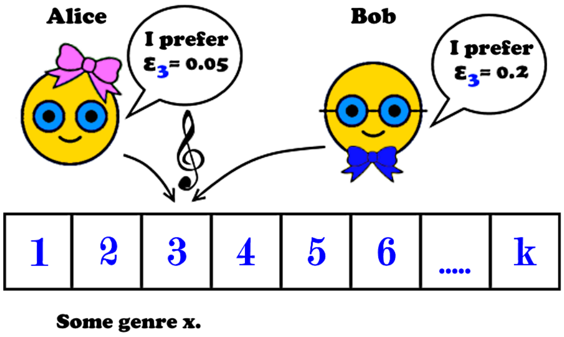
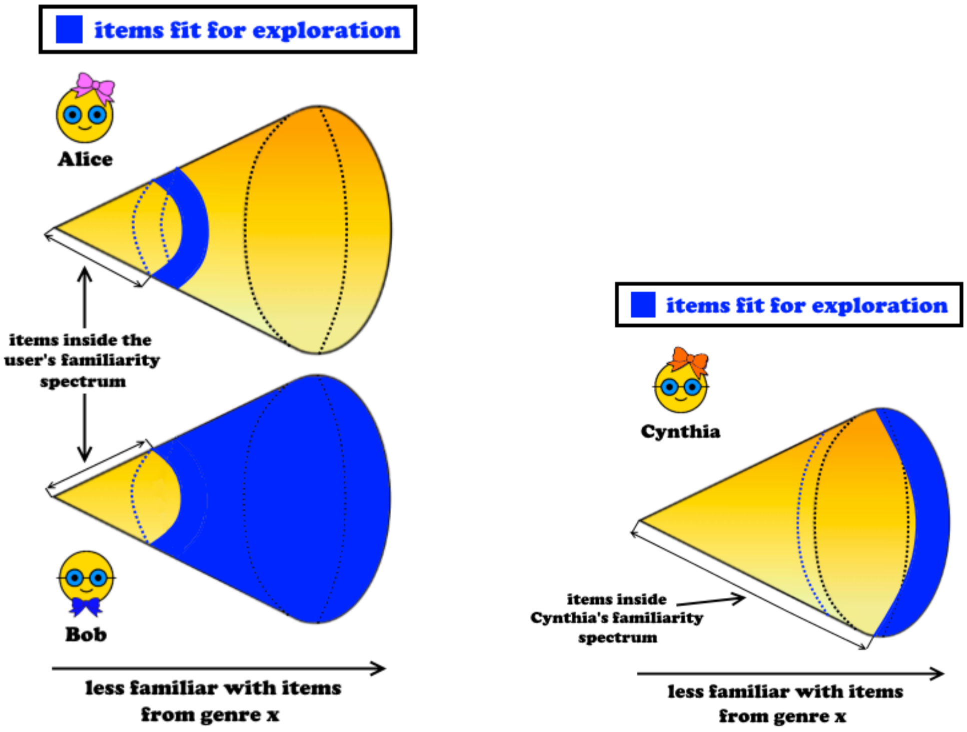


Figure 1.1: Different Exploration Preferences of Two Users for a List of Songs From Genre x. In Reality, Users Are Not This Direct!

Moreover, the fixed parameter ϵ does not allow for differentiation between users. Consider Figure 1.1 and imagine that Bob is a very adventurous person, who tries everything new that appears on his path, while Alice is a rather conservative person, who rarely tries something new. In other words, the two users prefer different exploration rates. This implies for Alice that the ϵ_i values that she would assign $\forall i$, are on average lower than the values Bob would assign. Therefore, aside from the preferred item placement, by integrating the preferred exploration rate, we can tweak the probability distribution further to optimize the fit.

In the context of taste broadening recommendation, we assume that a digital collection can be partitioned into different categories (e.g., musical genres). Within a category, we consider two personalization opportunities related to exploration: (1) a user's natural willingness towards exploration, as well as (2) the user's familiarity with the category. The different characters of Alice and Bob illustrate the need to consider their exploration willingness. Although Bob would be able to pick the most random exploration items that are presented to him, Alice would only carefully pick the ones that are closest to her familiarity spectrum. We consider that for a given user, items within a particular category can be ranked on the user's familiarity with them, as expressed in the form of a familiarity score. Multiple items may be equally (un)familiar to a user; generally, the further away we will move out of a user's existing consumption profile, the more items are expected to be equally (un)familiar. Alice and Bob are familiar with their own 'tip of the iceberg' of genre x. Imagine that this iceberg is conelike shaped. If we would merge the cones of all genres, where each cone has some overlap with another cone, we would end up with a spherelike object that represents the entire music domain. Such a cone is finite in volume, as there are a limited number of items that belong to a genre. The cones of Alice and Bob are visualized in Figure 1.2a, which shows their different exploration willingness levels.

Differences in category familiarity between users call for adjusting the way in which the recommender explores as well. Imagine that Alice and Bob have a friend called Cynthia. Cynthia is a music expert, specialized in genre x. Also, she is very eager to learn everything there is (left) to learn about genre x. Thus, when compared to Alice and Bob, Cynthia already knows a lot more about items from genre x. As a consequence, the border of her familiarity spectrum for genre x is further away. The cone of Cynthia is visualized in Figure 1.2b, which shows a different familiarity level of genre x. Therefore, from the cones of Figure 1.2, by integrating user preferences with respect to exploration willingness and category familiarity in exploration, we can


🟦 items fit for exploration
Alice
items inside the user's familiarity spectrum
Bob
less familiar with items from genre x
🟦 items fit for exploration
Cynthia
items inside Cynthia's familiarity spectrum
less familiar with items from genre x

(a) Different Exploration Willingness Levels of Two Users.

(b) Different Familiarity Level of Genre x, When Compared to the Users from Figure 1.2a.

Figure 1.2: In the Context of Taste Broadening, Different Exploration Willingness and Genre Familiarity Levels Between Users.

tweak the way in which we explore to optimize the fit. Note that over time, when interacting with our imaginary adapted taste broadening recommender system, the borders of the familiarity spectra of Alice and Bob grow towards Cynthia's current border, and eventually further, in a way that is comfortable to each of them.

1.4. Motivation

The research conducted in this thesis is motivated by a user-centered and a research motive. Let's begin with the former motive. We stated in the previous section that we explore exploration from a user perspective to enhance user awareness in exploration. The research is shaped by the questions of where, how much, and for whom to let the recommender explore. Therefore, our research focuses on improving the user experience of exploration in recommendation. This holds for any recommendation context, especially for taste broadening recommendation, since exploration is fundamental in this context. To further explain this motive, based on the previous section, it can be divided into two submotives.

Firstly, our research aims for an explore-exploit item placement in recommendation result overviews (e.g., lists and mosaics) that suits the user. It leads to result overviews where users better like the item placement of the recommended exploitation items. Likewise, it leads to result overviews where users feel more comfortable with the item placement of the recommended exploration items. The latter result could also yield a better horizon broadening process. If users are in the mood to try something new, they could be more tempted to explore when exploration items are at comfortable positions. Moreover, the placement strategy takes into account a suitable exploration rate, so that neither too few nor too many exploration items are included.

Secondly, in the context of taste broadening recommendation, aside from a suitable explore-exploit item result overview, our research aims for a suitable exploration approach for the user. It results in an approach that takes into account the item category familiarity of the user. The familiarity of different users differs for the relevant recommendation domain (e.g., movies and music) of a recommender system. This means that some appropriate way of exploration for one user may not be equally appropriate for another. For instance, a music expert is familiar with a relatively wider range of music genres and, for at least most genres, familiar with a relatively larger set of songs than an average human. Therefore, when a music expert is in the mood to broaden his/her taste, a different way of exploration is preferred. Moreover, the exploration approach takes into account the user's exploration willingness, so that neither too random nor too familiar content is explored. Based on the user-aware explore-exploit item placement and exploration approach, we see that the main motive helps to tackle the filter bubble. This is a self-reinforcing pattern of narrowing exposure that reduces user creativity, learning, and connection [61, 63].

Furthermore, the research motive can also be divided in two submotives. Firstly, our research contributes to the taste broadening recommendation research field, which is relatively in its infancy [49]. Secondly, our research contributes to the research field that studies the integration of explore-exploit strategies in ranking approaches, which is almost an empty field [32]. In other words, this thesis enters relatively little explored research fields, which positively affects the value of its contribution. While this is true, the fields are very much *relevant* for the recommender system and IR research domains. To illustrate relevancy, this research covers several main topics that are of interest to CHIIR, RecSys, and Special Interest Group on Information Retrieval (SIGIR)[1], based on their "Call for Contributions" pages over the last three years [2–4]. In the case of RecSys, for example, hot topics like "Personalization" and "Novel machine learning approaches to recommendation algorithms (e.g., RL)" are two important ones that our research covers.

1.5. Research Goal and Scope

The main research goal of this thesis is to explore exploration to deal with the questions of "Where?", "How much?", and "For whom?" to let the recommender explore. Section 1.3 explained how and Section 1.4 explained why we want to achieve this. In short, we are convinced that users should have a say in how to let the recommender explore. We study the user preference integration of explore-exploit item placement, exploration, and familiarity to answer the questions of where, how much, and for whom to explore, respectively. To do so, we aim to propose methodology that makes exploration more user-aware and design an interactive user interface. We then aim to evaluate the methodology's effectiveness by means of a quantitative user study.

The observant reader might have detected that Section 1.1 poses one additional question, aside from the questions that we study. This question is about "When?" to let the recommender explore, which covers the temporal aspect of the problem. In our research, one could change the user-aware explore-exploit item placement and exploration approach over time. For instance, during the first sessions with a new user, the recommender system could emphasize exploration and decrease it over time. This could be interesting, since it could help to deal with the cold start problem. On the other hand, one could argue that, to some extent, the temporal aspect could be taken into account by recommender system updates of the user preferences over time. However, we will not evaluate this aspect in the thesis experiments or further investigate it, as it adds another complexity layer. In this thesis we aim to lay the foundations for the research fields covered in the previous section. Therefore, this aspect will be considered to be out of scope and future work.

Moreover, for the thesis experiments we choose to work from MAB-based recommender systems. However, this can be considered to be merely an implementation detail. In fact, we could have chosen any other kind of recommender system. MABs are closely related to RL. One of the main challenges that arises in RL, and not in other kinds of learning, is the exploration-exploitation trade-off [77]. The main reason for implementing a MAB-based recommender system, is the fact that MAB serves as a natural framework for the recommendation and personalization of content [28, 76]. When compared to RL problems, we only need one state to do so. Yet another approach could be to apply Evolutionary Algorithms (EAs), from the evolutionary computation domain. However, the fundamental concepts of exploration and exploitation are not that well understood and developed for the EA domain. Besides, it suffers from some important internal weaknesses like premature convergence and low computational efficiency – although research is done on this and improvements are proposed [37, 84]. Also, MAB has been studied for almost a century and picked up by a large research community, meaning it is interesting and we relatively know a lot about the theory by now [46, 76]. A more precise description of the relation between MABs and RL is provided later, as described in Section 1.8.

1.6. Research Questions

We research exploration by the recommender system from a user perspective to make it more user-aware. More specifically, we study the user preference integration of explore-exploit item placement, exploration, and familiarity to answer the questions of "Where?", "How much?", and "For whom?" to let the recommender explore, respectively. Based on this, we define the following research questions, which we refine as we progress through the chapters.

[1] CHIIR, RecSys, and SIGIR are conferences that are yearly organized by ACM, which are respectively on: computer-human information interaction and retrieval, recommender systems, and information retrieval.

RQ1 How much do users prefer our recommendations that integrate user preferences with respect to:

- explore-exploit item placement;
- exploration; and
- familiarity?

RQ2 How do the preferences of **RQ1** individually contribute?

RQ3 Where do users draw the line between items fit and unfit for broadening their horizon from our recommendations?

To help answer the research questions and structure the thesis, we define the following subquestions:

Q1 What state-of-the-art methods have already been proposed to integrate user preferences in recommender systems and how do they work?

Q2 Based on the state-of-the-art literature about MAB-based recommender systems, how does exploration work with MABs?

Q3 How do we integrate user preferences with respect to explore-exploit item placement, exploration, and familiarity in exploration?

Q4 How do we design and set up our experiments to answer our research questions?

Q5 How can the effectiveness of our proposed methodology be evaluated?

To each of these subquestions we dedicate a chapter, which further breaks down its question and answers it.

1.7. Muziekweb

At the start of this chapter, we sketched how the assisting ability of recommenders in broadening the users their horizons and making them discover items they did not yet know they liked, becomes increasingly important to the positioning of public libraries. Muziekweb is aware of this perspective and takes a leading role in encouraging music taste broadening in the Netherlands. The public music library is part of the Centrale Discotheek Rotterdam (CDR) and has since 1961 built a collection of 600,000 CDs, 300,000 LPs, and 30,000 music DVDs [56]. Items from its collection are catalogued and can be borrowed both physically at the library and virtually through its website. Its website is a source of information about music that has been released in the Netherlands during the past fifty years. This digital information has been gathered by processing the music collection. Muziekweb's primary mission, since 1961, is "to make it possible to expand taste and musical development for everyone in the Netherlands" [57]. Indeed, its mission perfectly fits the motivation of this research (see Section 1.4). Therefore, we decided to team up for the thesis experiments. The nature of this cooperation is strictly scientific.

Over the years, Muziekweb shifted the focus of its main task from lending out LPs to maintaining and developing its digital services. Nowadays, Muziekweb offers three main 'Explore' features on its website to assist users in broadening their tastes. Each feature has been composed by music experts of Muziekweb. Supplementary to the following explanation, Appendix B contains impressions of the features. With Muziekweb Intro's, users can make 'exploration journeys' to 100 different music genres [59]. Each of these journeys introduces the user to a specific music genre through a slide show containing plenty of relevant songs. Besides, with Muziekweb Playlist, users can broaden their music tastes around specific themes through composed playlists, like travel or feel good music [60]. Such playlists are added on a regular basis.

Moreover, Muziekweb's third Explore feature is automated music advice generation [58]. With this feature, users can request artist advice by providing several artist names, or music album advice by proving several album names. The latter recommendation setting is most interesting for our thesis experiments, as it involves music recommendation in a taste broadening setting. Figure 1.3 provides an impression of Muziekweb's albums advice generation page. The recommender system of this page accepts at least three input albums and, based on these, generates an output advice, which is presented as a 2-dimensional album mosaic. The user can use a slider to indicate the preferred popularity of the albums. This is an interesting feature for our research, as it is an explicit way for users to indicate their preference with respect to music album exploration. The recommender itself is not yet mature. The recommendations are determined using a greedy algorithm

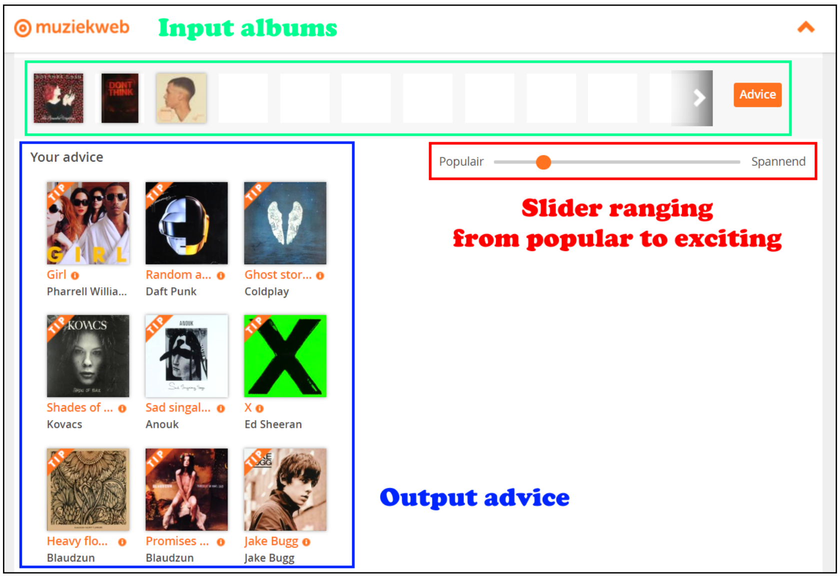


Figure 1.3: Muziekweb's Album Advice Generation Page.

that works with the album borrowing data and takes into account the album popularity preference. We will use the ideas of the slider and the 2-dimensional mosaic as inspiration for our experiments.

To support the experiments, Muziekweb provided a sufficient part of its data, an unlimited number of API requests, and special API access. The provided data was allowed to be used for the developed recommender system and, to a certain level, to be displayed during the experiments. Furthermore, Muziekweb promoted the research through its homepage and social media to encourage its users to participate in the experiments. Also, the music genre experts working at Muziekweb offered their knowledge and participation.

1.8. Thesis Outline

To complete the research introduction, this section provides an outline of the rest of the thesis report. The following enumeration defines the goal of each upcoming chapter:

- Chapter 2 presents work related to integrating user preferences in recommender systems;
- Chapter 3 explains how exploration works in the MAB framework;
- Chapter 4 covers our developed filtering methodology to make exploration more user-aware;
- Chapter 5 presents the experiments that have been conducted in cooperation with Muziekweb;
- Chapter 6 presents and discusses the experiment results to evaluate our proposed methodology; and
- Chapter 7 concludes the thesis by answering the research questions, summarizing the contributions, and presenting future work directions.

To support these chapters, several appendices have been added. Therefore, throughout the thesis report, we will refer several times to these appendices.

- Appendix A contains a paper created based on this thesis and submitted to SIGIR '20;
- Appendix B contains impressions of Muziekweb Explore features;
- Appendix C covers the technologies used to set up the Exploration Explorer website;
- Appendix D contains a walkthrough of Exploration Explorer; and
- Appendix E contains the research approval by TU Delft's Human Research Ethics Committee (HREC), considering our thesis experiments.

For acronyms and technical jargon, the reader is referred back to page xv. Note that acronyms are only written in full the first time that they are introduced in the thesis report.

(This page has been intentionally left blank.)

2

Related Work

Knees et al. (2019) conclude their literature review on user awareness in music recommender systems with the following statement: "User awareness is an essential aspect to adapt and balance systems between exploitation and exploration settings and not only identify the 'right music at the right time', but also help in discovering new artists and styles, deepening knowledge, refining tastes, broadening horizons — and generally be a catalyst to enable people to enjoy listening to music." [42]. This conclusion directly motivates the need for our research in the music domain and indirectly in any item domain.

In this chapter, we cover related work to present state-of-the-art methods that have been proposed to integrate user preferences in recommender systems. Section 2.1 covers work that discusses how to describe users to integrate user preferences in recommender systems. Section 2.2 then covers work related to explore–exploit item placement. State-of-the-art work related to the MAB framework is covered in Section 2.3. Finally, Section 2.4 discusses evaluation for user-aware exploration.

2.1. Describing Users

In this thesis, we consider user preferences with respect to exploration by recommender systems. In general, the question of how to describe users to serve them for recommendation has been widely studied. By means of a literature review, Laplante shows in the music domain that there are associations between music preferences and demographic characteristics, personality traits, personal values, and beliefs [45]. By exploiting these for recommender system design, one could improve the prediction ability of recommender systems [81]. Each of the aforementioned human factors has been shown to be a valid construct to describe people in psychology [45]. Based on state-of-the-art literature on recommendation, the focus has been so far on the development of personality-based systems [53, 69].

A widely accepted theory to describe personality in psychology is the five-factor OCEAN model [23]. Table 2.1 shows the five factors that it establishes. These can be assessed explicitly, using questionnaires for which a five-point Likert scale is often used, or implicitly, by means of data analysis [22]. Cantador et al. use the model to study the relations between personality types and user preferences in movies, TV shows, music, and books [17].

Schwartz proposes a model to describe values [74]. Roccas et al. relate the factors to personal values [71]. They conclude that the relations found help to clarify some issues regarding the meaning of the factors, which were previously debated. Furthermore, their findings support the idea that the influence of values on behavior depends more on cognitive control than the influence of traits. Using Netflix data, Golbeck and Norris are the first to apply the model to relate personality with movie rating and viewing history [31]. Adamopoulos and Todri automatically infer personality traits, needs, and values from social media content and build different recommender system models [5]. Using Amazon data, they find that the under-explored models of needs and values result in an improved predictive performance when compared to the more popular five-factor model. Manolios et al. (2019) hypothesize that value awareness in recommendation could especially be of interest for broadening the user's horizon, as it challenges the user to look beyond what is intuitively preferable [53]. Whereas personality defines who people are at the present moment, personal values define who people want to be in the future. Based on a qualitative study, they conclude that openness to change, self-transcendence, and self-enhancement were the values that appeared most frequently.

Table 2.1: The Five Factors From the Five-Factor OCEAN Model

Factor	Range	Reflects a person's tendency to
Openness	from cautious/consistent to curious/inventive	intellectual curiosity, creativity, and preference for novelty and variety of experiences
Conscientiousness	from careless/easy-going to organized/efficient	show self-discipline, aim for personal achievements, and to have an organized and dependable behavior
Extraversion	from solitary/reserved to outgoing/energetic	seek stimulation in the company of others and to put energy in finding positive emotions
Agreeableness	from cold/unkind to friendly/compassionate	be kind, concerned, truthful, and cooperative towards others
Neuroticism	from secure/calm to unconfident/nervous	experience unpleasant emotions and refers to the degree of emotional stability and impulse control

Based on these works, we conclude that certain traits and values relate to exploration by recommender systems, and thus will be relevant in assessing to what degree a user will be inclined towards exploration. To us, the most important personality trait is 'openness to experience' (see Table 2.1) [23] and the most important personal value is 'openness to change' [74].

2.2. Explore–Exploit Item Placement

Although explore–exploit strategies are widely researched in sequential contexts, their integration in ranking approaches is almost an empty field, according to Guillou (2016) [32]. To date, not much has changed since then. Therefore, we contribute to a rather unexplored research area. The most notable work on personalized ranking is by Rendle et al., which presents from a Bayesian analysis a generic optimization criterion BPR-OPT for Bayesian Personalized Ranking (BPR) [68]. Also, it presents a generic pairwise learning algorithm for optimizing models with respect to BPR-OPT.

Hofmann et al. are the first to propose methodology for pairwise and listwise online learning to rank that balances exploration and exploitation by recommender systems in implicit feedback settings [36]. Their methodology is inspired by the ϵ-greedy algorithm. In their pairwise approach, they generate an exploitative and an exploratory list. For each rank, their algorithm picks the corresponding item from the exploratory list with a fixed probability ϵ. Else, it picks the corresponding item from the exploitative list. In their listwise approach, they introduce a probabilistic interleave comparison function to allow Dueling Bandit Gradient Descent to learn a ranking function that balances exploration and exploitation. Radlinski et al. are the first to propose methodology to learn diverse rankings online based on clicking behavior [66]. They propose a greedy algorithm and an algorithm that uses a MAB on each rank, where each MAB instance is treated independently. Only the MAB that corresponds to the clicked item is updated.

Deep learning approaches have recently been applied to personalize ranking. He and McAuley work upon Rendle et al.'s work to propose Visual BPR, a scalable method that incorporates visual features extracted from product images into matrix factorization to uncover what visually most influences people's behavior [34]. Their model is trained with BPR using stochastic gradient ascent. He et al. propose Adversarial Personalized Ranking that enhances BPR by performing adversarial training [35]. Pei et al. propose a personalized re-ranking model to refine the initial list that is produced by state-of-the-art learning to rank methods [64]. They apply a Transformer network to encode the dependencies among items themselves as well as the interactions between the user and items. Using a pre-trained embedding to learn personalized encoding functions for different users, the performance of the re-ranking model can be further improved. Liu et al. propose a deep generative ranking model to enhance the accuracy and generalization of recommender systems [50]. Their experiments show significant ranking estimation improvements, especially for near-cold-start-users.

We have decided to not take a deep learning approach for our methodology. Firstly, because in our application context, we simply will not have the amounts of data such an approach demands. Secondly, for our purposes, *where* exploration vs. exploitation items are globally concentrated will be a more relevant question than devising an explicit ordered ranking of these items. In other words, we do not optimize against a specific ranking criterion, hence we speak of item 'placement'. Instead, we want our method to find a balance that suits the user in terms of the exploration–exploitation trade-off. For our item placement strategies, we want to introduce an exploration gradient over the result overview that behaves like the user wishes. This is strengthened by user experience constraints of Muziekweb; music album suggestions will be presented in a 2-dimensional mosaic, rather than a 1-dimensional ordered list. Therefore, we have decided to stay close to reinforcement learning and multi-armed bandits, as these concepts are all about finding such a balance [77].

2.3. Multi-Armed Bandits

To address the exploration–exploitation dilemma, Li et al. propose an adaptive bandit clustering algorithm that can perform collaborative filtering [48]. Efficiency is achieved through the use of sparse graph representations. McInerney et al. propose Bart for jointly personalizing recommendations and associated explanations to provide more transparent and understandable suggestions to users [54]. Bart learns how items and explanations interact within any provided context to estimate user satisfaction. Yet the authors would like to consider different exploration methodologies for more efficient exploration by the recommender. Sanz-Cruzado et al. propose nearest-neighbor bandits that consider users to be the arms, which can be chosen as potential neighbors [73]. Recommendations are produced based on the advice from a neighbor of the target user, where neighbor selection is based on a stochastic choice.

Pereira et al. propose Counterfactual Dueling Bandits (CDB) that take an online learning to rank (OL2R) approach for sequential music recommendation suited for scarce-feedback problems [65]. CDB continuously learns from the user's listening feedback with a reward model that requires a single implicit feedback signal from the user at each interaction. Wang et al. propose Document Space Projection (DSP) for OL2R to reduce variance in gradient estimation and to improve performance [85]. They propose DSP Dueling Bandit Gradient Descent as an example of their general method.

These recent works tackle the exploration–exploitation dilemma with fairly diverse approaches to increase the use of recommender systems. Our work contributes to this with methodology that makes exploration by recommenders more user-aware through integration of user preferences with respect to explore–exploit item placement, exploration, and familiarity.

2.4. Measuring the Effect of User Awareness

To evaluate our methodology, we want to measure the effect of the increase in user awareness in our recommendations. In Section 2.2, we stated that we do not want to optimize against a ranking criterion. Instead, we want our methodology to find a balance that suits the user in terms of the exploration–exploitation trade-off. In other words, aside from data-centric measurement, we want to measure the effect on user experience.

"Being accurate is not enough" for recommender systems, argue McNee et al. [55]. They state that recommenders should not only be accurate and helpful, but also a pleasure to use. To this end, we need metrics that act on the recommendation result overview, instead of on the items appearing in the overview. Fazeli et al. demonstrate that data-centric and user-centric evaluations results are not necessarily in line [24]. They assess recommenders based on five quality metrics: usefulness, accuracy, novelty, diversity, and serendipity. These findings reflect what we stated in Section 1.2. We take these warnings into account for the design of our thesis experiments.

(This page has been intentionally left blank.)

3

Exploring (With) Multi-Armed Bandits

In this chapter, we formally introduce the MAB framework and zoom in on exploration algorithms. Section 3.1 covers key concepts in RL and Section 3.2 explains the MAB framework.

3.1. Some Notes on Reinforcement Learning

In Section 1.5, we stated that the exploration-exploitation trade-off is one of the main challenges that arise in RL and that it makes RL different from other types of ML. In general, as an abstract definition, RL is learning what to do. In other words, it is about learning a mapping from situations to actions that maximizes a numerical reward [77]. Given the current situation, the learning agent discovers which actions yield which rewards by simply trying an action in the environment, which is termed 'trial-and-error search'. This becomes even more interesting as actions influence future actions and their corresponding rewards, which is termed 'delayed reward'. These two terms are the most important distinguishing features of RL [77]. Indeed, it could take a long time before the learner gets close to an optimal mapping.

With respect to the exploration-exploitation dilemma, the trade-off lies at the agent and is about the concern that, on the one hand, focusing on the best options seen so far is preferable to maximize its performance (need for exploitation), while on the other hand, focusing on the best options seen so far might make it miss out on some better options, possibly discarded due to unlucky trials (need for exploration) [28]. In general, it is at the core of optimal decision making under uncertainty. Also, historically related, it is at the core of the 'sequential design of experiments' field of the statistics domain, in which the MAB problem was originally formulated by Thompson (1933, [79]) and Robbins (1952, [70]) [28, 77].

Aside from the agent and environment, an RL system consists of four main elements: policy π, reward signal R, value function V, and, optionally, a model of the environment [77]. Policy π defines the learning agent's way of behaving at a given time step. Besides, reward signal R defines the goal in an RL problem, where for each time step the environment sends the relevant reward to the learner. In relation to this, the objective of the agent is to maximize the cumulative reward over the long run. Note that reward signals may be stochastic functions of the state of the environment and the actions taken. Furthermore, value function V specifies what is good in the long run. In other words, for a given state, it returns the best total reward obtained by performing the best known future steps based on previously gained experience. Lastly, the model allows inferences to be made about how the environment will behave. Models can be used for planning, as they may be used to predict the next state and next reward. Methods for solving RL problems that use a model are called model-based methods. Similarly, methods that do not are called model-free methods.

The rewards obtained by the actions taken for the given situations serve as evaluative feedback, instead of instructive feedback, which again distinguishes RL from other types of ML. Whereas evaluative feedback indicates how 'good' an action was, instructive feedback indicates the correct action to take, independent of the action taken [77]. As evaluative feedback does not indicate whether the action taken was the worst or best possible, it calls for active exploration. That is, to obtain a 'good' performance, a thorough search is needed. Indeed, efficiently obtaining a performance close to the optimal one, therefore, demands a clever approach.

3.2. Multi-Armed Bandits

Until now, we have been a bit vague about the precise relation between MABs and RL. The reason for this is that literature, in general, is as well. With respect to the previous section, bandit problems could be considered as unique settings of RL problems with the argument that they only have one state. This implies that the learner does not have to learn how to act in more than one situation. In other words, there is no need to plan for the future [46]. Therefore, one could argue that these problems are simplified settings of RL problems [77]. However, there is no consensus in literature; there is a large body of work that considers MAB to be a domain on its own, instead of a subdomain of RL. The main reason for this is the assumption that for a problem to fall into the RL realm, it does require long-term planning [46]. The goal of this section is to formalize the MAB framework, present bandit models, and explain exploration strategies.

3.2.1. A Formalization of the Basics

The term 'multi-armed bandit' is derived from slot machines, which are also known as one-armed bandits in American slang. A one-armed bandit has one lever, which implies that, considering bandit problems, it has an action set $\mathscr{A}$ that consists of one element. In general, for a bandit problem with a fixed number of arms, Definition 3.1 holds. Indeed, from this we can deduce Remark 3.2. For MABs, Definition 3.3 applies [46].

Definition 3.1. A K-armed bandit problem is a bandit problem with $K \in \mathbb{N}^+$ arms.

Remark 3.2. For a K-armed bandit, it holds that: $|\mathscr{A}| = K$.

Definition 3.3. A multi-armed bandit problem is a bandit problem of which, aside from the fact that it has at least two arms, the number of arms is irrelevant.

Moreover, Definition 3.4 formalizes bandit problems. To make a decision, we are primarily interested in mean rewards $\{\mu_1, \mu_2, \ldots, \mu_K\}$ of the bandits' reward distributions [76]. If the agent observes a reward, it does not observe the rewards of other arms that could have been pulled. This type of feedback is known as 'bandit feedback' [46, 76]. For simplicity, the rewards are often restricted to the interval $[0, 1]$. In the most basic formulation of a bandit problem, it is assumed that the arms set $\mathscr{A}$ is finite and the rewards are independent and identically distributed (iid), according to an unknown law with an unknown expectation [12]. Independence includes the independence of rewards across bandits.

Definition 3.4. In the bandit problem, a learner has access to an arms set $\{a_1, a_2, \ldots, a_K\}$ that corresponds with a set of real reward distributions $\{R_1, R_2, \ldots, R_K\}$, where $K \in \mathbb{N}^+$. The learner and the environment interact sequentially over $T \in \mathbb{N}^+$ rounds. In each round $t \in \{1, 2, \ldots, T\}$, the learner pulls some arm i, observes the reward, and obtains reward $R_{i,t}$ as a result. The learner's goal is to maximize the sum of the obtained rewards.

The objective to maximize the cumulative reward is given by Definition 3.5 [16]. To analyze the behavior of a learner, one could look at the learner's regret for not always behaving optimally [16]. Thus, the regret is defined as the difference of the cumulative optimal rewards and the cumulative obtained rewards over the time horizon. Definition 3.6 shows the relevant formula. For evaluation, expected regret $\mathbb{E}[\texttt{regret}_T]$ is often used [16, 76]. Remark 3.7 follows from these definitions.

Definition 3.5. Given a horizon of T time steps and each time step t pulled arm I_t, the cumulative reward is given by: $\texttt{cumulative\_reward}_T = \sum_{t=1}^{T} R_{I_t,t}$.

Definition 3.6. Given a horizon of T time steps, an arms set of size k, and each time step t pulled arm I_t, the cumulative regret is given by: $\texttt{regret}_T = \max_{i=1,2,\ldots,K} \sum_{t=1}^{T} R_{I_t,t} - \texttt{cumulative\_reward}_T$.

Remark 3.7. Maximizing the learner's cumulative reward is equivalent to minimizing the learner's regret.

3.2.2. Bandit Models

Many bandit models have been proposed over the past few decades and each model works with a different set of problem assumptions. The enumeration below provides an overview of the most popular bandit model types, including their main characteristics [46] and pointers to recent applications.

- *Stochastic bandits.* These are the most basic bandits and assume that each action has an iid sequence of rewards drawn from some distribution. Despite their relatively old age, researchers still come up with inventive ways to improve them, like with protection against adversarial attacks [1, 21, 40, 51, 80].

- *Adversarial bandits* [11]. These abandon almost all assumptions on how rewards are generated. The environment is often called the adversary. Strategies include randomization to deal with the adversary. Recent developments include the introduction of adversarial bandits with knapsack [10, 30, 38, 67].
- *Contextual bandits*. These include contextual information, which is dealt with by assuming that the mean reward of an action is a function of both the action and context features. Recent developments widely involve personalized recommendation, including fairness and explainability [39, 47, 54, 78].

Bandit models can also be non-stationary, which implies that the mean rewards of the arms change over time [29]. This can be the case for stochastic bandits as well, which assume stationarity by default [8]. Moreover, although the models above all work with a finite set of arms, there are also models that work with an infinite amount of arms. These may seem unattractive, as the term 'infinite' is often associated with 'intractable'. However, continuum-based bandits do so and, in fact, work with continuously many arms [6]. Here, the set of arms is $\mathscr{A} = [0,1]$ and expected rewards $\mu(\cdot)$ satisfy the following Lipschitz condition:

$$|\mu(a) - \mu(b)| \leq L \cdot |a - b| \quad \forall a, b \in \mathscr{A},$$

given a Lipschitz constant L. The Lipschitz condition is what makes them tractable. Such bandits are part of the Lipschitz bandits, which make no assumptions on the metric and allow for $\mathscr{A}$ to be a finite set [52].

Furthermore, several bandit models have recently been proposed that are gaining popularity, namely:

- *Combinatorial bandits* (2010) [19, 26, 62]. These demand the agent at each time step to choose multiple discrete variables, instead of a single one. Each arm represents a possible combination of the values. The number of arms to choose from is exponential in the number of variables.
- *Dueling bandits* (2012) [87]. These introduce the notion of relative feedback, where at each time step the agent chooses two arms. The feedback is a binary value that indicates which arm returned the highest reward. Researchers have recently translated existing algorithms for dueling bandits [27, 43, 86, 88, 89].
- *Collaborative bandits* (2016) [48]. These were also covered in Section 2.3. They extend contextual bandits to try to find preference related patterns, like collaborative filtering methods aim for. These bandits use adaptive clustering procedures at both the user and the item sides.

Another important bandit is the Bayesian bandit, for which the popular Thompson Sampling method can be applied [75, 79]. We explain this method in the Subsection 3.2.4. These bandits extend stochastic bandits with the Bayesian assumption. This is the assumption that the problem instance $\mathscr{I}$ is drawn initially from a known distribution $\mathbb{P}$, which is called the Bayesian prior [76]. Considering Definition 3.6, the goal of these bandits is to optimize the Bayesian regret, which starts from the expected regret:

$$\texttt{bayesian\_regret}_T = \underset{\mathscr{I} \sim \mathbb{P}}{\mathbb{E}} [\mathbb{E}[\texttt{regret}_T | \mathscr{I}]]$$

3.2.3. Choosing the Right Bandit

> "Your assumptions are your windows on the world.
> Scrub them off every once in a while,
> or the light won't come in."
>
> — Isaac Asimov

Although the preceding quote was originally used in a different context, it is applicable to choosing a bandit model for a given problem as well. In practice, problems can be dynamic, which implies that holding on to assumptions can be unrealistic. As seen in the previous subsection, many different types of bandit models exist, each built on a different set of assumptions.

Given a problem, a model can fail in two fundamentally different ways [46]. It can be too specific; that is, it makes assumptions that together are too strict and thus detach the model from reality, causing a mismatch in actual and predicted performance. On the other hand, it can be too general; that is, the assumptions of the model are too loose, which hurts the performance. A 'good' model should be somewhere in between, making assumptions that are specific for the problem, yet not too specific such that variance is allowed for.

One should also note that not all assumptions made by a model are equally important. For example, stochastic bandits assume that the mean reward of an arm does not significantly change over time, if not change at all. Also, they assume that rewards are drawn from an arm-dependent probability distribution. The former assumption is much more significant than the latter for the performance. Therefore, even though a model makes an assumption that is not in line with the problem, it may still be an appropriate model, if the assumption is not too important and other assumptions are.

To summarize, choosing the 'right' bandit is not trivial. However, in general, we should not worry too much about finding the perfect bandit model. To quote statistician George E. P. Box: "All models are wrong, but some are useful". In other words, we should pick the models that seem most useful. Therefore, before trying a bandit model, a careful analysis of the problem already gives a head start.

3.2.4. Exploration Strategies

Like there are many proposed bandit models, many bandit algorithms have been proposed to deal with the exploration-exploitation trade-off. We cover the most popular ones in this subsection. One could perform no exploration at all; however, this is the most naive approach. In general, one could argue that there are two types of exploration strategies: *random exploration* and *uncertainty-based exploration*.

There also exist exploration algorithms that perform exploration and exploitation in phases. Such algorithms usually explore in the exploration phase by trying all arms at least once. For instance, the Explore-First algorithm, as the name suggests, first explores during a fixed number of rounds and then exploits in the remaining rounds. However, the performance of this method is rather bad, as its performance in the exploration phase is bad. In general, such methods are hardly ever used, because it is usually better for the performance to spread exploration over time.

Random Exploration

In Section 1.3, we informally explained the ϵ-greedy algorithm, which is an algorithm that spreads exploration over time. This is a random exploration method, as it explores with probability ϵ_t, given time step t. Algorithm 1 formally defines the method, where each $\epsilon_t \in [0,1]$ and T is defined as in Subsection 3.2.1.

Algorithm 1 ϵ-greedy

```
function epsilon_greedy($\epsilon$, $T$)
    for each time step $t = 1, 2, \dots, T$ do
        Toss a coin with success probability $\epsilon_t$
        if success then
            Explore: pull an arm uniformly at random
        else
            Exploit: pull the arm of the action with the highest obtained average reward
```

For adversarial bandits, the simplest algorithm is Exponential-weight algorithm for Exploration and Exploitation (Exp3) [13], for which a family of variant algorithms is proposed. It is given by Algorithm 2, where $\gamma \in (0,1]$, and K and T are defined as in Subsection 3.2.1. This method is a random exploration method too and a variant of the popular Hedge algorithm for dynamic resource allocation [25]. Based on the probability mixtures, Exp3 explores uniformly randomly with probability γ. Moreover, it exploits with probability $1-\gamma$, where it prefers arms with high weights. If the reward is received, the weights are updated accordingly. The method assigns significantly higher weights to arms that were pulled and obtained relatively high rewards.

Algorithm 2 Exp3

```
function exp3($\gamma$, $K$, $T$)
    Initialize $w_i(1) = 1$ for $i = 1, 2, \dots, K$
    for each time step $t = 1, 2, \dots, T$ do
        $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$ for $i = 1, 2, \dots, K$
        Draw action $i_t$ randomly accordingly to the probabilities $p_1(t), p_2(t), \dots, p_K(t)$
        Receive reward $x_{i_t}(t) \in [0,1]$
        for each action $j = 1, 2, \dots, K$ do
            $\hat{x}_j(t) = \begin{cases} \frac{x_j(t)}{p_j(t)}, & \text{if } j = i_t \\ 0, & \text{otherwise} \end{cases}$
            $w_j(t+1) = w_j(t) \exp(\gamma \hat{x}_j(t)/K)$
```

A simple algorithm for contextual bandits is Exponential-weight algorithm for Exploration and Exploitation using Expert advice (Exp4) [13, 46], for which a family of variant algorithms is proposed as well. As the name suggests, it extends the Exp3 algorithm, namely with expert advice. In general, these experts could represent simple decision rules and, depending on the context, could be literal human beings [18]. The pseudocode is given by Algorithm 3, where $N \in \mathbb{N}^+$ is the number of experts and γ, K, and T are the same as in Exp3. Also, $\boldsymbol{y}(t) \in [0,1]^N$ is the vector that corresponds to the gains of the experts. No assumptions are made about the quality of the experts beforehand, as line 2 of Algorithm 3 shows. As in Exp3, Exp4 explores uniformly randomly with probability γ and exploits with probability $1-\gamma$. However, for exploitation, it is not the actions that compete, but the experts. The algorithm exploits with a preference towards the advice of the experts that are assigned the highest weights, which for each iteration are updated based on the experts' performances with respect to the drawn action. The most simple expert is the uniform expert, which always assigns a uniform weight to all actions.

Algorithm 3 Exp4

```
1:  function exp4($\gamma, K, T, N$)
2:      Initialize $w_i(1) = 1$ for $e = 1, 2, \ldots, N$
3:      for each time step $t = 1, 2, \ldots, T$ do
4:          Get advice vectors $\boldsymbol{\xi}^1(t), \boldsymbol{\xi}^2(t), \ldots, \boldsymbol{\xi}^N(t)$
5:          Set $W_t = \sum_{i=1}^N w_i(t)$
6:          Set $p_j(t) = (1-\gamma)\sum_{i=1}^N \frac{w_i(t)\xi_j^i(t)}{W_t} + \frac{\gamma}{K}$ for $j = 1, 2, \ldots, K$
7:          Draw action $i_t$ randomly accordingly to the probabilities $p_1(t), p_2(t), \ldots, p_K(t)$
8:          Receive reward $x_{i_t}(t) \in [0,1]$
9:          for each action $j = 1, 2, \ldots, K$ do
10:             $\hat{x}_j(t) = \begin{cases} \frac{x_j(t)}{p_j(t)}, & \text{if } j = i_t \\ 0, & \text{otherwise} \end{cases}$
11:         for each expert $e = 1, 2, \ldots, N$ do
12:             $\hat{y}_e(t) = \boldsymbol{\xi}^e(t) \cdot \hat{\boldsymbol{x}}(t)$
13:             $w_e(t+1) = w_e(t) \exp(\gamma \hat{y}_e(t)/K)$
```

Uncertainty-Based Exploration

The previous three presented popular algorithms explore uniformly randomly, which gives us the opportunity to try out options that we do not know much about. However, it is possible that we end up exploring an item that we already explored in a previous round and turned out to yield a low reward. There are multiple ways to prevent this kind of inefficient exploring. One way is to decrease ϵ over time. Another is to prefer actions for which we have not obtained a confident value estimation yet. Therefore, in the latter approach, we prefer exploration of actions with a strong potential to yield high, if not optimal, rewards.

The Upper Confidence Bound (UCB) algorithm measures this potential, as the name suggests, by means of an upper confidence bound of the true mean reward [12]. A family of UCB algorithms has been proposed, where each algorithm member works with an upper confidence bound. That is, $X_a \leq \hat{X}_{a,t} + \hat{U}_{a,t}$, where, given action a and time step t, X_a is the true mean reward, $\hat{X}_{a,t}$ is the predicted mean reward, and $\hat{U}_{a,t}$ is the upper bound. Upper bound $\hat{U}$ is computed using the number of times that we have pulled the arm so far. If the number of pulls increases, the upper confidence bound decreases. UCB selects an action in a greedy fashion:

$$a_t = \arg\max_{a \in \mathscr{A}} \hat{X}_{a,t} + \hat{U}_{a,t}.$$

What makes each member of the UCB family different, is the way in which it computes the upper bound $\hat{U}$.

The simplest algorithm member is UCB1, which is derived from index-based policies [7] and derives the upper bound using Chernoff-Hoeffding bounds. It is given by Algorithm 4, where K and T are the same as in the previous subsection. Also, $\hat{\boldsymbol{X}}$ and $\boldsymbol{n}$ contain of each action the average reward and the number of pulls so far, respectively. We rely on Hoeffding's inequality to make a very general estimation of each reward distribution. However, for the sake of efficiency, we could assume a prior on the reward distribution to narrow the estimated bound. Bayesian UCB allows for such a prior.

Algorithm 4 UCB1

```
function ucb1($K$, $T$)
    Initialize $\hat{\boldsymbol{X}}$ by pulling each action's arm once
    Initialize $n_i = 1$ for $i = 1, 2, \dots, K$
    for each time step $t = 1, 2, \dots, T$ do
        Pull the arm of action $i = \arg\max_{\text{action } j} \hat{X}_j + \sqrt{\frac{2 \ln \sum_{k=1}^{K} n_k}{n_j}}$
        Update $\hat{X}_i$ and $n_i$ accordingly
```

A simple algorithm for Bayesian bandits is Thompson Sampling (TS) [46, 72, 76, 79], which employs a policy according to its action and reward history. It is given by Algorithm 5, where T is defined as before. Also, H contains the history of the previously performed actions and obtained rewards, and $\boldsymbol{\mu}_t \in [0,1]^K$ is the sampled mean reward. For each time step t and action a, it computes the posterior probability that a is the best action and then chooses the action with the highest mean reward. If we assume the priors to be independent, line 5 of Algorithm 5 becomes simpler, as we then consider the posterior distribution $\mathbb{P}_H^a$ of each action a separately.

Although TS is mathematically elegantly defined, its computation of and sampling from the posterior distribution may become intractable. There exist several special bandit cases that allow for a faster computation, which relax the problem setting. Two well-known cases are the beta-Bernoulli and Gaussian bandit [46, 72]. The beta-Bernoulli bandit assumes Bernoulli rewards and the prior to be the uniform distribution on the $[0,1]$ interval. The posterior distribution is then the Beta distribution. Furthermore, the Gaussian bandit assumes a Gaussian reward distribution and a Gaussian prior. The posterior distribution is then also a Gaussian. These posterior distributions are well-known. The speed-up comes from the facts that they are easy to derive and there exist time-efficient algorithms to sample from them.

Algorithm 5 Thompson Sampling

```
1: function thompson_sampling($T$)
2:     Initialize list $H \leftarrow \emptyset$
3:     for each time step $t = 1, 2, \dots, T$ do
4:         Observe $H_{t-1} = H$, for some feasible $(t-1)$-history $H$
5:         Sample $\boldsymbol{\mu}_t$ from the posterior distribution $\mathbb{P}_H$
6:         Pull the best action's arm according to $\boldsymbol{\mu}_t$
7:         Update $H$ accordingly
```

4 Filtering Methodology

In this chapter, we propose two filtering techniques, which deal with the problem that is stated in Chapter 1. In Section 4.1, we lay the foundations for our methodology by defining the core of recommender systems that work from the exploration-exploitation paradigm. The two sections that follow, Section 4.2 and 4.3, present the methods that build upon these foundations: explore-exploit item placement strategies and a user-aware exploration algorithm, respectively. Finally, in Section 4.4, the two methods are merged to yield our taste-broadening recommendation approach. Each of the last three sections introduces the concept, highlights the architectural implications, provides the pseudocode, and shows time efficiency. In the presented architectures, Alice from Section 1.3 performs the role of a general user.

4.1. Starting From the Core

As an abstract starting point, we adopt Ricci et al.'s definition of recommenders: "Recommender systems are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user" [69]. In other words, this definition assumes the recommender system to be a black box that outputs useful suggestions to the user. For our methods, we take an RL approach, which assumes that the black box can perform processes related to exploration and exploitation, and can seek to learn what to do with respect to user preferences [77].

Figure 4.1 visualizes the minimal architecture of such an abstract recommender that works from the exploration-exploitation paradigm to make a recommendation. We have not visualized further details, like how the user's feedback is handled, to keep the figure as simple as possible. The exploration process produces item candidates. It is an autonomous process, since it does not have any incoming arrows. The user preferences gathered by the recommender are used as input for the exploitation process. This process is a function that outputs item candidates that are optimal to recommend for the user, according to some score function. The item candidates output by each of the two processes are, upon request of the user, delivered to the explore-exploit item placement process. This process selects items from the two candidate sets, according to some algorithm, and merges these into a single representation. This item overview is then sent to the user. The user, in turn, interprets the representation and consumes it. The response (e.g., clicks) of the user serves as feedback to the recommender system.

These processes together form the cycle of actions that represent the recommendation process. Each request by the user triggers an iteration of this cycle. Throughout the iterations, with some appropriate explore-exploit balance, the recommender system improves the quality of the recommendations for the user.

For the sake of completeness, as for corner cases, this blueprint allows for each produced candidate set to be of size zero or one. Moreover, aside from the minimal flow, it does not make any assumptions on the underlying processes' algorithms. This implies that the covered recommender systems could recommend one item, as well as only explore or only exploit. Thus, this blueprint covers the core idea of *any* recommender that works from the exploration-exploitation paradigm – even the extreme ones.

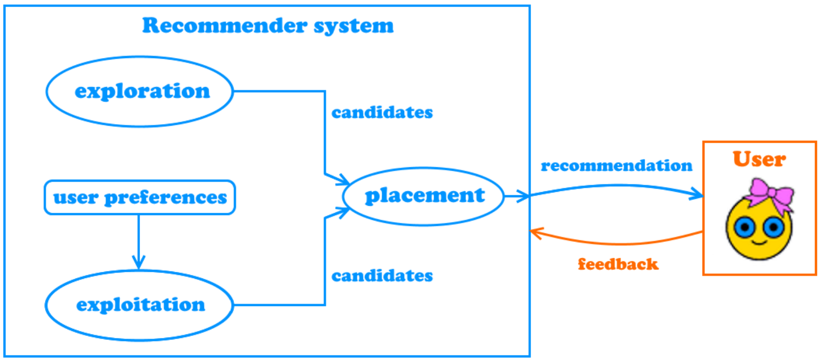


Figure 4.1: The Minimal Recommendation Flow of an Exploration-Exploitation Paradigm Recommender.

4.2. Explore-Exploit Item Placement Strategies

In this section, we build forth on the recommender core that was explained in the previous section by proposing explore-exploit item placement strategies. We introduce the concept in Subsection 4.2.1. Then, we describe the implication on the architecture in Subsection 4.2.2. Finally, in Subsection 4.2.3, we present the pseudocode of our algorithm and show its time efficiency.

4.2.1. Concept

We introduce explore-exploit item placement strategies as defined by Definition 4.1. To integrate these in a recommender system, we need to introduce an explore-exploit item placement strategy generator. The generator takes as input user preferences with respect to *explore-exploit item placement* and *exploration*, and outputs an explore-exploit item placement strategy that suits the user. As for ϵ-greedy, which was explained in Section 1.3, one could interpret this strategy to be a sequence of ϵ values, where $\forall i\colon \epsilon_i$ is tuned using the user preferences. The produced item placement strategy is then used to place the exploration and exploitation candidates accordingly, which results in the recommendation. As a result of integrating the strategy, the placement process becomes user-aware.

Definition 4.1. Explore-exploit item placement strategies are strategies to merge exploration and exploitation item candidates deduced by the recommender system in a single item overview such that the explore-exploit item placement suits the user.

4.2.2. Architectural Implication

To integrate explore-exploit item placement strategies, a strategy generator should be added to the architecture. It should receive user preferences with respect to explore-exploit item placement (p) and exploration (e) as input, such that a placement strategy can be generated from it. The generator can be part of the placement process or a separate process that passes the strategy to the placement process. The difference is subtle; it depends on the level of detail from which we look at it. For ease of illustration, we assume the former. The adapted architecture that achieves this, is visualized in Figure 4.2.

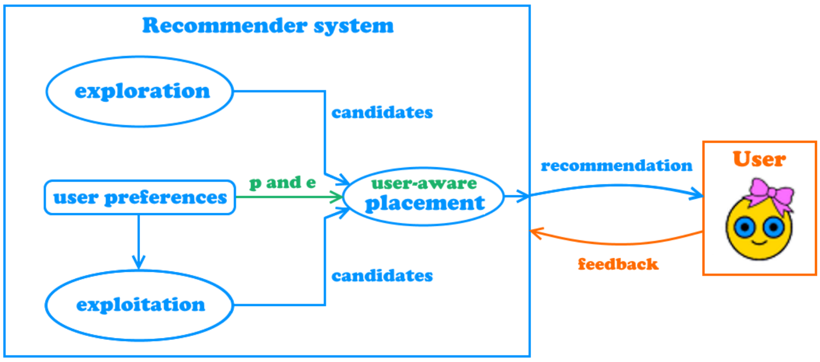


Figure 4.2: Minimal Recommendation Flow to Include Explore-Exploit Item Placement Strategies.

4.2.3. Towards an Optimal Strategy

The goal of the explore-exploit item placement strategy generator is to optimize the fit between the user's preferences it receives and the placement strategy it produces. Our generator takes into account user variables with respect to explore-exploit item placement and exploration rate. As a starting point, we use the basic RL algorithm ϵ-greedy, as introduced in Section 1.3. Independent of the output representation, a placement strategy can then be modeled using two factors: a probability distribution and a scaling factor. Whereas a probability distribution gives shape to the model, the scaling factor corrects for the model's emphasis towards either exploration or exploitation. The scaling factor allows the area under the eventual distribution to be unequal to one. Indeed, the model's distribution is not necessarily a probability distribution. The individual values that can be drawn from the distribution require to be accurate with respect to the user's needs, which does not introduce any restrictions on the area under the distribution. The question that then remains is: how to determine the probability distribution and the scaling factor?

Determining the Probability Distribution and the Scaling Factor

For the probability distribution, we choose to use the multivariate skew-normal distribution, proposed by Azzalini and Dalla Valle [15]. This is a multivariate parametric family of distributions that includes the standard normal as a special case, where the extra parameter regulates the skewness [14, 15]. This is a promising probability distribution, as it inherently allows for 'customizing' the probability distribution, yet including the standard normal distribution, to an extent that is useful for our problem. For this reason, our developed method uses the multivariate skew-normal distribution. We term the skew-normal Probability Density Function (PDF) '`skewnormal_pdf(x, a)`', which returns the probability of representation position-based value x, for a skew-normal distribution skewed with parameter a.

In 1D, if the skewness value decreases, the PDF gets skewed relatively more to the left. Similarly, if it increases, the PDF gets skewed relatively more to the right. If it is zero, the PDF is a standard normal distribution. In 2D, to realise Muziekweb's mosaics from Section 1.7, we adopt a simplified version, namely a skew-normal PDF over the diagonal. Figure 4.3 illustrates this. With this version, we assume that users start to consume items at the top-left of the mosaic and proceed by moving to the bottom-right. Therefore, in the figure, the item positions on every k-antidiagonal of the heatmaps are treated the same.

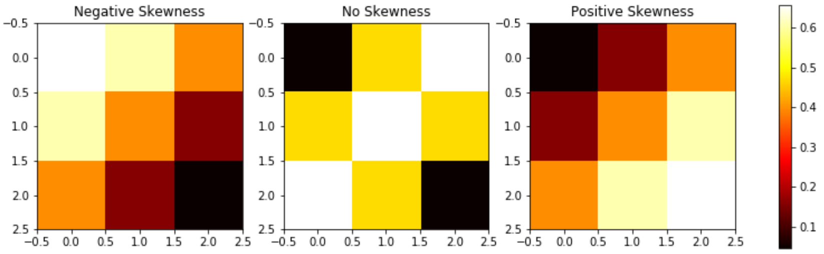


Figure 4.3: Three 3x3 Example Skewness Heatmaps.

To use the `skewnormal_pdf` function, we come up with a one-to-one function that maps the positions of the representation to the range of possible x values. This mapping strictly depends on the underlying representation that the recommender produces. The requirements of this mapping can be simplified further for our purposes. In practice, we are only interested in representations that are discretized and either 1-dimensional or 2-dimensional (i.e., lists or mosaics, respectively). A logical starting point for the mapping is to map the center of the representation to the origin and respect the topological structure accordingly. There are two reasons for doing this: (1) the standard normal distribution, which is obtained if $a = 0$, is symmetric with respect to the origin and (2) the skew-normal distribution obtained using a, corresponds to the skew-normal distribution mirrored in the y-axis that is obtained using $-a$. Also, to limit the range that the location values are mapped upon, in 1D, which is in fact $(-\infty,\infty)$, a rule of thumb could be to shrink it to $[-4, 4]$ and distribute the positions accordingly. The reason for doing this, is the fact that from very large to very small values of a, values outside this range result in probabilities smaller than 0.027%. Note that it is a rather weak rule of thumb, as depending on the scaling factor and the representation, a larger range might be desirable. However, it does show the need to define a more practical range.

The parameters that remain to be determined are the scaling parameter and, to use the function, skewness parameter a. These parameters are determined using the input of the strategy generator; the user preferences with respect to explore-exploit item placement and exploration. The user's preferred explore-exploit item placement is used to determine a. It is relatively easy to determine the sign of a, as well as whether a is zero or non-zero. If the user prefers exploration at the start of the representation, $a < 0$. Similarly, if the user prefers exploration at the end of the representation, $a > 0$. For exploration at the center of the representation, $a = 0$. However, the precise value of the a should be tuned. To shrink the possibilities, a rule of thumb could be to choose $a \in [-15, 15]$, as values outside of this range will only slightly further affect the skew-normal PDF. Besides, the user's preferred exploration rate is used to determine scaling parameter s, which should be tuned as well. Trivially, the lower bound of the range of s is 0. However, there is no exact upper bound, as probabilities resulting from the skew-normal PDF can be arbitrarily small. For simplicity, we assume the mappings that compute the skewness and scaling factor to be linear. However, linearity is not a requirement.

Our Algorithm

The pseudocode of our resulting algorithm to generate explore-exploit item placement strategies is given by Algorithm 6. Table 4.1 summarizes the notation used. Depending on the dimension of the representation, the `positions_to_pdf_input()` function maps the item positions of the representation to a list of inputs for the skew-normal PDF that respects the item position ordering. The hyperparameters are tuned beforehand to calculate a skewness and scaling factor that suit the user. Assuming that the `positions_to_pdf_input()` function runs in $O(|X|)$ time, the algorithm runs in $O(|X|)$ time too, as the for-loop does not exceed this bound.

Table 4.1: Explore-Exploit Item Placement Strategy Summary of Symbols

Symbol	Meaning
p	Explore-exploit item placement, $p \in [0,1]$
e	Exploration rate, $e \in [0,1]$
a	Skewness factor, $a \in \mathbb{R}$
s	Scaling factor, $s \in \mathbb{R}_{\geq 0}$
$\alpha_1, \alpha_2, \beta_1, \beta_2$	Hyperparameters
X	List of skew-normal PDF input values
S	Produced item placement strategy

Algorithm 6 Item Placement Strategy Generation

```
function generate_item_placement_strategy($p$, $e$)
    Initialize list $S \leftarrow \emptyset$
    Determine list $X \leftarrow$ positions_to_pdf_input()
    $a \leftarrow \alpha_1 \cdot p + \beta_1$
    $s \leftarrow \alpha_2 \cdot e + \beta_2$
    for $i = 1, 2, \ldots, |X|$ do
        $S(i) = s \cdot$ skewnormal_pdf($X(i)$, $a$)
    return $S$
```

4.3. A User-Aware Exploration Algorithm

In this section, we build forth on the recommender core that was explained in Section 4.1 by proposing a user-aware exploration algorithm for taste broadening recommendation. We introduce the concept in Subsection 4.3.1. Then, we describe the implication on the architecture in Subsection 4.3.2. Finally, in Subsection 4.3.3, we present the pseudocode of our algorithm and show its time efficiency.

4.3.1. Concept

For taste broadening recommendation, we propose a user-aware exploration algorithm that integrates user preferences with respect to *exploration willingness* and *familiarity*. Figure 4.4 visualizes the concept for three different users. The figure is analogous to Figure 1.2 and is its practical version. It shows three user Familiarity Profiles (FPs) for items that are fit for their taste broadening, which fan out in a *2D conelike shape*. We assume that a user's exploration willingness and category familiarity will influence from what part of the FPs we can draw recommender items. The exploration willingness and category familiarity are mapped to familiarity border b and exploration range r, respectively. For Alice, who is a rather conservative person, suitable items should remain close to her existing familiarity boundaries. For Bob, who is more adventurous, we can afford searching in a wider range. For Cynthia, who is an expert in category x, only truly unfamiliar items will allow for exploration. Therefore, the FPs can be used to search for taste broadening items. An FP is built and maintained based on the familiarity scores of the items in the available category item set, as opposed to relevance scores. These values are computed using a familiarity score function.

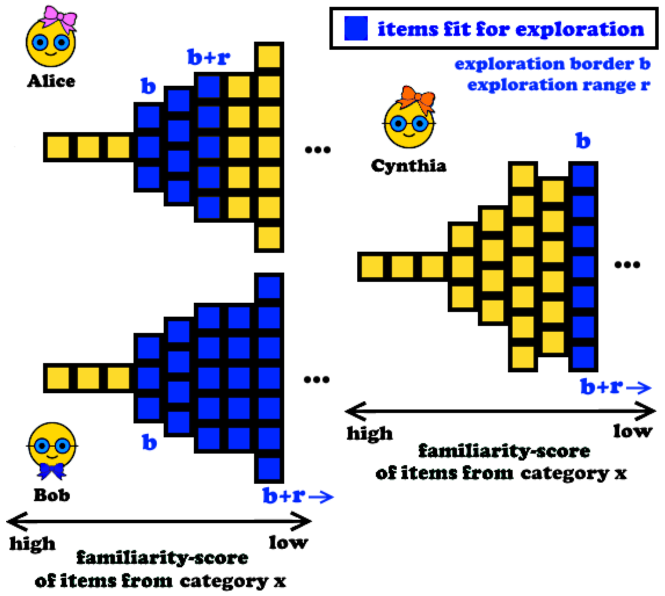


Figure 4.4: Different Taste Broadening Recommendation User Requirements.

4.3.2. Architectural Implication

To implement a user-aware exploration algorithm, the exploration process should receive user preferences. Our algorithm receives user preferences with respect to category familiarity (f) and exploration willingness (w) as input, such that exploration item candidates that are fit for taste broadening can be generated from it. The adapted architecture that achieves this, is visualized in Figure 4.5.

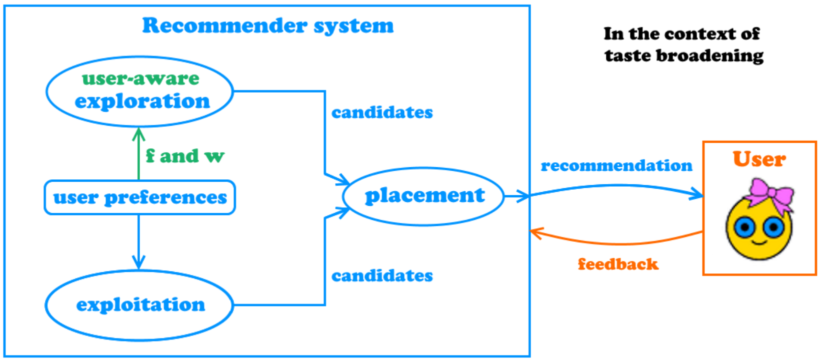


Figure 4.5: Minimal Recommendation Flow for a User-Aware Exploration Algorithm, in the Context of Taste Broadening.

4.3.3. Our User-Aware Exploration Algorithm

For taste broadening recommendation, the goal of a user-aware exploration algorithm is to optimize the fit between the user's taste broadening preferences and the exploration item candidates. Our algorithm takes into account user variables with respect to category familiarity and exploration willingness. It uses these variables to search the user's FP, which was introduced in Subsection 4.3.1. To build an FP, the familiarity score of each item in the item set needs to be initialized beforehand.

The Preprocessing Step

The pseudocode to initialize the user's FP is given by Algorithm 7. Table 4.2 summarizes the notation used. Note that A is only a temporary list. The items from the item set are sorted on familiarity score in nonincreasing order and are added to the FP, depending on the familiarity score range, respecting their order. Depending on the familiarity score function, a familiarity score range is picked. The range serves as a measure to generally have a sufficient number of taste broadening items to choose from, as we move away from the user's comfort zone. Depending on the familiarity score function, a balance should be found for δ by choosing it between the minimum and maximum value that correspond to the following two conditions: (1) there exists a reset of A, where before the reset: $|A| > 1$ and (2) before the first reset of A: $A \neq I$, respectively. If the familiarity score function allows it, for ease, the range could be set to 0. Assuming that an efficient sorting algorithm is used, the algorithm runs in $O(|I| \log |I|)$ time. As the algorithm to initialize the FP is now presented, we can proceed to our user-aware exploration algorithm.

Table 4.2: FP Initialization Summary of Symbols

Symbol	Meaning
j	The first in-order item that falls outside of range δ, $j \in \mathbb{N}$
f_i	Familiarity score of i, $f_i \in \mathbb{R}$
d_i	Identifier of i, $d_i \in \mathbb{N}$
δ	Familiarity score range, $\delta \in \mathbb{R}_{\geq 0}$
I	Category item set
F	Produced Familiarity Profile

Algorithm 7 Familiarity Profile Initialization

```
function initialize_familiarity_profile($I$)
    Sort list $I$ on familiarity score in nonincreasing order
    Initialize list $F \leftarrow \emptyset$
    Initialize list $A \leftarrow \emptyset$
    $j \leftarrow 1$
    for $i = 1, 2, \ldots, |I|$ do
        if $f_i + \delta \geq f_j$ then
            Append $d_i$ to $A$
        else
            Append $A$ to $F$
            $A \leftarrow \emptyset$
            $j \leftarrow i + 1$
    return $F$
```

Our Algorithm

The FPs can be used to search for taste broadening items. To do so in a way that suits the user, the user's category familiarity and exploration willingness are mapped to a suitable familiarity border and exploration range, respectively. This mapping is done beforehand for the sake of efficiency. For every request from the user to make a recommendation, before the exploration process is called, these mappings are executed. We introduce the mappings in the following section, which serves as an overview section to combine our two proposed filtering techniques.

The pseudocode for our user-aware exploration algorithm is given by Algorithm 8 and 9, which together define three cooperating functions. Table 4.3 summarizes the notation used. Note that F was defined earlier. Algorithm 9 serves as a helper function that gets the taste broadening exploration subspace of the FP for Algorithm 8. To ensure the absence of duplicates in the recommendation, the items in the recommendation so far are excluded (see line 8 of Algorithm 9). Algorithm 8 is a recursive function that draws a random item from the retrieved subspace. A recursive call is performed if the retrieved subspace is empty. These algorithms *do not update the FP*, as the taste broadening item is not yet consumed by the user.

Moreover, we define three corner cases that can occur:

(1) *Negative exploration range.* For the sake of generality, we take into account that a user could have a negative attitude towards exploration, which means that the user prefers to stay inside the own comfort zone.

(2) *Empty taste broadening subspace.* As line 8 of Algorithm 9 takes a set difference , the taste broadening subspace could be empty. This requires an exploration range change.

(3) *Start or end FP index reached.* At initialization or on the fly, the start or the end index of the FP could be reached. If the retrieved taste broadening subspace is empty, the exploration range direction is inverted.

If we assume the size of the retrieved taste broadening subspace to be at least equal to the number of item positions in the representation (i.e., we are not in corner case 2), Algorithm 8 and 9 run in $O(|F|)$ time, where $|F|$ refers to the number of lists in the FP. Under this assumption, the worst-case is reached if $|K|$ from line 7 in Algorithm 9 equals $|F|$, which implies that the familiarity border is at the start or end index of the FP and the exploration range covers the full length of the FP.

Table 4.3: FP-Based Exploration Summary of Symbols

Symbol	Meaning
b	Familiarity border, $b \in \mathbb{N}$
r	Exploration range, $r \in \mathbb{Z}$
B	Taste broadening subspace of F
K	Taste broadening indices subset of F
O	So far generated recommendation output

Algorithm 8 Familiarity Profile Based Exploration

```
function explore($F, b, r, O$)                              ▷ Corner case 1
    Determine $B \leftarrow$ get_tb_subspace($F, b, r, O$)
    return explore($F, b, r, O, B$)

function explore($F, b, r, O, B$)
    if $|B| > 0$ then
        return draw a random item from $B$
    else                                                    ▷ Corner case 2
        if $r \geq 0$ then
            if $b + r \geq |F|$ then                        ▷ Corner case 3a
                $r = -1$
            else
                $r$ += 1
        else
            if $b + r \leq 1$ then                          ▷ Corner case 3b
                $r = 0$
            else
                $r$ -= 1
        $B$ = get_tb_subspace($F, b, r, O$)
        return explore($F, b, r, O, B$)
```

Algorithm 9 Taste Broadening Subspace Retrieval

```
function get_tb_subspace($F$, $b$, $r$, $O$)
    Initialize $K \leftarrow \emptyset$
    if $r \geq 0$ then
        $K \leftarrow$ the indices from $b$ to $b + r$
    else
        $K \leftarrow$ the indices from $b + r$ to $b - 1$
    $B \leftarrow$ the flat list of $F(k)\ \forall k \in K$
    return $B - O$
```

For the sake of completeness, if user feedback is received of the recommendation consumption, it could be used to update the FP. Consumed items are then given the highest possible familiarity score. Firstly, a database update is required to guarantee an up-to-date category item set when it is queried. Secondly, we use the familiarity score range and the old familiarity score to efficiently determine the location of the item in the FP. Based on the old familiarity score, if we have found the item's list in the FP using the familiarity score range, we use an efficient search algorithm (e.g., binary search) to locate the item. We move the located item to the start of the FP. Finally, we remove empty lists from the FP. An update runs in $O(\log|I|)$ time, where in the worst-case the length of the FP is one. However, the familiarity score range tries to prevent this.

4.4. Merging the Two Methods

The two proposed filtering techniques can be merged to construct our overall recommender for taste broadening recommendation. In other words, the architectural changes required for the recommendation system can be merged without any conflicts. The union of the architectural changes is presented by Figure 4.6.

We propose Algorithm 10 for taste broadening recommendation. Table 4.4 summarizes the notation introduced. The algorithm calls Algorithm 6 to generate the item placement strategy and, based on it, decides whether to explore with Algorithm 8 or to exploit. Aside from returning an item, we do not make any assumptions about the `exploit()` function, which implies that *any exploit algorithm can be implemented*. The hyperparameters are tuned beforehand to calculate a familiarity border and an exploration range that suit the user. For simplicity, we assume the mappings that compute the familiarity border and exploration range to be linear. However, linearity is not a requirement. Leaving out the underlying `exploit()` algorithm, the algorithm runs in $O(|S| \cdot |F|)$ time, because the dominant part of the algorithm is the for-loop, for which the explore algorithm that runs in $O(|F|)$ time, is called at most $|S|$ times. Based on the derived time-complexities, the proposed algorithms are time-efficient by definition, which is of great value for recommendation.

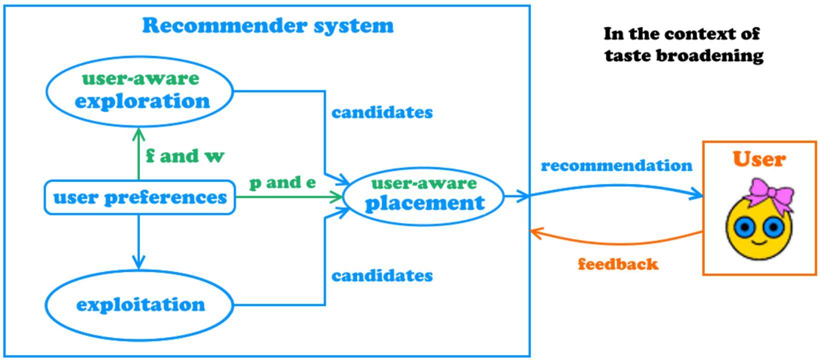


Figure 4.6: Minimal Recommendation Flow for Both Filtering Techniques, in the Context of Taste Broadening.

Table 4.4: Taste Broadening Recommendation Summary of Symbols

Symbol	Meaning
f	Category familiarity, $f \in [0,1]$
w	Exploration willingness, $w \in [-1,1]$
$\alpha_3, \alpha_4, \beta_3, \beta_4$	Hyperparameters

Algorithm 10 Taste Broadening Recommendation

```
function recommend_tb_representation($F$, $p$, $e$, $f$, $w$)
    Initialize $O \leftarrow \emptyset$
    Determine $S \leftarrow$ generate_ranking_strategy($p$, $e$)
    $b \leftarrow \alpha_3 \cdot f \cdot |F| + \beta_3$
    $r \leftarrow \alpha_4 \cdot w \cdot |F| + \beta_4$
    for $i = 1, 2, \ldots, |S|$ do
        Toss a coin with success probability $S(i)$
        if success then
            $O(i)$ = explore($F$, $b$, $r$, $O$)
        else
            $O(i)$ = exploit()
    return $O$
```

(This page has been intentionally left blank.)

5

Experiments

In this chapter, we frame the experiments to measure the effect of the filtering methodology proposed in the previous chapter. That is, the goal of the experiments is to measure the effect of including the user through item exploration preferences in the algorithmic exploration process. However, there are many effects that we can measure, as well as many metrics that we can use to measure. Therefore, framing the experiment is not straightforward. As we believe that users should have a say in how to let the recommender explore, we decide to test our belief through our experiments by conducting a user study in a taste broadening recommendation setting. We aim to allow the user to influence the algoritmic exploration process through an interactive user interface, as we believe that users are especially willing to personalize the exploration process when broadening their tastes. In our study, we measure the user's tendency to prefer recommendations from recommenders that include our filtering methodology. Because user studies inevitably introduce time constraints, the implemented recommender is pre-trained and does not perform any updates. (Although the filtering methodology could allow for updates, as supported at the end of Subsection 4.3.3.) We divide the user study in two parts: participants will sequentially fill out (1) a survey and (2) an experiment during which our filtering methodology is tested. We run our filtering methodology based on a user profile that is built using the gathered survey answers.

In Section 5.1, we introduce the details of the study setting. After that, we explain how we pre-train our recommender in Section 5.2. Then, we cover the survey and experiment design in Section 5.3 and Section 5.4, respectively. Finally, Section 5.5 presents Exploration Explorer – the realisation of the survey and experiment.

5.1. User Study Setting

Our user study focuses on music taste broadening recommendation. Participants of the study are users of Muziekweb, which we presented in Section 1.7. In short, Muziekweb is the Dutch national music library that focuses on encouraging music taste broadening and has an online platform through which users can explore their music taste and borrow music albums.

From an internal online customer survey, conducted with 691 participants in November 2019, several interesting data skews emerged. We take these skews into account for our study. The survey was filled out by 691 participants. The following enumeration summarizes relevant demographics related findings. Moreover, 413 out of 529 participants indicated to borrow music from the library. Therefore, the presented demographics information is highly indicative for users that borrow music.

- 80% are male, 14% female, and the rest preferred not to say.
- 82% are at least 50 years old (5% preferred not to say). More specifically, the age of 27% is between 50 and 59, 38% is between 60 and 69, and the remaining 17% is at least 70.
- 60% hold a higher education degree (9% preferred not to say).

As introduced in Section 1.7, for online music taste broadening recommendation, Muziekweb employs mosaics as result overviews. These are a compact way to suggest albums and are also used by large commercial music platforms, like Spotify. Moreover, mosaics help us to put our explore-exploit item placement strategies to an interesting test. Therefore, we have decided to implement mosaics for recommendation in our study, as visualized in Figure 5.1.

5.2. Pre-Training

To pre-train the recommender system used in the experiment, Muziekweb allowed us access to anonymized borrower data of 1,956 albums that were entered into their database between January and August 2018, and have been borrowed at least thrice. The data consists of 17,528 borrowing records, traced back to 2,721 unique borrowers. As borrowing an album requires the payment of a fee and considers a physical transaction, we assume that users will only borrow albums of which they have strong expectations they will like them. Thus, borrowing behavior may be indicative of a user's comfort zone. Therefore, we use the data for pre-training the exploitation algorithm of the recommender. Our exploration algorithm does *not* require any pre-training.

For each genre, we pre-train a finite-armed stochastic MAB with our dataset's borrowing records for that genre, to recommend an album based on a target album. For the bandit, the context is the target album, the arms are the albums in the item collection, the reward function equals the familiarity score function, and the arms are updated after each recommendation. We define the familiarity score function as:

$$\text{familiarity}(s,t) = n(s) + \mathbb{1}_{n(s,t)\geq 1} \cdot (m + n(s,t)) + \mathbb{1}_{s=t} \cdot m,$$

where, from the genre's collection, $n(\cdot)$ computes the number of times its input albums appear together in each borrower's record and m is the maximum number of times an album has been borrowed. As we compute familiarity using each borrower's records, this means that the policy that the MAB learns corresponds to exploitation for a 'prototypical genre fan'. We use the random exploration method ϵ-greedy (see Subsection 3.2.4) to train it, until the policy converges, which mostly happens between $1,000$ and $5,000$ iterations with $\epsilon = 0.1$. We assume that the policy converged if it does not change during 150 consecutive iterations.

5.3. Survey Design

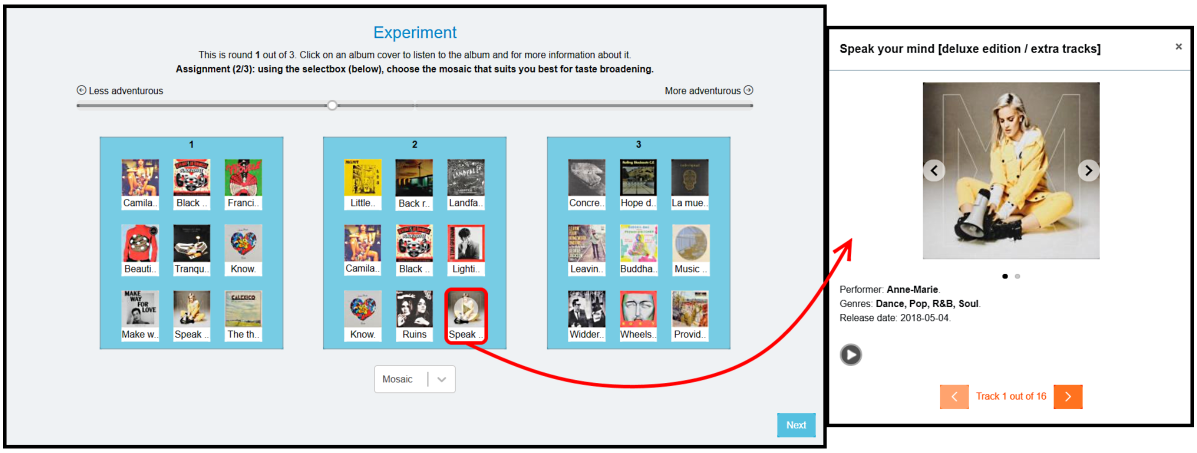
Through the answers of the survey, we build a user profile. The aims of the survey are twofold, namely to gather information to (1) take into account any participant profile skews during evaluation and (2) make item exploration preference predictions for the experiment. The questions are ordered in the same manner. Firstly, we ask the participants for their age, gender, nationality, country of current residency, and country of formative years. Secondly, we pose questions to determine the user variables covered in Chapter 4 explicitly.

More specifically, given a preferred category, we estimate the item exploration preferences formally introduced in the previous chapter for our filtering methodology. That is, explore-exploit item placement p, exploration rate e, category familiarity f, and exploration willingness w. Table 5.1 shows the three relevant questions. The multi-select options of questions 1a. and 1c. are derived from our borrower data covered in the previous subsection. The top 40 most occurring genres, of which we have at least 30 albums, are presented to the participants to choose from. We also allow participants to add missing genres to their list. For each of the picked genres from the top 40, participants choose their familiarity and, if at least rather familiar (2 on the Likert scale), pick at least one familiar artist from our data's musicians of the genre.

We normalize the filled out 5-point Likert scale score with the following formula: $n_5(s) = \frac{s-1}{4}$. We compute category familiarity f as: $f = n_5(s_{1b})$, where the subscript of s corresponds to the question number of Table 5.1. A 9-point scale that ranges from -1 to 7 is used for question 2d. to determine the personal value 'openness to change', as introduced by Schwartz [74]. Likewise, we normalize the 9-point scale score with the following formula: $n_9(s) = \frac{s}{7}$. To predict exploration rate e and exploration willingness $\hat{w}$, we average the filled out scores of the two particular questions, as these positively correlate. That is, $e = \frac{n_5(s_{2a})+n_9(s_{2b})}{2}$ and $\hat{w} = \frac{n_5(s_{2c})+n_9(s_{2d})}{2}$, respectively, where $\hat{w} \in [-1,1]$ is our w prediction. The multiple choice options of question 3. are limited to 'At the beginning', 'Around the middle', 'At the end', and 'No specific preference', for which we assign the values 0, 0.5, 1, and `null` to explore-exploit item placement p, respectively. If p is `null`, we apply a uniform distribution. Based on p and e, we generate an explore-exploit ranking strategy. Besides, based on f and $\hat{w}$, we determine where in the FP to initially draw exploration items from.

Table 5.1: User Preference Related Survey Questions

No.		Assignment/Question/Statement	Format	User variable
1	a.	Which music genres do you prefer?	Multi-select	Preferred category
	b.	For each of your given music genres: how familiar are you with the genre?	5-point Likert	Category familiarity f
	c.	For each of the genres you are minimally 'Rather familiar' with: select at least 1 artist you know already	Multi-select	-
2	a.	I browse on a recommender system, like Youtube or Netflix, actively for something new with the following frequency:	5-point Likert	Exploration rate e
	b.	In the same context as before, if I stumble upon something new, then I try it	5-point Likert	Exploration rate e
	c.	In general, I see myself as someone who is curious about many different things	5-point Likert	Exploration willingness $\hat{w}$
	d.	As a guiding principle in my life, I rate 'being curious' to be a personal value on a nine-point scale as	9-point scale	Exploration willingness $\hat{w}$
3.		In the context of taste broadening: given a list of recommendations, where do you prefer the emphasis of taste broadening items to be?	Multiple choice	Item placement p



Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (2/3): using the selectbox (below), choose the mosaic that suits you best for taste broadening.

Speak your mind [deluxe edition / extra tracks]

Performer: **Anne-Marie**.
Genres: **Dance, Pop, R&B, Soul**.
Release date: 2018-05-04.

Figure 5.1: An Impression of the Album Recommendation Experiment.

5.4. Experiment Design

In this section, we introduce the experiment design. The experiments are composed of three assignments and has three rounds. Our aims are to answer our research questions. In Section 5.4.1, we introduce the assignments. Section 5.4.2 then presents the trained MAB that we have implemented. After that, Section 5.4.3 covers limitations of the experiments. Finally, in Section 5.4.4, we cover our hypotheses that we defined beforehand. For the sake of completeness, the following enumeration refines the research questions defined in Chapter 1, based on the user preference variables that our filtering methodology uses.

RQ1 To what extent do users prefer recommendations that take into account user preferences that consider:

- explore-exploit item placement;
- exploration rate;
- exploration willingness; and
- category familiarity?

RQ2 How do the preferences of **RQ1** individually contribute?

RQ3 Where do users draw the line between items fit and unfit for their taste broadening from our recommendations?

5.4.1. Assignments

In each round, three album mosaics are presented, as visualized in Figure 5.1, which are produced by different methods and ordered randomly. Participants can listen to an album by clicking on it; this opens a modal that contains the album information and songs. The chosen genres depend on the number of indicated preferred genres in the survey. If one genre is given, it is used for each round. Else, if two are given, either one of them is assigned twice. Else, three unique genres are picked at random. In the three rounds, we separately test the performance contribution of the explore-exploit item placement p, exploration rate e, and category familiarity f, respectively. The performance contribution of the exploration willingness w is tested through the adventurousness slider at the top of the page.

Participants sequentially complete the following assignments:

A1 Using the slider, choose the adventure value in the blue area that suits you best for taste broadening.

A2 Using the selectbox below, choose the mosaic that suits you best for taste broadening.

A3 Select the albums that you find unfit for taste broadening.

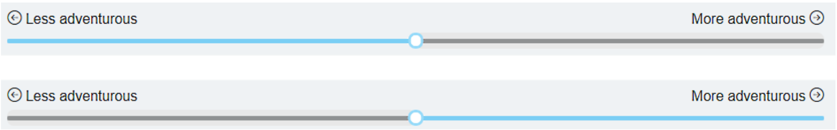


Figure 5.2: The Two Different Slider Settings.

Assignment 1

In **A1**, the participant can tune the recommendation adventurousness. The value at the center of the slider corresponds to our exploration willingness prediction $\hat{w}$. The slider's value is initialized at its center and, depending on the round, can either be moved to the left or right. Figure 5.2 illustrates this. More specifically, in round 1 it can only be moved to the left (the upper setting), round 2 only to the right (the lower setting), and round 3 to either side at random. This procedure encourages participants to look for an adventurousness value away from the center, to both the left and right of it. Also, it helps to answer **RQ2**, since results that are consistent over the rounds and have low variance over the participants likely indicate a high contribution. Given adventurousness value $n \in \{0, 1, \ldots, 100\}$, we compute the exploration willingness as: $w = \frac{2n}{100} \cdot \hat{w}$.

Assignment 2

In **A2**, the participant chooses one mosaic out of three that suits best for his/her taste broadening. These are the following mosaics:

- Our *prediction mosaic*. It is generated using our methods and the determined user variables.
- A *mutation mosaic*. It is generated like the prediction mosaic, except for the fact that exactly one of the determined user variables is mutated beforehand. A variable is mutated in the following way. The relatively large value changes help to determine the user variable's contribution.
  - If the original value is left of its range's center, assign the range's maximum value.
  - Else if the original value is right of its range's center, assign the range's minimum value.
  - Else, assign either extreme at random.
- A *random mosaic*. It contains randomly drawn items from the category's item collection.

The mutation mosaics help to answer **RQ2**, as results that show the prediction mosaic is generally picked over the mutation mosaic for the same mutated variable demonstrate the contribution of the individual user preference. As for the mutation types, we mutate in round 1 explore-exploit item placement, round 2 exploration rate, and round 3 category familiarity. In general, this assignment helps to answer **RQ1**, as results that show the prediction mosaic is generally picked over the others demonstrate the usefulness of our methods.

Assignment 3

In **A3**, the participant labels the albums unfit for his/her taste broadening of the mosaic chosen in **A2**. This assignment serves as a sanity check and helps to answer **RQ3**. If the mutation or random mosaic is chosen, we determine which labeled albums are outside of the FP's taste broadening subspace. Else, we determine where users draw the line for user-aware recommendation.

5.4.2. Trained Multi-Armed Bandit

After pre-training, which was discussed in Section 5.2, we construct the experiment's MAB. This MAB adopts the pre-learned policies of each genre for its exploitation algorithm. These policies correspond to exploitation for a 'prototypical genre fan' and are used by the experiment's MAB to determine exploitation items for an individual participant. For the sake of completeness, we do not know whether any borrowers from the training data are participants of the experiment. As a sanity check, we will check whether our sample is representative in the next chapter. Furthermore, the experiment's MAB implements our proposed algorithms for explore-exploit item placement strategy generation and FP-based exploration. For our proposed algorithms, we require predictions of the item exploration preferences and do *not* require pre-training.

We adopt the familiarity score function that was defined in Section 5.2. Recall that it was defined as:

$$\text{familiarity}(s,t) = n(s) + \mathbb{1}_{n(s,t)\geq 1} \cdot (m + n(s,t)) + \mathbb{1}_{s=t} \cdot m,$$

where, from the genre's collection, $n(\cdot)$ computes the number of times its input albums appear together in each borrower's record and m is the maximum number of times an album has been borrowed. Our familiarity score function guarantees that the beginning of each FP will contain several items that are rather familiar, if not familiar, while the end will contain relatively many albums that are irrelevant to the target and borrowed only a few times. Therefore, we set familiarity score range δ to 0.

In the experiment, the target albums are the albums of the indicated familiar artists in the survey. Each time the bandit explores, it picks a random target from the target albums and then initializes the user's FP. We cannot pre-initialize FPs, as we do not know the target albums provided by the user beforehand, which is required by our familiarity score function. However, this way of defining the familiarity score simplifies its definition for the experiment, as is reflected by the fact that we can set familiarity score range δ to 0. For this experiment, our familiarity score function suffices to measure the effect of including the user's exploration preferences during one recommendation 'time step'.

As for our remaining hyperparameters, based on an analysis of the skew-normal distribution, we set $\alpha_1 = 3$, $\alpha_2 = 2$, $\beta_1 = -1.5$, and $\beta_2 = 0$. We set $\beta_3 = 0.05$ to slightly encourage taste broadening. Lastly, for ease, we set $\alpha_3 = \alpha_4 = 1$ and $\beta_4 = 0$. Our pilot study, which is presented in the next chapter, serves as a sanity check.

5.4.3. Limitations

There are two experiment limitations that we consider to be important. The first one is related to the decision that we set the number of rounds of the experiment to three. Like there is a round for the explore-exploit item placement, exploration rate, and category familiarity, we could also have included a fourth round for exploration willingness. We decided not to, because the adventurousness slider allows for evaluation of this user variable's contribution, as explained in Section 5.4.1. Also, our collaboration partner, Muziekweb, advised to not make the experiment lengthy, especially because users participate completely voluntarily. Trivially, the risk of a lengthy experiment is that participants may not find the motivation to complete it. However, if we take this decision into account for **RQ2**, we can not analyze the relative contribution of exploration willingness as precise as we can do with the other user variables.

The second limitation is related to hyperparameter tuning. In Section 2.2, we explained that we do not want to optimize against a single ranking criterion. Instead, as explained in Section 2.4, we want our methodology to find a balance that suits the user in terms of the exploration-exploitation trade-off and measure the effect of the introduced user awareness. This decision makes hyperparameter optimization more difficult and rather time-consuming, as the optimization requires human-oriented involvement. However, we hypothesize that the hyperparameters are not sensitive to small value changes. Confirmation would allow us to relax the optimization. In the next chapter, the results of a post hoc hyperparameter evaluation are presented to verify this.

5.4.4. Hypotheses

With respect to the research questions, we hypothesize beforehand that:

RQ1 To what extent do users prefer recommendations that take into account user preferences that consider:

- explore-exploit item placement;
- exploration rate;
- exploration willingness; and
- category familiarity?

H1 Overall, users will prefer mosaics based on our personalized recommendations.

RQ2 How do the preferences of **RQ1** individually contribute?

H2 The integration of each user preference variable will contribute positively.

RQ3 Where do users draw the line between items fit and unfit for their taste broadening from our recommendations?

H3 For mosaics with our personalized recommendations, fewer items will be marked unfit. Also, the items marked as unfit will mostly be outside of the FP's taste broadening subspace.

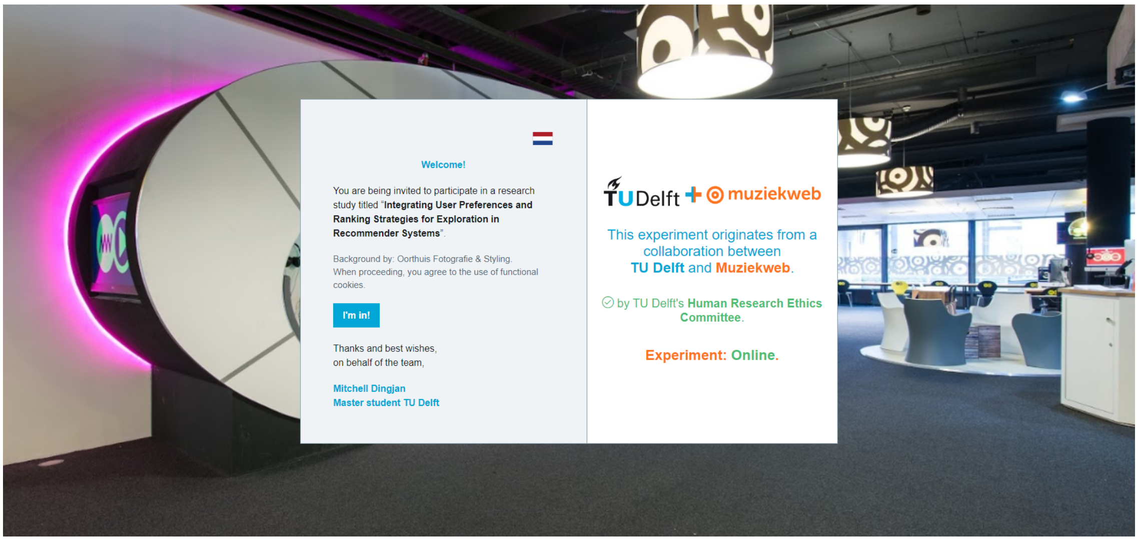
5.5. Experiment Realisation: Exploration Explorer

In this section, "Exploration Explorer" is presented, which is the name of the experiments website that we have developed for the thesis. In Subsection 5.5.1, the website is introduced. After that, Subsection 5.5.2 discusses the design choices that have been made. The flow of the experiment data is then explained in Subsection 5.5.3. Finally, we discuss ethical implications of our research for participants and the measures that we have taken in Subsection 5.5.4. Moreover, supplementary to this section, Appendix C covers the technologies that have been used to set up the website and Appendix D includes a walkthrough of the website.

5.5.1. Introduction

We conduct our thesis experiments online through Exploration Explorer. Figure 5.3 provides an impression of the website. We refer to Appendix D for an impression of the entire website. The visitors (i.e., participants) are primarily Muziekweb users. Because Muziekweb is internationalizing and to allow for a wide audience, the website is available in both Dutch and English. Aside from the possible languages that visitors are comfortable with, we do not make any further visitor related assumptions. In other words, the website has to be understandable and comfortable to everyone, regardless of, for instance, their age or education degree.

Aside from a welcome and closing page, the website consists of three parts, namely an informed consent, survey, and experiment. Because the experiment involves on-demand audio previews, the website is desktop only. We zoom in on the informed consent in Subsection 5.5.4. The goal of the survey is to gather user data that can be used in the experiment to make predictions. Through the experiment, we gather data to evaluate the methods that we introduced in the previous chapter. Our website supports these goals by providing participants with an intuitive website design and guiding them step-by-step through the process with clear instructions. The most challenging part for obtaining the website's goal was the experiment, as we will see in Subsection 5.5.2.



Welcome!

You are being invited to participate in a research study titled "**Integrating User Preferences and Ranking Strategies for Exploration in Recommender Systems**".

Background by: Oorthuis Fotografie & Styling.
When proceeding, you agree to the use of functional cookies.

Thanks and best wishes,
on behalf of the team,

Mitchell Dingjan
Master student TU Delft

This experiment originates from a collaboration between **TU Delft** and **Muziekweb**.

✓ by TU Delft's **Human Research Ethics Committee**.

Experiment: Online.

Figure 5.3: Exploration Explorer's Landing Page (English Version).

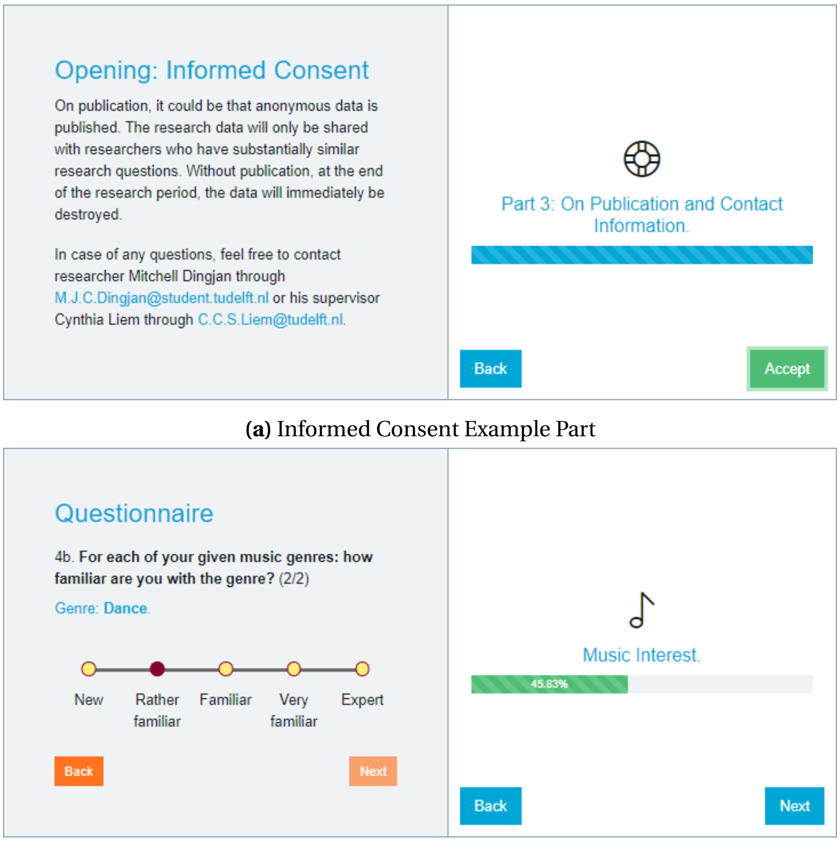
5.5.2. Frontend Design Choices

Because we work with React, as explained in Appendix C, for the sake of simplicity, we have decided to implement only one page and update its Document Object Model (DOM) as the user progresses. This unburdens the client and server of more page related communication. Furthermore, it prevents the possibility of users navigating to parts of the website that they should not (yet) access. Also, upon receipt of the page, it allows users to work offline until data is needed from the server for the experiments; thus, an internet disconnection will do no harm to the user's progress. Therefore, it also simplifies the data flow, as we will elaborate on in Subsection 5.5.3.

Also, we allow users to navigate back to adjust their answers until the experiment starts. Figure 5.4 gives an impression of the navigation buttons. After that, their data gets processed for recommendation and we will not allow them to return, because it adds complexity and may interfere with the results. We mention this clearly on the website before proceeding to the experiment.

Moreover, we have decided to deliver information step-by-step to the participant to prevent confusion and keep the information load easy to digest. Before proceeding to the survey and the experiment, we also prepare the participant for what is to be expected. Figure 5.4 illustrates the small information loads by means of examples. The informed consent is split into three parts to make sure the user has the opportunity to read the different subjects and lower the information load, as Figure 5.4a shows. In the survey, we only pose one question at a time. Figure 5.4b illustrates this. Also, it illustrates that subquestions can be navigated to through the orange navigation buttons, at the left part of the screen. In the experiment, we pose one question at a time as well. Figure 5.4c illustrates this. We disable, instead of hide, the few elements that are irrelevant for the active experiment instruction (i.e., the slider or select box) so that users can take them into account.

Furthermore, to inform the users about their progress, we have implemented progress bars in the informed consent and the survey. We also present the current round and assignment at the top of the page during the experiment. These progress trackers can be seen in Figure 5.4. The informed consent also informs the users about how much time is approximately required to complete their submission.



Opening: Informed Consent

On publication, it could be that anonymous data is published. The research data will only be shared with researchers who have substantially similar research questions. Without publication, at the end of the research period, the data will immediately be destroyed.

In case of any questions, feel free to contact researcher Mitchell Dingjan through M.J.C.Dingjan@student.tudelft.nl or his supervisor Cynthia Liem through C.C.S.Liem@tudelft.nl.

Part 3: On Publication and Contact Information.

(a) Informed Consent Example Part

Questionnaire

4b. For each of your given music genres: how familiar are you with the genre? (2/2)

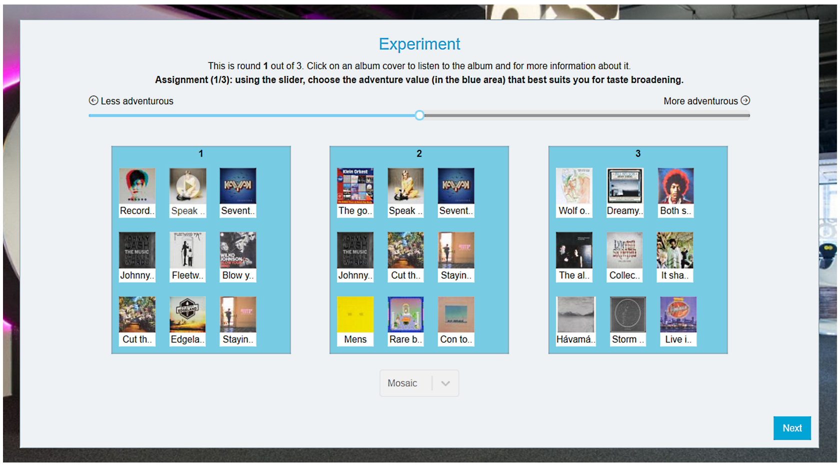
Genre: Dance.

○ New ● Rather familiar ○ Familiar ○ Very familiar ○ Expert

Music Interest.

(b) Questionnaire Example Part

Figure 5.4: Easy to Digest Information Loads.



Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (1/3): using the slider, choose the adventure value (in the blue area) that best suits you for taste broadening.

(c) Experiment Example Part

Figure 5.4: Easy to Digest Information Loads.

5.5.3. Data Flow

The fact that the website consists of only one page, simplifies the data flow. Also, we do not save any data to the database until the experiment is completed, because we only want to accept complete submissions. A sequence diagram of the data flow is given by Figure 5.5, where n equals the number of slider changes plus the number of rounds. If the participant navigates to the Exploration Explorer, the server sends the webpage to the client. The participant then has the time to complete the informed consent and the survey. During the experiment, the client and server will communicate to send the mosaics to the client for each of the three experiment rounds. At the beginning of each experiment round and for each slider change, the client sends a POST request to the server that includes the data from the survey. If the participant completes the experiment, the gathered survey and experiment data is sent through a POST request to the server, which in turn saves the data to the database. Aside from the information given by the participant, this data includes for evaluative purposes the generated mosaics, round genres, and duration of filling out each part. From the data, we can reconstruct the explore-exploit item placement strategy and the FP of each round for analysis.

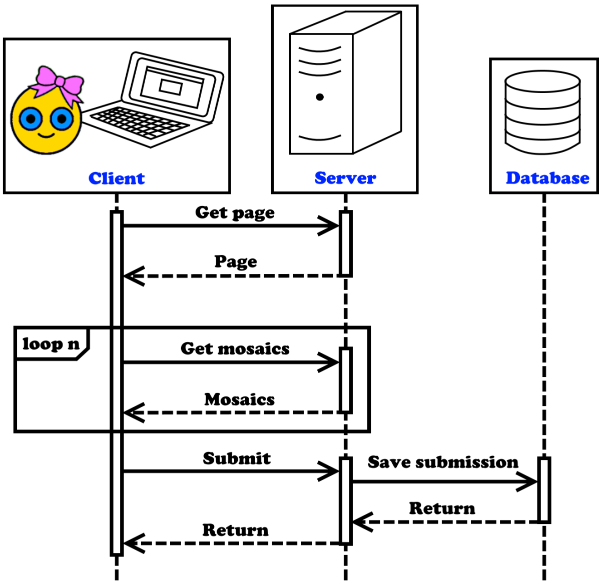


Figure 5.5: Sequence Diagram of the Communication Between Client, Server, and Database.

5.5.4. Ethical Implications

As with every online related activity, the risk of a data breach is always possible. That is why there are is an ethical implication related to participation. Our research has been approved by TU Delft's Human Research Ethics Committee, which implies that, according to them, the measures we have taken to limit the ethical implications are sufficient. We refer to Appendix E for the HREC approval. To obtain this approval, we informed them about our experiments in detail and submitted a Data Management Plan, as well as a draft version of the informed consent.

To minimize the risks of a breach, (1) the client and server communicate through an https certified website, (2) we process the data on a private server connected to a private database, and (3) we store back-ups on a TU Delft assigned private storage. We informed the participants that, on publication, it could be that anonymous data is published. We only share the research data with researchers who have substantially similar research questions. Without publication, at the end of the research period, we will immediately destroy the data.

(This page has been intentionally left blank.)

6

Results and Discussion

In this chapter, we present the experiment results. Section 6.1 and 6.2 present the results of our pilot and live study, respectively. These studies sequentially ran during October and November 2019. We then discuss the results in Section 6.3. Finally, Section 6.4 provides a post hoc hyperparameter evaluation.

6.1. Pilot Study

The aims of the pilot study are to check whether the instructions in the online user study are clear to participants, the UI design is intuitive, and the overall functionality works as planned in a real-user setting. For the study, we recruited eight participants. These people were excluded from the live experiments. Together with each participant, we went through the survey and experiment. We asked the participants beforehand to think aloud, to enable us to follow their thought process. They could only pose us urgent questions.

Based on the outcomes of the user profile study presented in Subsection 5.1, we consider our sample to be representative. The following enumeration summarizes the participant profiles:

- 75% are male and 25% female.
- the average age is 46 and 50% are at least 50 years old.
- 62.5% hold a higher education degree.

During the evaluative talks at the end of the sessions, the participants consistently confirmed that they broadened their taste. Each participant found artists or songs that they recognized, if not vaguely recognized, and based on these, successfully found a mosaic that suited them best. To quote one participant, "I learned to appreciate new songs of artists I did not know, who are close to my favorite artists." Overall, we received positive feedback.

6.2. Live Study

35 users participated in our live study. Based on the existing customer statistics of the user profile study presented in Section 5.1, we consider this sample to be representative. Although no education degree data was gathered, we have the following profile statistics:

- 86% are male and 14% are female.
- the average age is 50 and 66% are at least 50 years old. More specifically, the age of 23% is between 50 and 59, 32% is between 60 and 69, and the remaining 11% is at least 70.
- 94% live in the library's country and 6% live in a neighboring country.

Table 6.1 shows the means and Standard Deviations (SDs) of the user variables deduced from the survey. Recall that exploration willingness is defined on $[-1,1]$, whereas the other user variables are defined on $[0,1]$. Participants picked a remarkably large number of preferred genres and familiar artists. Moreover, there is no consensus on the preferred explore-exploit item placement.

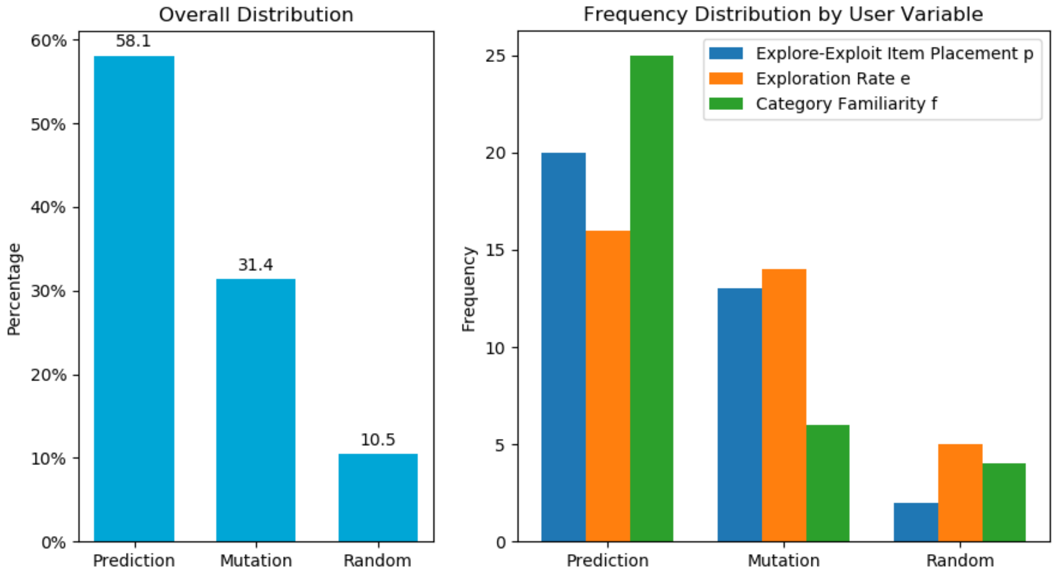
We would like to emphasize that participation was completely voluntary. By this, we mean that no rewards to encourage participation were attached, as monetary incentives may diminish the quality of the responses. On average, participants spent 45.2 minutes ($SD = 56.0$) on their response. Their committed participation is also visible from Table 6.1, which shows a high number of indicated genres and artists. Therefore, participants put much effort in doing the experiments, which resulted in a valuable dataset.

Based on our training data, on average, users borrow albums from 6.1 different genres ($SD = 5.5$), filtered on our genre top 40. If we compare this average with the average number of preferred genres from Table 6.1, we find that these are close. The strengthens the confirmation that the user sample is representative. In Section 5.1 we stated that the albums borrowed are probably close to, if not inside, the customers' comfort zones, because of borrowing costs. The fact that these averages are close confirms that the training data should be used for exploitation, in taste broadening recommendation. Moreover, on average, the borrowed albums per user are from 5.9 different artists ($SD = 12.0$). Table 6.1 shows a larger average, since the number is related to familiarity and familiarity does not imply preference.

Table 6.1: User Variable Statistics

Variable	Mean	SD
Number of preferred genres	5.5	4.4
Category familiarity f	0.59	0.18
Number of familiar artists per genre	9.2	10.7
Exploration rate e	0.52	0.25
Exploration willingness $\hat{w}$	0.19	0.14
Explore-exploit item placement p	0.53	0.43

Relating to **RQ1**, we analyze the experiment results visualized in Figure 6.1a. We find that our prediction mosaic was significantly preferred, $\chi^2(2, N = 105) = 35.89, p < .001$. Furthermore, relating to **RQ2**, we analyze the results visualized in Figure 6.1b. We find that our prediction mosaic was significantly preferred in the explore-exploit item placement, $\chi^2_p(2, N = 35) = 14.11$, and category familiarity, $\chi^2_f(2, N = 35) = 23.03$, mutation rounds, $p < .001$. However, this does not hold for the exploration rate mutation rounds, $\chi^2_e(2, N = 35) = 5.89, p \approx .053$.



(a) Overall

(b) Individual Preferences

Figure 6.1: Histograms of the Chosen Album Mosaics

Relating to **RQ3**, Table 6.2 shows statistics of the albums that users marked as unfit for their taste broadening. For our prediction mosaics, all exploration items are strictly from the FP's taste broadening subspace. This does not hold for the other mosaic types. The familiarity border distance is measured as the absolute difference between the FP indices of the (unfit) item and familiarity border b. The relatively high distance SDs for the prediction and mutation mosaics are mostly caused by the marked mix of extreme low (exploration) and high distance (exploitation) items. We use a Linear Model analysis to test the difference between the mosaic types on the number of marked unfit albums. We find a significant effect ($F(2, 102) = 12.35, p < .001$) for the mosaic types on the number of unfit marked items. Post hoc comparisons using the Tukey HSD test show that the random mosaics and each other group differ significantly at $p < .05$. However, they show no significant difference between the prediction mosaics and the mutation mosaics.

Relating to **RQ2**, we analyze the contribution of exploration willingness by studying the values that participants chose with the adventurousness slider. These are visualized in Figure 6.2. We find that the participants significantly preferred adventurousness values that are relatively more to the right ($M = 55$, $SD = 17.4$) of the slider's center, $t(104) = 32.53$, $p < .001$. We find that the values on the same side of the slider are consistently chosen by the participants over the rounds ($M = 3.62$, $SD = 5.02$).

Table 6.2: Items Marked as Unfit Statistics

Mosaic	Mean	SD	Exploration Items (%)	Familiarity Border b Distance Mean	Familiarity Border b Distance SD
Prediction	1.16	1.15	71.9	5.33	7.15
Mutation	1.79	1.77	70.4	8.59	7.58
Random	3.54	1.92	-	10.12	5.65

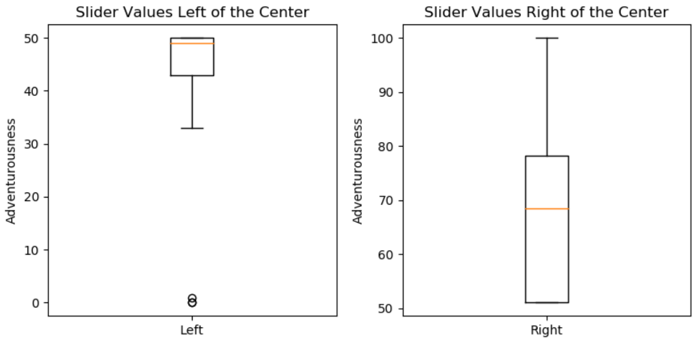


Figure 6.2: Boxplots of the Adventurousness Values.

6.3. Discussion

We find that our hypotheses, as defined in Subsection 5.4.4, are mostly in line with our results. As for **RQ1** and **RQ2**, we found that our mosaic is overall, and in two out of the three types of mutation rounds, significantly preferred. We expected the exploration rate to be the least contributing user variable, as it affects the content of the recommendation least, namely only through the exploration probability. Also, we expected the genre familiarity to be the most contributing variable, as it directly influences the connection between the user and the recommendation content. However, although we found that our mosaic was significantly preferred in the explore-exploit item placement mutation rounds, we expected the influence of this user variable to be bigger. One reason for this is the fact that five participants indicated to not have a specific item placement preference.

Based on Figure 6.2, participants significantly preferred values to the right of the slider's center. An influential factor that could explain this, is the fact that it is natural for humans to be curious about what is beyond the default value, especially in the context of adventurousness. This implies that participants will anyway choose a value that is relatively right to the slider's center. As we also found that participants chose adventurousness values that are consistent over the rounds, we conclude that the exploration willingness contributes positively to our recommendations as well.

As for **RQ3**, we found that for our recommendations, participants labeled significantly fewer items as unfit. Table 6.2 shows that, on average, approximately one item is marked as unfit for our recommendations. An influential factor could be that participants have the urge to mark at least one item as unfit – the item that least suits their taste broadening. Also, pilot study participants marked items from our mosaic as unfit mostly by comparison with the other albums inside the mosaic. This suggests that participants look in our mosaics for the item that relatively least contributes to their taste broadening. The table also shows a relatively high SD for the mutation mosaics. These deviations mainly result from the category familiarity mutation rounds, which show a relatively large number of unfit marked mosaics ($M = 4.60$, $SD = 1.36$). The exploration items of these rounds also have an increased distance to the actual familiarity border because of their mutation, which is reflected by the table. The unfit items from the random mosaics show a relatively high mean distance and a low distance SD, as the items come from anywhere in the FP, in contrast to the other mosaics that are produced from subspaces. Moreover, we found that items outside of the FP's taste broadening subspace are less likely to be fit for taste broadening. However, based on this research, we cannot give a precise answer of where to draw the line between items fit and unfit for taste broadening in the FP itself. To advance to this level, the question deserves separate qualitative research.

6.4. Post Hoc Hyperparameter Evaluation

In Section 5.4.3, we argued that a stability evaluation of the hyperparameters is needed. We perform an evaluation for each hyperparameter in the following way. We start from our initial hyperparameter settings (see 5.4.2) and then first repeatedly increase the value of the hyperparameter by .05. Afterwards, we reset the value and decrease its value by .05 repeatedly. At each value change, we observe the changes in the mosaic.

We find that small changes to hyperparameters α_1, α_2, β_1, and β_2 are hardly noticeable to the user, because of the probabilistic nature of our explore-exploit item placement strategies. Small changes to familiarity score range δ are hardly noticeable too, as such changes yield small shape changes to the FP. The effect is limited by the FP length-based computation of familiarity border b and exploration range r (see Algorithm 10). Therefore, the presented exploration items could be from the neighboring, if not from the same, index of the original FP. These items could still be fit for taste broadening, because they are at least close to, if not in, the original taste broadening subspace.

In contrast to the previous hyperparameters, small changes to remaining hyperparameters α_3, α_4, β_3, and β_4 could directly change the size and position of the FP's taste broadening subspace. Such small changes likely lead to a slightly increased average distance of the exploration items to the original familiarity border. Based on the results presented in Table 6.2, our experiments demonstrate that such mutations could lead to an increased number of items that are marked as unfit for taste broadening. Also, Figure 6.1 shows that such mosaics could be less preferred. Therefore, the performance is relatively more prone to changes of these hyperparameters. However, as explained in Section 5.4.1, our mutation mosaics include extreme changes. As the changes remain small, their effect on the performance remains small as well.

7 Conclusion

In this chapter, we wrap up the thesis report. Section 7.1 answers our research questions. Section 7.2 then summarizes the contributions of the thesis. Finally, Section 7.3 covers promising future work directions.

7.1. Conclusions

In this thesis, we introduced explore-exploit item placement strategies and, in the context of taste broadening, a user-aware exploration algorithm. These filtering techniques integrate user variables with respect to explore-exploit item placement, exploration, and familiarity to answer the questions of "Where?", "How much?", and "For whom?" to let the recommender explore, respectively. Based on the derived time-complexities in Chapter 4, our algorithms are time-efficient by definition, which is of great value for recommendation.

To conclude our research, we recap our research questions, hypotheses, and findings.

RQ1 To what extent do users prefer recommendations that take into account user preferences that consider:

- explore-exploit item placement;
- exploration rate;
- exploration willingness; and
- category familiarity?

H1 Overall, users will prefer mosaics based on our personalized recommendations.

C1 We accept **H1**, based on Figure 6.1a and the results of the corresponding significance tests described in Section 6.2.

RQ2 How do the preferences of **RQ1** individually contribute?

H2 The integration of each user preference variable will contribute positively.

C2 We accept **H2**, based on Figure 6.1b, Figure 6.2, and the results of the corresponding significance tests described in Section 6.2. Moreover, based on our variable mutation approach, category familiarity contributes most and exploration rate contributes least.

RQ3 Where do users draw the line between items fit and unfit for their taste broadening from our recommendations?

H3 For mosaics with our personalized recommendations, fewer items will be marked unfit. Also, the items marked as unfit will mostly be outside of the FP's taste broadening subspace.

C3 We accept **H3**, based on Table 6.2 and the results of the corresponding significance tests described in Section 6.2. For a more precise answer that, for instance, reflects the user's motivations for marking items as unfit from the taste broadening subspace, the question deserves separate qualitative research.

Therefore, the experiments demonstrate the effectiveness of our proposed user-aware methods in a taste broadening recommendation setting. We hope that this work serves as an eye-opener to trigger more research to be centered around exploration in recommender systems. As shown, there is more ground to be gained.

7.2. Contributions

This thesis contributes to the recommender systems research domain. Our main contribution is the fact that we allow users to have a say in how to let the recommender explore, based on item exploration preferences, through both new filtering methodology and an interactive user interface. This contribution is especially useful for taste broadening recommendation, since exploration is fundamental in this context. Our introduction of an interactive user interface to influence the algorithmic exploration process is also relevant for interactive IR. The following enumeration summarizes our most important contributions.

- Our explore-exploit item placement strategy generation method is a new approach to introduce user awareness in recommendation item placement.
- Our FP-based exploration method is a new approach to introduce user awareness in exploration by the recommender system. It applies to taste broadening recommendation.
- In general, our filtering methodology is a new approach to personalize the algorithmic exploration process for recommender systems that work from the exploration-exploitation paradigm. The methodology extends a random exploration method with the integration of item exploration preferences.
- Our interactive user interface design is a new approach to allow the user to influence the algoritmic exploration process. Also, we demonstrated that users show willingness to invest effort through the user interface in personalizing exploration for taste broadening recommendation.
- The results of our user study demonstrate the effectiveness of our filtering methodology combined with the interactive user interface.

7.3. Future Work

In this section, we cover potentially interesting future work directions.

7.3.1. Beyond Where, How Much, and For Whom

In Section 1.5, we posed an interesting future work direction, namely to analyze item placement strategies and FPs over time. This raises the question “When?” to let the recommender explore. For item placement strategies, we could research the applications of exploration decay functions that deal with the cold-start problem. And for FPs, we could research familiarity decay functions that decrease the familiarity score of items over time. Moreover, the user model that includes item exploration preferences could change over time, which calls for (efficient) item placement strategy and FP updates. These future work directions could make the proposed methodology even more useful.

7.3.2. Collaborative Settings

In Subsection 3.2.2, we covered collaborative bandits, which try to find preference related patterns by means of clustering procedures. Methodology could be designed to introduce our methods in collaborative settings as well. For instance, explore-exploit item placement strategies and FPs could be compared across users to find user preference related patterns and exploit these to increase the use of our methods further.

7.3.3. More User-Aware Exploration Algorithms

In Section 3.2.4, we stated that there are three types of exploration strategies for MABs. Our user-aware exploration method extends the basic random exploration method ϵ-greedy with the integration of user preference variables. The strongest assumption that our methodology relies on, is that the explore-exploit item placement algorithm probabilistically decides to explore or exploit. Since we do not make any strong assumptions on the random exploration method itself, other random exploration methods can be extended to work with our methodology, like Exp3 and Exp4.

Moreover, uncertainty-based exploration methods may be extended to introduce user awareness. An interesting future work direction would be to translate our methodology for such exploration methods, like UCB and TS. This requires some more effort than for random exploration methods, due to our previously mentioned strongest assumption. The performance of user-aware versions for both other random exploration methods and, if such a translation is found, uncertainty-based methods, could then be compared with our findings.

7.3.4. Processing Recommendation Feedback

In Chapter 4, we invested work in making our methodology open for extension with respect to recommendation feedback. The last paragraph of Subsection 4.3.3 illustrates this. Although we did not dive deeper into this to limit the scope of our research, we do acknowledge the importance and potential benefit of updating the explore-exploit placement strategy and FP after each recommendation. For example, if the user has listened to a recommmended music album, the item position in the FP should be updated accordingly. Our algorithms could be extended to handle such updates and experiments could be designed to demonstrate their potential benefits.

7.3.5. Exploring FP Data Structures and Placement Probability Distributions

In Subsection 4.3, we introduced an FP to be a list of lists that by means of a familiarity score function fans out in a 2-D conelike shape. This feels like a very natural data structure to use for FPs. However, we could research the use of other data structures. The goal of that research is to find speed-ups for operations on the FP. Although we found that the operations are efficient by definition, based on the derived time-complexities, speed-ups are still interesting. As a starting point, tree-based FP implementations could be researched. For instance, we could analyze self-balancing search tree FPs, which may simplify operations.

We could also research the use of other probability distributions for explore-exploit item placement strategies. In Subsection 4.2, we decided to use the multivariate skew-normal distribution, which we demonstrated to work well. Additionally, we could ask ourselves: what other distributions would work well and why?

7.3.6. Qualitative Research

In Section 6.3 and 7.1, we stated that a qualitative research could be performed, based on our research. The goal of this research is to find a more precise answer to what motivations drive users to mark items as unfit from the FP's taste broadening subspace. A careful look at the users' motivations may lead to interesting insights to improve our methodology further.

7.3.7. Putting Knowledge Into Practice: Improving Muziekweb's Recommender

Since the findings of this thesis confirm the benefit of user-aware exploration and user-aware explore-exploit item placement for taste broadening recommendation, we could directly put our methodology into practice. Therefore, Muziekweb could directly act upon this by implementing the MAB of our experiments. As more research gets completed on the previously mentioned future work directions, the recommender may be improved even further. The current most important future work direction for practical applications is processing recommendation feedback (see Subsection 7.3.4), for which we have laid the foundations.

(This page has been intentionally left blank.)

A

Paper

This appendix contains the paper written during the thesis and submitted to SIGIR 2020[1], an international ACM Conference on Research and Development in Information Retrieval. For the submission, we had to preserve our anonymity.

[1] https://sigir.org/sigir2020/

Exploring Exploration in Recommender Systems: Where? How Much? For Whom?

Anonymous Author(s)

ABSTRACT

Recommender systems focus on automatically surfacing suitable items for users from digital collections that are too large for the user to oversee themselves. A considerable body of work exists on surfacing items that match what a user liked in the past; this way, the recommender system will exploit its knowledge of a user's comfort zone. However, application scenarios exist in which it is explicitly important to offer the user opportunities to explore items beyond their existing comfort zones. In such cases, the recommender should include items for which there is less existing evidence that the user will like them. This calls for the recommender to explore to what extent a user would be tolerant to exploration.

In this paper, we consider that different users will likely have different preferences with respect to item exploration. We propose personalized item filtering techniques for this, modeled under a multi-armed bandit framework, that consider (1) how and where exploration vs. exploitation items should be distributed in a result overview and (2) how adventurous exploration items can be, considering a user's general willingness to explore and existing familiarity with item categories in the collection. We present the results of a survey and an online quantitative experiment with 43 users of an anonymous public music library, demonstrating the effectiveness of both our proposed filtering techniques that aim for personalized exploration.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Personalization**; • **Human-centered computing** → **User studies**.

KEYWORDS

Recommender systems; Exploration; Personalization; Filtering; User preferences; Taste broadening; Multi-armed bandits

ACM Reference Format:
Anonymous Author(s). 2020. Exploring Exploration in Recommender Systems: Where? How Much? For Whom?. In *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Xi'an, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In many digital multimedia consumption scenarios, the size of digital collections has grown so large that automatic filtering methods have become indispensable to users. In particular, personalized recommendations are considered to be good means to get the right content to the right users. At the same time, concerns have arisen that too much content personalization may lead to filter bubbles [21, 22], confronting users with too narrowly scoped selections. As possible ways to mitigate this, the notions of diversity and novelty have become increasingly important in both the recommender systems and information retrieval (IR) communities [5, 13, 15, 26, 33].

Many commercial recommender applications may most obviously benefit from getting users the items they explicitly want, in a way that is as frictionless as possible. This criterion is reflected in common success metrics, such as engagement and click-through rates. At the same time, other scenarios exist in which a more serendipitous perspective is important, and the recommender system should assist users in broadening their horizons and making them discover items they did not yet know they liked. Such a perspective is increasingly becoming important to the positioning of public libraries. While online search engines may offer easy at-home access to a broader body of information, libraries can offer users trusted, curated and locally approachable ways to also engage with more specialized content in the long tail.

One domain in which discovering new content (and possibly, new tastes and interests) is natural, is music. However, a system that solely would provide items outside of a user's known comfort zone will likely hamper the user experience. Therefore, proposals have been presented to balance recommendations that exploit known user preferences with recommendations that explore what acceptable recommendations may exist beyond these known preferences [14, 20]. The exploration-exploitation paradigm has more broadly been proposed as a means to increase utility beyond relevance, both in the recommender systems and IR communities [2, 12, 32]. From a technical perspective, reinforcement learning paradigms and multi-armed bandits offer suitable models to tackle the corresponding exploration-exploitation trade-off [30].

In this paper, we focus on the challenge of exploration for taste broadening. Considering user-aware personalization, we consider that exploration preferences may consist of multiple facets. Focusing on where to physically distribute exploration vs. exploitation items, general exploration-oriented tendencies of the user, and category-specific user familiarity, we propose automated filtering techniques that customize where, how much, and for whom to explore. In an experiment in collaboration with a public music library, we focus on the following three research questions:

RQ1 To what extent do users prefer recommendations that take into account user preferences with respect to:
- explore-exploit item placement;
- exploration; and

SIGIR '20, July 25–30, 2020, Xi'an, China

ACM ISBN 978-1-4503-9999-9/20/07…$15.00
https://doi.org/10.1145/1122445.1122456

- category familiarity?

RQ2 How do the preferences of **RQ1** individually contribute?

RQ3 Where do users draw the line between items fit and unfit for broadening their horizon from our recommendations?

The rest of the paper is organized as follows. In the next section, we discuss related work. In Section 3, we propose explore-exploit item placement strategies and a user-aware exploration algorithm. We present our user study in Section 4 and 5, which is designed to answer the research questions above, using our proposed methods. We present and discuss our results in Section 6. Finally, we conclude our paper and present future work directions in Section 7.

2 RELATED WORK

2.1 Describing Users

In our work, we consider user preferences with respect to exploration. In general, the question of how to describe users to serve them for recommendation has been widely studied. By means of a literature review, Laplante shows in the music domain that there are associations between music preferences and demographic characteristics, personality traits, personal values, and beliefs [16]. By exploiting these for recommender system design, one could improve the prediction ability of recommenders [31]. Although each of the aforementioned human factors has been shown to be a valid construct to describe people in psychology [16], the focus has been so far on personality-based recommenders [19, 26].

A widely accepted theory to describe personality in psychology is the five-factor OCEAN model [7]. The five factors can be assessed explicitly, using surveys for which a five-point Likert scale is often used, or implicitly, by means of data analysis [6]. Using Netflix data, Golbeck and Norris are the first to apply the model to relate personality with movie rating and viewing history [8]. Cantador et al. use the model to study the relations between personality types and user preferences in movies, TV shows, music, and books [4].

Schwartz proposes a model to describe values [29]. Roccas et al. relate the factors to values [27]. They conclude that the relations found help to clarify some issues regarding the meaning of the factors, which were previously debated. Furthermore, their findings support the idea that the influence of values on behavior depends more on cognitive control than the influence of traits. Adamopoulos and Todri automatically infer personality traits, needs, and values from social media content and build different recommender system models [1]. Using Amazon data, they find that the under-explored models of needs and values result in an improved predictive performance when compared to the more popular five-factor model. Manolios et al. hypothesize that value-aware recommendation is especially of interest for broadening the user's horizon, as it challenges the user to look beyond what is intuitively preferable [19].

Based on these works, we conclude that certain traits and values relate to exploration, and thus will be relevant in assessing to what degree a user will be inclined towards exploration. More specifically, we will focus on the 'openness to experience' personality trait [7], and 'openness to change' personal value [29].

2.2 Explore-Exploit Item Placement

Although explore-exploit strategies are widely researched in sequential contexts, their integration in ranking approaches is almost an empty field, according to Guillou (2016) [9]. To date, not much has changed since then. The most notable work on personalized ranking is by Rendle et al., which presents from a Bayesian analysis a generic optimization criterion BPR-OPT for personalized ranking [25]. Also, it presents a generic pairwise learning algorithm for optimizing models with respect to BPR-OPT.

However, deep learning approaches have recently been applied to personalize ranking. He and McAuley work upon Rendle et al.'s work to propose Visual BPR, a scalable method that incorporates visual features extracted from product images into matrix factorization to uncover what visually most influences people's behavior [10]. Their model is trained with BPR using stochastic gradient ascent. He et al. propose Adversarial Personalized Ranking that enhances BPR by performing adversarial training [11]. Pei et al. propose a personalized re-ranking model to refine the initial list that is produced by state-of-the-art learning to rank methods [23]. They apply a Transformer network to encode the dependencies among items themselves as well as the interactions between the user and items. Using a pre-trained embedding to learn personalized encoding functions for different users, the performance of the re-ranking model can be further improved. Liu et al. propose a deep generative ranking model to enhance the accuracy and generalization of recommender systems [18]. Their experiments show significant ranking estimation improvements, especially for near-cold-start-users.

We have decided to not take a deep learning approach for our method. Firstly, because in our application context, we simply will not have the amounts of data such an approach demands. Secondly, for our purposes, where exploration vs. exploitation items are globally concentrated will be a more relevant question than devising an explicit ordered ranking of these items. This is strengthened by user experience constraints of our collaboration partner; music album suggestions will be presented in a 2-dimensional mosaic, rather than a 1-dimensional ordered list.

2.3 Multi-Armed Bandits

To address the exploration-exploitation dilemma, Li et al. propose an adaptive bandit clustering algorithm that can perform collaborative filtering [17]. Efficiency is achieved through the use of sparse graph representations. McInerney et al. propose Bart to jointly personalize recommendations and associated explanations to provide more transparent and understandable suggestions to users [20]. Bart learns how items and explanations interact within any provided context to estimate user satisfaction. Sanz-Cruzado et al. propose nearest-neighbor bandits that consider users to be the arms, which can be chosen as potential neighbors [28]. Recommendations are produced based on the advice from a neighbor of the target user, where neighbor selection is based on a stochastic choice.

Pereira et al. propose Counterfactual Dueling Bandits (CDB) that take an online learning to rank (OL2R) approach for sequential music recommendation suited for scarce-feedback problems [24]. CDB continuously learns from the user's listening feedback with a reward model that requires a single implicit feedback signal from the user at each interaction. Wang et al. propose Document Space Projection (DSP) for OL2R to reduce variance in gradient estimation and to improve performance [34]. They propose DSP Dueling Bandit Gradient Descent as an example of their general method.

These recent works tackle the exploration-exploitation dilemma with fairly diverse approaches to increase the use of recommender systems. Our work contributes by proposing explore-exploit item placement strategies and a user-aware exploration algorithm.

3 METHODS

In this section, we propose two filtering techniques to answer our research questions. We introduce explore-exploit item placement strategies in Subsection 3.1 and describe our user-aware exploration algorithm in Subsection 3.2. Finally, we merge the two proposed methods to yield our taste-broadening recommendation approach in Subsection 3.3. Each subsection introduces the concept, highlights the architectural implications, provides the pseudocode, and shows time efficiency.

As an abstract starting point, we consider recommender systems, as defined by Ricci et al., to be software tools and techniques that provide suggestions for items that are most likely of interest to a particular user [26]. In other words, this definition assumes the recommender to be a black box that outputs useful suggestions to the user. For our methods, we take a reinforcement learning approach, which assumes that the black box can perform processes related to exploration, exploitation, and tracking user preferences [30].

3.1 Explore-Exploit Item Placement Strategies

Explore-exploit item placement strategies are strategies to merge exploration and exploitation candidates deduced by the recommender system in a single representation such that the explore-exploit item placement suits the user. To integrate these in a recommender system, we need to introduce an explore-exploit item placement strategy generator. The generator takes as input the user preferences with respect to explore-exploit item placement and exploration, and outputs an explore-exploit item placement strategy that suits the user. The produced item placement strategy is then used to merge the exploration and exploitation candidates accordingly, which results in the recommendation.

By 'representation' we mean any human-interpretable representation. In this paper, we consider lists and mosaics. Whereas lists position items in a one-dimensional structure, mosaics do so in a two-dimensional structure. We assume a left-to-right top-to-bottom ordering of the positions.

Our explore-exploit item placement strategies take into account user variables with respect to explore-exploit item placement and exploration rate. As a starting point, we use the basic ϵ-greedy reinforcement learning algorithm. We can then compose our item placement strategies using a probability distribution to include the preferred explore-exploit item placement and a scaling factor to include the exploration rate. We choose to use the multivariate skew-normal distribution: a multivariate parametric family of distributions including the standard normal as a special case, where the extra parameter regulates the skewness [3]. We use the probability density function (PDF) and tune the skewness parameter based on the preferred explore-exploit item placement. In 1-D, if the skewness value decreases, the PDF gets skewed relatively more to the left. Similarly, if it increases, the PDF gets skewed relatively more to the right. If it is zero, the PDF is a standard normal distribution. In 2-D, we adopt a simplified version, namely a skew-normal PDF over the diagonal. Figure 1 illustrates this. With this version, we assume that users search the mosaic from the top-left to the bottom-right. Therefore, in the figure, the item positions on every k-antidiagonal of the heatmaps are treated the same.

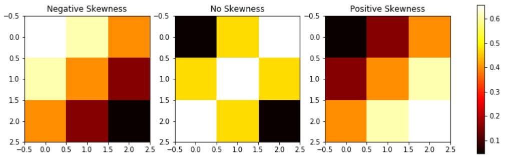


Figure 1: Three 3x3 Example Skewness Heatmaps

We propose Algorithm 1 to generate explore-exploit item placement strategies. Table 1 summarizes the notation used. Depending on the dimension of the representation, the `positions_to_pdf_input()` function maps the item positions of the representation to a list of inputs for the skew-normal PDF that respects the item position ordering. The hyperparameters are tuned beforehand to calculate a skewness and scaling factor that suit the user. For simplicity, we assume the mappings that compute the skewness and scaling factor to be linear. However, linearity is not a requirement. Assuming that the `positions_to_pdf_input()` function runs in $O(|X|)$ time, the algorithm runs in $O(|X|)$ time too, because the for-loop does not exceed this bound.

Table 1: Item Placement Strategy Summary of Symbols

Symbol	Meaning
p	Explore-exploit item placement, $p \in [0, 1]$
e	Exploration rate, $e \in [0, 1]$
a	Skewness factor, $a \in \mathbb{R}$
s	Scaling factor, $s \in \mathbb{R}_{\geq 0}$
$\alpha_1, \alpha_2, \beta_1, \beta_2$	Hyperparameters
X	List of skew-normal PDF input values
S	Produced item placement strategy

Algorithm 1 Item Placement Strategy Generation

```
function generate_item_placement_strategy($p$, $e$)
    Initialize list $S \leftarrow \emptyset$
    Determine list $X \leftarrow$ positions_to_pdf_input()
    $a \leftarrow \alpha_1 \cdot p + \beta_1$
    $s \leftarrow \alpha_2 \cdot e + \beta_2$
    for $i = 1, 2, \ldots, |X|$ do
        $S(i) = s \cdot$ skewnormal_pdf($X(i)$, $a$)
    return $S$
```

3.2 A User-Aware Exploration Algorithm

In the context of taste broadening, we assume that a digital collection can be partitioned into different categories (e.g., musical genres). Within a category, we propose to consider two personalization opportunities related to exploration: (1) a user's natural willingness towards exploration, as well as (2) the user's familiarity with the category. We consider that for a given user, items within a particular category can be ranked on the user's familarity with them, as expressed in the form of a familiarity score. Multiple items may be equally (un)familiar to a user; generally, the further away we will move out of a user's existing consumption profile, the more items are expected to be equally (un)familiar. Therefore, within a given category, user-specific familiarity profiles (FPs) can be constructed, that fan out in a 2-D conelike shape, as visualized in Figure 2. We assume that a user's exploration willingness and category familiarity will influence from what part of the FPs we can draw recommender items. For Alice, who is a rather conservative person, suitable items should remain close to her existing familiarity boundaries. For Bob, who is more adventurous, we can afford searching in a wider range. For Cynthia, who is an expert in category x, only truly unfamiliar items will allow for exploration.

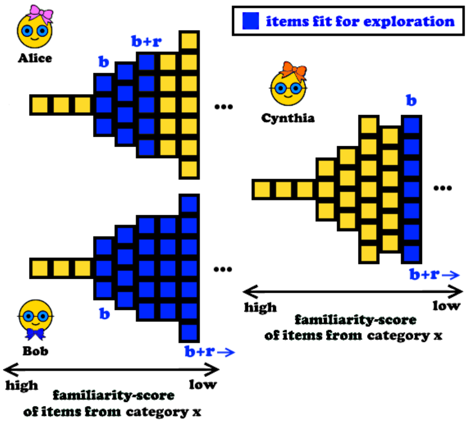


Figure 2: Different Taste Broadening User Requirements

Table 2: FP Initialization Summary of Symbols

Symbol	Meaning
j	The first in-order item that falls outside of range δ, $j \in \mathbb{N}$
f_i	Familiarity-score of i, $f_i \in \mathbb{R}$
d_i	Identifier of i, $d_i \in \mathbb{N}$
δ	Familiarity-score range, $\delta \in \mathbb{R}_{\geq 0}$
I	Category item set
F	Produced familiarity profile

We propose Algorithm 2 to initialize FPs. Table 2 summarizes the notation used. Note that A is only a temporary list. Depending on the familiarity-score function, a familiarity-score range is picked. The range serves as a measure to generally have a sufficient number of taste broadening items to choose from, as we move away from the user's comfort zone. If the familiarity-score function allows it, for ease, the range could be set to 0. Assuming that an efficient sorting algorithm is used, the algorithm runs in $O(|I| \log|I|)$ time.

Algorithm 2 Familiarity Profile Initialization

```
function initialize_familiarity_profile($I$)
    Sort list $I$ on familiarity-score in nonincreasing order
    Initialize list $F \leftarrow \emptyset$
    Initialize list $A \leftarrow \emptyset$
    $j \leftarrow 1$
    for $i = 1, 2, \ldots, |I|$ do
        if $f_i + \delta \geq f_j$ then
            Append $d_i$ to $A$
        else
            Append $A$ to $F$
            $A \leftarrow \emptyset$
            $j \leftarrow i + 1$
    return $F$
```

The FPs can be used to search for taste broadening items. To do so in a way that suits the user, as we will see in the following subsection, the user's category familiarity and exploration willingness are mapped to a suitable familiarity border and exploration range beforehand for the sake of efficiency, respectively.

We propose Algorithm 3 and 4 to explore with a user FP in a way that suits the user. Table 3 summarizes the notation used. Note that F was defined earlier. Algorithm 4 serves as a helper function that gets the taste broadening exploration subspace of the FP for Algorithm 3. To ensure the absence of duplicates in the recommendation, the items in the recommendation so far are excluded. Algorithm 3 is a recursive function that draws a random item from the retrieved subspace. A recursive call is performed if the retrieved subspace is empty. These algorithms *do not update the FP*, as the taste broadening item is not yet consumed by the user.

Moreover, we define three corner cases that can occur:

(1) *Negative exploration range.* For the sake of generality, we take into account that a user could have a negative attitude towards exploration, which means that the user prefers to stay inside the own comfort zone.
(2) *Empty taste broadening subspace.* As line 8 of Algorithm 4 takes a set difference , the taste broadening subspace could be empty. This requires an exploration range change.
(3) *Start or end FP index reached.* At initialization or on the fly, the start or the end index of the FP could be reached. If the retrieved taste broadening subspace is empty, the exploration range direction is inverted.

If we assume the size of the retrieved taste broadening subspace to be at least equal to the number of item positions in the representation, i.e., we are not in corner case 2, Algorithm 3 and 4 run in $O(|F|)$ time, where $|F|$ refers to the number of lists in the FP. Under this assumption, the worst-case is reached if $|K|$ from line 7 in Algorithm 4 equals $|F|$, which implies that the familiarity border is at the start or end index of the FP and the exploration range covers the full length of the FP.

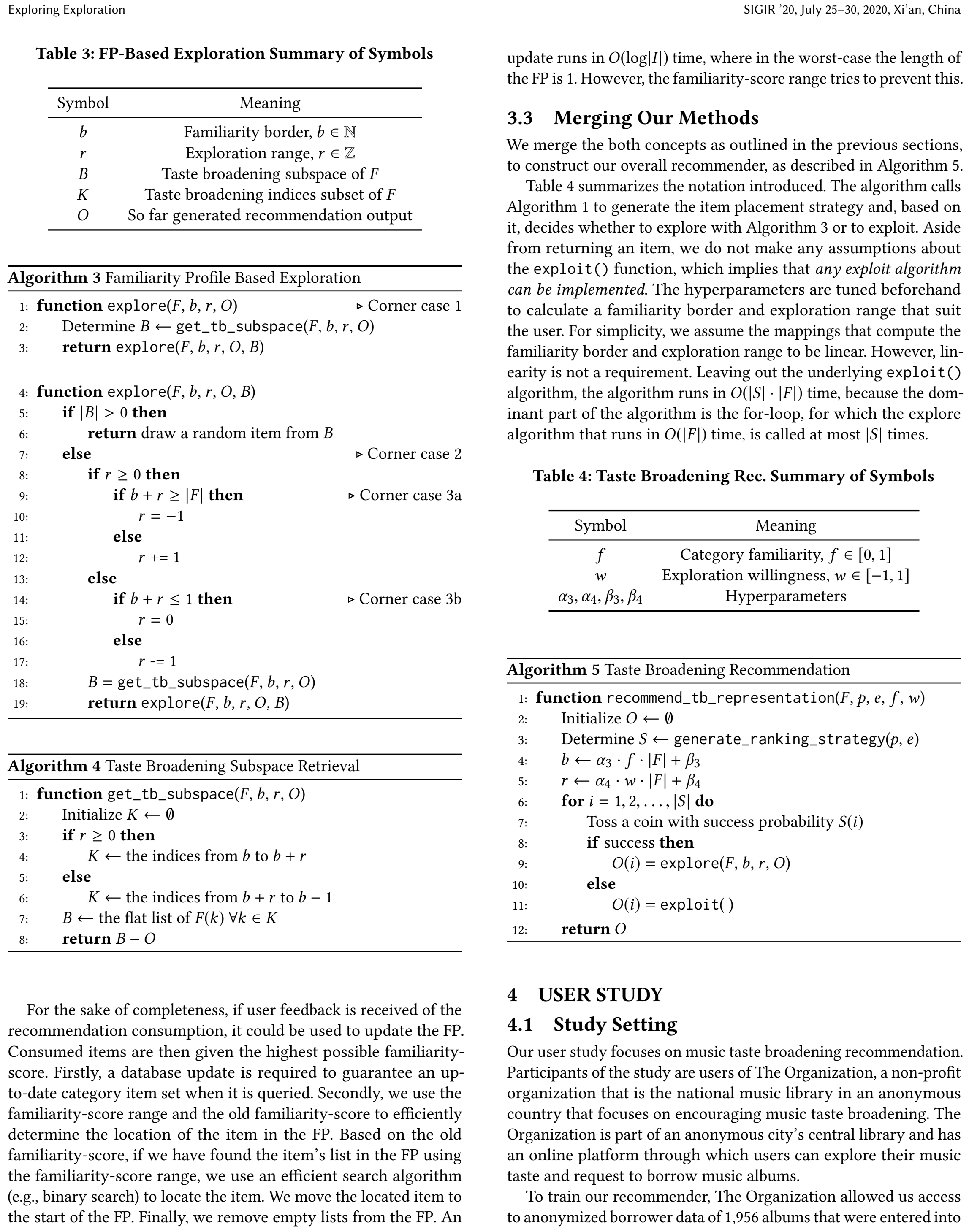


Table 3: FP-Based Exploration Summary of Symbols

Symbol	Meaning
b	Familiarity border, $b \in \mathbb{N}$
r	Exploration range, $r \in \mathbb{Z}$
B	Taste broadening subspace of F
K	Taste broadening indices subset of F
O	So far generated recommendation output

Algorithm 3 Familiarity Profile Based Exploration

```
function explore($F, b, r, O$)                          ▷ Corner case 1
    Determine $B \leftarrow$ get_tb_subspace($F, b, r, O$)
    return explore($F, b, r, O, B$)

function explore($F, b, r, O, B$)
    if $|B| > 0$ then
        return draw a random item from $B$
    else                                                ▷ Corner case 2
        if $r \geq 0$ then
            if $b + r \geq |F|$ then                    ▷ Corner case 3a
                $r = -1$
            else
                $r$ += 1
        else
            if $b + r \leq 1$ then                      ▷ Corner case 3b
                $r = 0$
            else
                $r$ -= 1
        $B$ = get_tb_subspace($F, b, r, O$)
        return explore($F, b, r, O, B$)
```

Algorithm 4 Taste Broadening Subspace Retrieval

```
function get_tb_subspace($F, b, r, O$)
    Initialize $K \leftarrow \emptyset$
    if $r \geq 0$ then
        $K \leftarrow$ the indices from $b$ to $b + r$
    else
        $K \leftarrow$ the indices from $b + r$ to $b - 1$
    $B \leftarrow$ the flat list of $F(k)$ $\forall k \in K$
    return $B - O$
```

For the sake of completeness, if user feedback is received of the recommendation consumption, it could be used to update the FP. Consumed items are then given the highest possible familiarity-score. Firstly, a database update is required to guarantee an up-to-date category item set when it is queried. Secondly, we use the familiarity-score range and the old familiarity-score to efficiently determine the location of the item in the FP. Based on the old familiarity-score, if we have found the item's list in the FP using the familiarity-score range, we use an efficient search algorithm (e.g., binary search) to locate the item. We move the located item to the start of the FP. Finally, we remove empty lists from the FP. An update runs in $O(\log|I|)$ time, where in the worst-case the length of the FP is 1. However, the familiarity-score range tries to prevent this.

3.3 Merging Our Methods

We merge the both concepts as outlined in the previous sections, to construct our overall recommender, as described in Algorithm 5.

Table 4 summarizes the notation introduced. The algorithm calls Algorithm 1 to generate the item placement strategy and, based on it, decides whether to explore with Algorithm 3 or to exploit. Aside from returning an item, we do not make any assumptions about the `exploit()` function, which implies that *any exploit algorithm can be implemented.* The hyperparameters are tuned beforehand to calculate a familiarity border and exploration range that suit the user. For simplicity, we assume the mappings that compute the familiarity border and exploration range to be linear. However, linearity is not a requirement. Leaving out the underlying `exploit()` algorithm, the algorithm runs in $O(|S| \cdot |F|)$ time, because the dominant part of the algorithm is the for-loop, for which the explore algorithm that runs in $O(|F|)$ time, is called at most $|S|$ times.

Table 4: Taste Broadening Rec. Summary of Symbols

Symbol	Meaning
f	Category familiarity, $f \in [0, 1]$
w	Exploration willingness, $w \in [-1, 1]$
$\alpha_3, \alpha_4, \beta_3, \beta_4$	Hyperparameters

Algorithm 5 Taste Broadening Recommendation

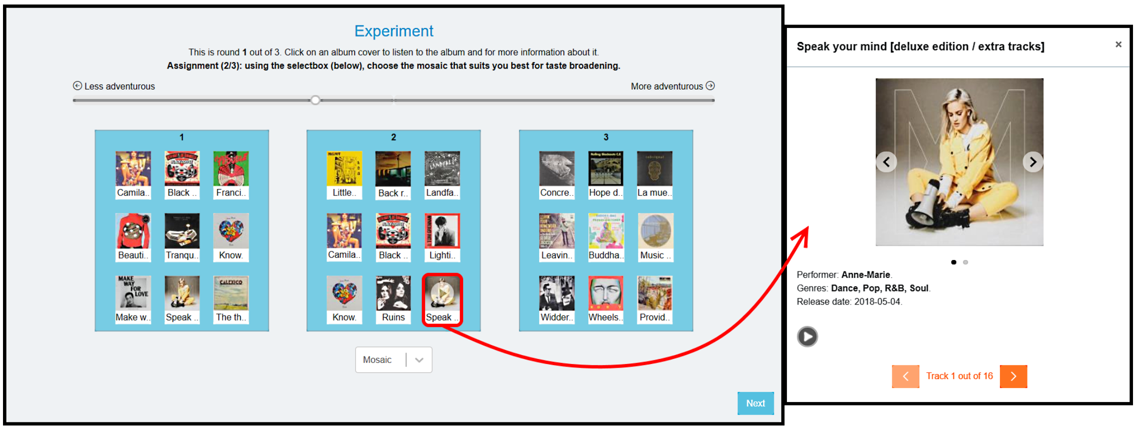
```
function recommend_tb_representation($F, p, e, f, w$)
    Initialize $O \leftarrow \emptyset$
    Determine $S \leftarrow$ generate_ranking_strategy($p, e$)
    $b \leftarrow \alpha_3 \cdot f \cdot |F| + \beta_3$
    $r \leftarrow \alpha_4 \cdot w \cdot |F| + \beta_4$
    for $i = 1, 2, \ldots, |S|$ do
        Toss a coin with success probability $S(i)$
        if success then
            $O(i)$ = explore($F, b, r, O$)
        else
            $O(i)$ = exploit( )
    return $O$
```

4 USER STUDY

4.1 Study Setting

Our user study focuses on music taste broadening recommendation. Participants of the study are users of The Organization, a non-profit organization that is the national music library in an anonymous country that focuses on encouraging music taste broadening. The Organization is part of an anonymous city's central library and has an online platform through which users can explore their music taste and request to borrow music albums.

To train our recommender, The Organization allowed us access to anonymized borrower data of 1,956 albums that were entered into



Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (2/3): using the selectbox (below), choose the mosaic that suits you best for taste broadening.

Mosaic

Speak your mind [deluxe edition / extra tracks]

Performer: **Anne-Marie**.

Genres: **Dance, Pop, R&B, Soul**.

Release date: 2018-05-04.

Track 1 out of 16

Figure 3: Impression of the Album Recommendation Experiment

their database between January and August 2018 and have been borrowed at least thrice. The data consists of 17,528 borrowing records, traced back to 2,721 unique users. As borrowing an album requires the payment of a fee and considers a physical transaction, we assume that users will only borrow albums of which they have strong expectations they will like them. Thus, borrowing behavior may be indicative of a user’s comfort zone. Section 5 elaborates on how we will use this information as user profiling training data.

From an internal online customer survey, conducted with 691 participants in November 2019, the following demographics information emerged for the users of The Organization. 80% are male, 14% female, and the rest preferred not to say. 82% are at least 50 years old (5% preferred not to say); more specifically, the age of 27% is between 50 and 59, 38% is between 60 and 69, and the remaining 17% is at least 70. 60% hold a higher education degree (9% preferred not to say). Moreover, 413 out of 529 participants indicated to borrow music from the library. Therefore, the presented demographics information is highly indicative for users that borrow music.

4.2 Survey

The survey consists of six questions, some of which are composed of several subquestions. Our aims are twofold, namely to gather information to (1) take into account any participant profile skews during evaluation and (2) make predictions in the experiment. The questions are ordered in the same manner. Firstly, we ask the participants for their age, gender, nationality, country of current residency, and country of formative years. Secondly, we pose questions to determine the user variables covered in Section 3 explicitly.

More specifically, given a preferred category, we estimate the preferred explore-exploit item placement, exploration rate, category familiarity, and exploration willingness. Table 5 shows the three relevant questions. The multi-select options of questions 1a. and 1c. are derived from our borrower data covered in the previous subsection. The top 40 most occurring genres, of which we have at least 30 albums, are presented to the participants to choose from. We also allow participants to add missing genres to their list. For each of the picked genres from the top 40, participants choose their

Table 5: User Preference Related Survey Questions

No.		Assignment/Question/Statement	Format	User variable
1	a.	Which music genres do you prefer?	Multi-select	Preferred category
	b.	For each of your given music genres: how familiar are you with the genre?	5-point Likert	Category familiarity f
	c.	For each of the genres you are minimally ‘Rather familiar’ with: select at least 1 artist you know already	Multi-select	-
2	a.	I browse on a recommender system, like Youtube or Netflix, actively for something new with the following frequency	5-point Likert	Exploration rate e
	b.	In the same context as before, if I stumble upon something new, then I try it	5-point Likert	Exploration rate e
	c.	In general, I see myself as someone curious about many different things	5-point Likert	Exploration willingness $\hat{w}$
	d.	As a guiding principle in my life, I rate ’being curious’ to be a personal value on a nine-point scale as	9-point scale	Exploration willingness $\hat{w}$
3.		In the context of taste broadening: given a list of recommendations, where do you prefer the emphasis of taste broadening items to be?	Multiple choice	Item placement p

familiarity and, if at least rather familiar (2 on the Likert scale), pick at least one familiar artist from our data's musicians of the genre.

We normalize the filled out Likert scale score with the following formula: $n_5(s) = \frac{s-1}{4}$. We compute the category familiarity as: $f = n_5(s_{1b})$, where the subscript of s corresponds to the question number of Table 5. A 9-point scale that ranges from -1 to 7 is used for question 2d. to determine the personal value 'openness to experience' [29]. Likewise, we normalize the 9-point scale score with the following formula: $n_9(s) = \frac{s}{7}$. To determine the exploration rate and exploration willingness, we average the filled out scores of the two particular questions, as these positively correlate. That is, $e = \frac{n_5(s_{2a})+n_9(s_{2b})}{2}$ and $\hat{w} = \frac{n_5(s_{2c})+n_9(s_{2d})}{2}$, respectively, where $\hat{w} \in [-1, 1]$ is our w prediction. The multiple choice options of question 3. are limited to 'At the beginning', 'Around the middle', 'At the end', and 'No specific preference', for which we assign 0, 0.5, 1, and `null` to p, respectively. If p is `null`, we apply a uniform distribution.

5 EXPERIMENTS

The experiment is composed of three assignments and has three rounds. Our aims are to answer our research questions, as defined in Section 1. For the sake of completeness, based on Section 3, in **RQ1**, exploration refers to exploration rate and exploration willingness, and familiarity refers to category familiarity.

5.1 Assignments

In each round, three album mosaics are presented, as visualized in Figure 3, which are produced by different methods and ordered randomly. Participants can listen to an album by clicking on it; this opens a modal that contains the album information and songs. The chosen genres depend on the number of indicated preferred genres in the survey. If one genre is given, it is used for each round. Else, if two are given, either one of them is assigned twice. Else, three unique genres are picked at random. In the three rounds, we separately check the performance contribution of explore-exploit item placement, exploration rate, and category familiarity, respectively. The performance contribution of the exploration willingness is adjusted through the adventurousness slider at the top of the page.

Participants sequentially complete the following assignments:

A1 Using the slider, choose the adventure value in the blue area that suits you best for taste broadening.

A2 Using the selectbox below, choose the mosaic that suits you best for taste broadening.

A3 Select the albums that you find unfit for taste broadening.

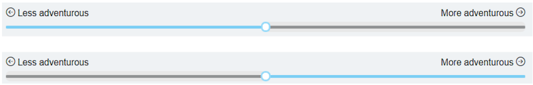


Figure 4: The Two Different Slider Settings

In **A1**, the participant can tune the recommendation adventurousness. The value at the center of the slider corresponds to our exploration willingness prediction $\hat{w}$. The slider's value is initialized at its center and, depending on the round, can either be moved to the left or right. Figure 4 illustrates this. More specifically, in round 1 it can only be moved to the left (the upper setting), round 2 only to the right (the lower setting), and round 3 to either side at random. This procedure encourages participants to look for an adventurousness value away from the center, to both the left and right of it. Also, it helps to answer **RQ2**, since results that are consistent over the rounds and have low variance over the participants likely indicate a high contribution. Given adventurousness value $n \in \{0, 1, \ldots, 100\}$, we compute the exploration willingness as: $w = \frac{2n}{100} \cdot \hat{w}$.

In **A2**, the participant chooses one mosaic out of three that suits best for his/her taste broadening. These are the following mosaics:

- Our prediction mosaic. It is generated using our methods and the determined user variables.
- A mutation mosaic. It is generated like the prediction mosaic, except for the fact that exactly one of the determined user variables is mutated beforehand. A mutated variable gets the inverted original value. If the original value is at the center of the range, it gets the value of an extreme at random.
- A random mosaic. It contains randomly drawn items from the category's item collection.

As for the mutations, we mutate in round 1 explore-exploit item placement, round 2 exploration rate, and round 3 category familiarity. This assignment helps to answer **RQ1**, as results that show the prediction mosaic is generally picked over the others demonstrate the usefulness of our methods. Moreover, it helps to answer **RQ2**, as results that show the prediction mosaic is generally picked over the mutation mosaic for the same mutated variable demonstrate the contribution of the individual user preference.

In **A3**, the participant labels the albums unfit for his/her taste broadening of the mosaic chosen in **A2**. This assignment serves as a sanity check and helps to answer **RQ3**. If the mutation or random mosaic is chosen, we determine which labeled albums are outside of the FP's taste broadening subspace. Else, we determine where users draw the line for user-aware recommendation.

5.2 Multi-Armed Bandit

For each genre, we train a finite-armed stochastic multi-armed bandit with our dataset's borrowing records for that genre, to recommend an album based on a target album. For the bandit, the context is the target album, the arms are the albums in the item collection, the reward function equals the familiarity-score function, and the arms are updated after each recommendation. We define the familiarity-score function as:

$$f(s,t) = n(s) + \mathbb{1}_{n(s,t)\geq 1} \cdot (m + n(s,t)) + \mathbb{1}_{s=t} \cdot m,$$

where, from the genre's collection, $n(\cdot)$ computes the number of times its input albums appear together in a borrower's record and m is the maximum number of times an album has been borrowed. We use ϵ-greedy to train it, until the policy converges.

Moreover, for the experiment, we implement our proposed algorithms for item placement and exploration. The target albums are the albums of the indicated familiar artists in the survey. Each time the bandit exploits, it picks a random target from the target albums. Our familiarity function guarantees that the beginning of each FP will contain several items that are rather familiar, if not familiar, while the end will contain relatively many albums that are irrelevant to the target and borrowed only a few times. Therefore, we set familiarity-score range δ to 0.

As for our remaining hyperparameters, based on an analysis of the skew-normal distribution, we set $\alpha_1 = 3$, $\alpha_2 = 2$, $\beta_1 = -1.5$, and $\beta_2 = 0$. We set $\beta_3 = 0.05$ to slightly encourage taste broadening. Lastly, for ease, we set $\alpha_3 = \alpha_4 = 1$ and $\beta_4 = 0$. Our pilot study, which is presented in the next section, serves as a sanity check.

5.3 Hypotheses

We hypothesize beforehand that:

H1 Overall, users will prefer mosaics based on our personalized recommendations.

H2 The integration of each user preference variable contributes positively.

H3 For mosaics with our personalized recommendations, fewer items are marked unfit. Also, the items marked as unfit are mostly outside of the FP's taste broadening subspace.

6 RESULTS AND DISCUSSION

This section covers the experimental results. Subsection 6.1 and 6.2 present the results of our pilot and live study, respectively. These studies sequentially ran during October and November 2019. We then discuss the results in Subsection 6.3.

6.1 Pilot Study

The aims of the pilot study are to check whether the instructions in the online user study are clear to participants, the UI design is intuitive, and the overall functionality works as planned in a real-user setting. For the study, we recruited eight participants. These people were excluded from the live experiments. Together with each participant, we went through the survey and experiment. We asked the participants beforehand to think aloud, to enable us to follow their thought process. They could only pose us urgent questions.

The participant profiles of our study are as follows. 75% are male and 25% female. The average age is 46 and 50% are at least 50 years old. 62.5% hold a higher education degree. Therefore, based on a comparison with the outcomes of the user profile study presented in Subsection 4.1, we consider our sample to be representative.

During the evaluative talks at the end of the sessions, the participants consistently confirmed that they broadened their taste. Each participant found artists or songs that they recognized, if not vaguely recognized, and based on these, successfully found a mosaic that suited them best. To quote one participant, "I learned to appreciate new songs of artists I did not know, who are close to my favorite artists." Overall, we received positive feedback.

6.2 Live Study

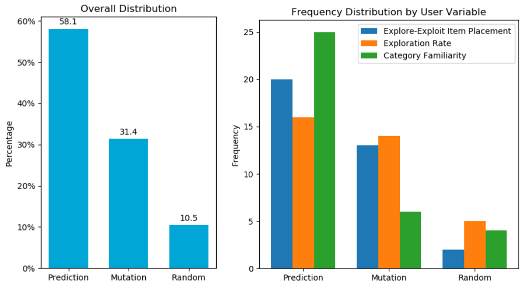
35 users participated in our live study. In terms of demographic information of our participants, we gathered the following summary statistics, which are in line with existing customer statistics as presented in Subsection 4.1. 86% are male and 14% are female. The average age is 50 and 66% are at least 50 years old; more specifically, the age of 23% is between 50 and 59, 32% is between 60 and 69, and the remaining 11% is at least 70. 94% live in the library's country and 6% live in a neighboring country.

Table 6 shows statistics of the user variables deduced from the survey. Recall that exploration willingness is defined on $[-1, 1]$, whereas the other user preferences variables are defined on $[0, 1]$. Participants picked a remarkably large number of preferred genres and familiar artists. Moreover, there is no consensus on the preferred explore-exploit item placement.

Table 6: User Variable Statistics

Variable	Mean	SD
Number of preferred genres	5.5	4.4
Category familiarity f	0.59	0.18
Number of familiar artists per genre	9.2	10.7
Exploration rate e	0.52	0.25
Exploration willingness $\hat{w}$	0.19	0.14
Explore-exploit item placement p	0.53	0.43

Relating to **RQ1**, we analyze the experiment results visualized in Figure 5a. We find that our prediction mosaic was significantly preferred, $\chi^2(2, N = 105) = 35.89, p < .001$. Furthermore, relating to **RQ2**, we analyze the results visualized in Figure 5b. We find that our prediction mosaic was significantly preferred in the explore-exploit item placement, $\chi^2_p(2, N = 35) = 14.11$, and category familiarity, $\chi^2_f(2, N = 35) = 23.03$, mutation rounds, $p < .001$. However, this does not hold for the exploration rate mutation rounds, $\chi^2_e(2, N = 35) = 5.89, p \approx .053$.



(a) Overall **(b) Individual Preferences**

Figure 5: Histograms of the Chosen Album Mosaics

Relating to **RQ3**, Table 7 shows statistics of the albums that users marked as unfit for their taste broadening. For our mosaics, all exploration items are from the FP's taste broadening subspace. This does not hold for the other mosaic types. We use a Linear Model analysis to test the difference between the mosaic types on the number of marked unfit albums. We find a significant effect ($F(2, 102) = 12.35, p < .001$) for the mosaic types on the number of unfit marked items. Post hoc comparisons using the Tukey HSD test show that the random mosaics and each other group differ significantly at $p < .05$. Besides, they show no significant difference between the prediction mosaics and the mutated mosaics.

Relating to **RQ2**, we analyze the contribution of exploration willingness by studying the values that participants chose with the adventurousness slider. These are visualized in Figure 6. We find that the participants significantly preferred adventurousness

Table 7: Items Marked as Unfit Statistics

Mosaic	Mean	SD	Exploration Items (%)
Prediction	1.16	1.15	71.9
Mutation	1.79	1.77	70.4
Random	3.54	1.92	-

values that are relatively more to the right ($M = 55$, $SD = 17.4$) of the slider's center, $t(104) = 32.53$, $p < .001$. We find that the values on the same side of the slider are consistently chosen by the participants over the rounds ($M = 3.62$, $SD = 5.02$).

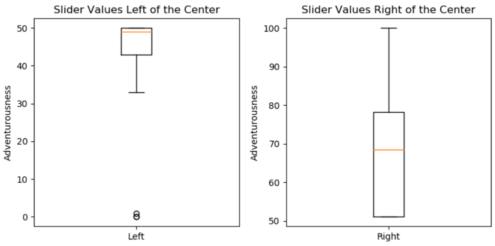


Figure 6: Boxplots of the Adventurousness Values

6.3 Discussion

We would first like to emphasize that participation was completely voluntary. By this, we mean that no rewards to encourage participation were attached, as monetary incentives may diminish the quality of the responses. On average, participants spent 45.2 minutes ($SD = 56.0$) on their response. Their committed participation is also visible from Table 6, which shows a high number of indicated genres and artists. Therefore, participants put much effort in doing the experiments, which resulted in a valuable dataset.

Based on our training data, on average, users borrow albums from 6.1 different genres ($SD = 5.5$), filtered on our genre top 40. If we compare this average with the average number of preferred genres from Table 6, we find that these are close. In Subsection 4.1 we stated that the albums borrowed are probably close to, if not inside, the customers' comfort zones, because of borrowing costs. The fact that these averages are close confirms that the training data should be used for exploitation, in taste broadening recommendation. Moreover, on average, the borrowed albums per user are from 5.9 different artists ($SD = 12.0$). Table 6 shows a larger average, since the number is related to familiarity and familiarity does not imply preference.

We find that our hypotheses, as defined in Section 5, are mostly in line with our results. As for **RQ1** and **RQ2**, we found that our mosaic is overall, and in two out of the three mutation round types, significantly preferred. We expected exploration rate to be the least contributing user variable, as it affects the content of the recommendation least, namely only through the exploration probability. Also, we expected category familiarity to be the most contributing variable, as it directly influences the connection between the user and the recommendation content. However, although we found that our mosaic was significantly preferred in explore-exploit item placement mutation rounds, we expected the influence of this user variable to be bigger. One reason for this is the fact that five participants indicated to not have a specific item placement preference.

Based on Figure 6, participants significantly preferred values to the right of the slider's center. An influential factor that could explain this, is the fact that it is natural for humans to be curious about what is beyond the default value, especially in the context of adventurousness. This implies that participants will anyway choose a value that is relatively right to the slider's center. As we also found that participants chose adventurousness values that are consistent over the rounds, we conclude that the exploration willingness contributes positively to our recommendations as well.

As for **RQ3**, we found that for our recommendations, participants labeled significantly fewer items as unfit. Table 7 shows that, on average, approximately one item is marked as unfit for our recommendations. An influential factor could be that participants have the urge to mark at least one item as unfit – the item that least suits their taste broadening. The table also shows a relatively high SD for the mutation mosaics. These deviations mainly result from the category familiarity mutation rounds, which show a relatively large number of unfit marked items ($M = 4.60$, $SD = 1.36$). Moreover, we found that items outside of the FP's taste broadening subspace are less likely to be fit for taste broadening. Also, pilot study participants marked items from our mosaic as unfit mostly by comparison with the other albums inside the mosaic. This suggests that participants look in our mosaics for the item that relatively least contributes to their taste broadening. However, based on this research, we cannot give a precise answer of where to draw the line between items fit and unfit for taste broadening in the FP itself. To advance to this level, the question deserves a separate qualitative research.

7 CONCLUSION

In this paper, we introduced explore-exploit item placement strategies and, in the context of taste broadening, a user-aware exploration algorithm. These filtering techniques integrate user variables with respect to explore-exploit item placement, exploration, and familiarity to answer the questions of "Where?", "How much?", and "For whom?" to explore, respectively. Based on the derived time-complexities, the proposed algorithms are time-efficient by definition, which is of great value for recommendation. Experiments demonstrate the effectiveness of our methods and the fact that the integration of familiarity contributes most, for taste broadening recommendation. We hope that this paper serves as an eye-opener to trigger more research to be centered around exploration in recommender systems. As shown, there is more ground to be gained.

In future work, we want to explore the possible benefits of tree-based familiarity profile implementations. Also, we want to research explore-exploit item placement strategies and FPs in collaborative settings. Another interesting future work direction could be to analyze item placement strategies and FPs over time. For item placement strategies, we could research the applications of exploration rate decay functions that deal with the cold-start problem. Likewise, for FPs, we could research familiarity decay functions that decrease the familiarity-score of items over time.

REFERENCES

[1] Panagiotis Adamopoulos and Vilma Todri. 2015. Personality-Based Recommendations: Evidence from Amazon.com. In *9th ACM Conference on Recommender Systems Poster Proceedings*. Association for Computing Machinery (ACM), United States.

[2] Kumaripaba Athukorala, Alan Medlar, Antti Oulasvirta, Giulio Jacucci, and Dorota Glowacka. 2016. Beyond Relevance: Adapting Exploration/Exploitation in Information Retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces (IUI '16)*. ACM, New York, NY, USA, 359–369. https://doi.org/10.1145/2856767.2856786

[3] Adelchi Azzalini and Alessandra Dalla Valle. 1996. The multivariate skew-normal distribution. *Biometrika* 83, 4 (12 1996), 715–726. https://doi.org/10.1093/biomet/83.4.715

[4] Iván Cantador, Ignacio Fernández-Tobías, Alejandro Bellogín, Michal Kosinski, and David Stillwell. 2013. Relating Personality Types with User Preferences Multiple Entertainment Domains, In CEUR workshop proceedings. *CEUR Workshop Proceedings* 997.

[5] 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

[6] Paul Costa and Robert McCrae. 2008. The revised NEO personality inventory (NEO-PI-R). *The SAGE Handbook of Personality Theory and Assessment* 2 (01 2008), 179–198. https://doi.org/10.4135/9781849200479.n9

[7] John M. Digman. 1990. Personality Structure: Emergence of the Five-Factor Model. *Annual Review of Psychology* 41, 1 (1990), 417–440. https://doi.org/10.1146/annurev.ps.41.020190.002221

[8] Jennifer Golbeck and Eric Norris. 2013. Personality, Movie Preferences, and Recommendations. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*. ACM, New York, NY, USA, 1414–1415. https://doi.org/10.1145/2492517.2492572

[9] Frédéric Guillou. 2016. *On Recommendation Systems in a Sequential Context*. Theses. Université Lille 3. https://tel.archives-ouvertes.fr/tel-01407336

[10] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 144–150. http://dl.acm.org/citation.cfm?id=3015812.3015834

[11] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 355–364. https://doi.org/10.1145/3209978.3209981

[12] Katja Hofmann, Shimon Whiteson, and Maarten Rijke. 2013. Balancing Exploration and Exploitation in Listwise and Pairwise Online Learning to Rank for Information Retrieval. *Inf. Retr.* 16, 1 (Feb. 2013), 63–90. https://doi.org/10.1007/s10791-012-9197-9

[13] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1, Article 2 (Dec. 2016), 42 pages. https://doi.org/10.1145/2926720

[14] Peter Knees, Markus Schedl, Bruce Ferwerda, and Audrey Laplante. 2019. User Awareness in Music Recommender Systems. In *Personalized human-computer interaction*. De Gruyter Open.

[15] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems – A survey. *Knowledge-Based Systems* 123 (2017), 154 – 162. https://doi.org/10.1016/j.knosys.2017.02.009

[16] Audrey Laplante. 2014. Improving Music Recommender Systems: What Can We Learn from Research on Music Tastes?. In *ISMIR*. 451–456.

[17] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative Filtering Bandits. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2911451.2911548

[18] Huafeng Liu, Jingxuan Wen, Liping Jing, and Jian Yu. 2019. Deep Generative Ranking for Personalized Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, New York, NY, USA, 34–42. https://doi.org/10.1145/3298689.3347012

[19] Sandy Manolios, Alan Hanjalic, and Cynthia C. S. Liem. 2019. The Influence of Personal Values on Music Taste: Towards Value-based Music Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, New York, NY, USA, 501–505. https://doi.org/10.1145/3298689.3347021

[20] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 31–39. https://doi.org/10.1145/3240323.3240354

[21] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. ACM, New York, NY, USA, 677–686. https://doi.org/10.1145/2566486.2568012

[22] Eli Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin UK.

[23] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, and Dan Pei. 2019. Personalized Re-ranking for Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, New York, NY, USA, 3–11. https://doi.org/10.1145/3298689.3347000

[24] Bruno L. Pereira, Alberto Ueda, Gustavo Penha, Rodrygo L. T. Santos, and Nivio Ziviani. 2019. Online Learning to Rank for Sequential Music Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, New York, NY, USA, 237–245. https://doi.org/10.1145/3298689.3347019

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[26] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook* (2nd ed.). Springer Publishing Company, Incorporated.

[27] Sonia Roccas, Lilach Sagiv, Shalom Schwartz, and Ariel Knafo-Noam. 2002. The Big Five Personality Factors and Personal Values. *Personality and Social Psychology Bulletin* 28 (06 2002), 789–801. https://doi.org/10.1177/0146167202289008

[28] Javier Sanz-Cruzado, Pablo Castells, and Esther López. 2019. A Simple Multi-armed Nearest-neighbor Bandit for Interactive Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, New York, NY, USA, 358–362. https://doi.org/10.1145/3298689.3347040

[29] Shalom H. Schwartz. 1992. Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries. Advances in Experimental Social Psychology, Vol. 25. Academic Press, 1 – 65. https://doi.org/10.1016/S0065-2601(08)60281-6

[30] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, USA.

[31] Alexandra Uitdenbogerd and R Schyndel. 2002. A review of factors affecting music recommender success. In *ISMIR 2002, 3rd International Conference on Music Information Retrieval*. IRCAM-Centre Pompidou, 204–208.

[32] Hastagiri P. Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. 2014. Explore-exploit in top-N Recommender Systems via Gaussian Processes. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 225–232. https://doi.org/10.1145/2645710.2645733

[33] Saúl Vargas and Pablo Castells. 2011. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. ACM, New York, NY, USA, 109–116. https://doi.org/10.1145/2043932.2043955

[34] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. 2019. Variance Reduction in Gradient Exploration for Online Learning to Rank. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, New York, NY, USA, 835–844. https://doi.org/10.1145/3331184.3331264

(This page has been intentionally left blank.)

B

Muziekweb’s Explore Features

In Section 1.7, the Explore features provided by Muziekweb were explained. Supplementary to it, this appendix contains impressions of the features other than music album advice generation. For an impression of Muziekweb’s Album Advice page, we refer back to Figure 1.3.

Figure B.1 provides an impression of the Muziekweb Intro’s page, which contains eight genres in this image [59]. Besides, Figure B.2 visualizes the Muziekweb Playlist page, which contains two playlists in the image [60]. Furthermore, Figure B.3 visualizes the Music Advice page, where the user can navigate to the Artist Advice or the Album Advice page [58]. Finally, Figure B.4 provides an impression of the Artist Advice page, where users can provide a set of input artists to generate an artist advice. The user can use a slider to indicate the preferred popularity of the artists for the artist recommendation.

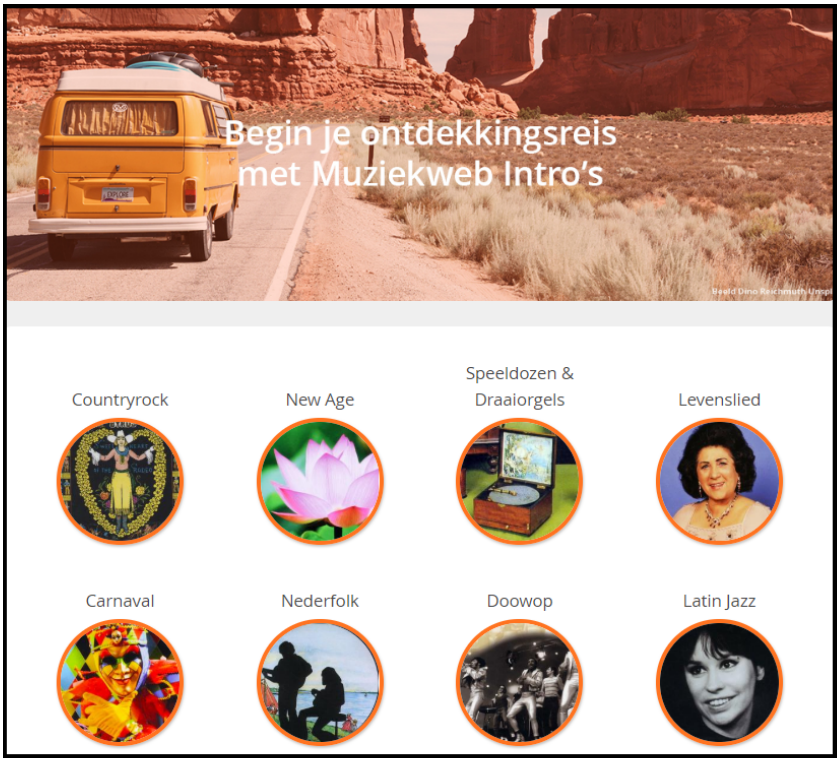
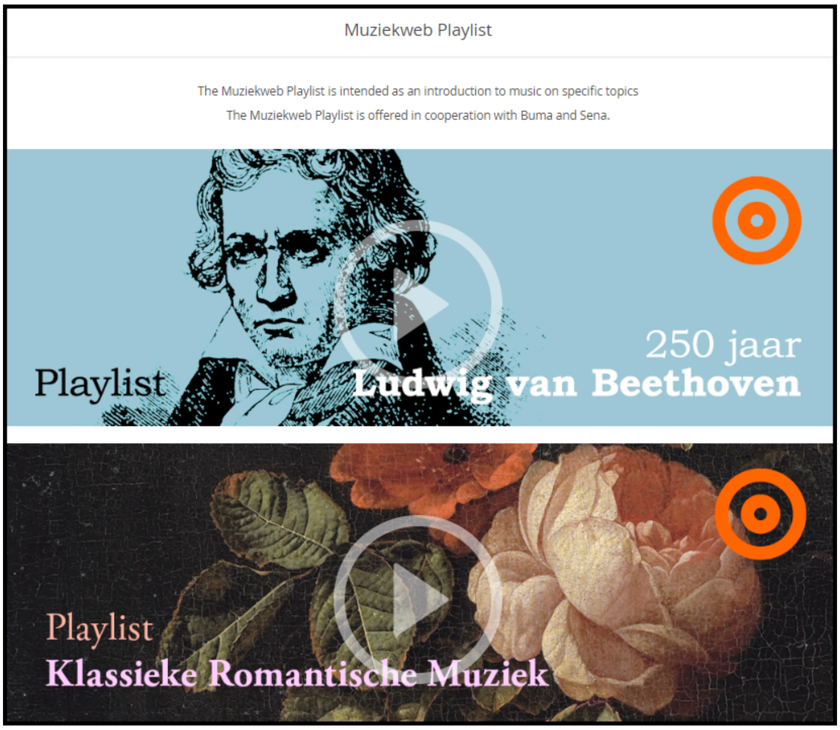


Figure B.1: Muziekweb Intros Page.

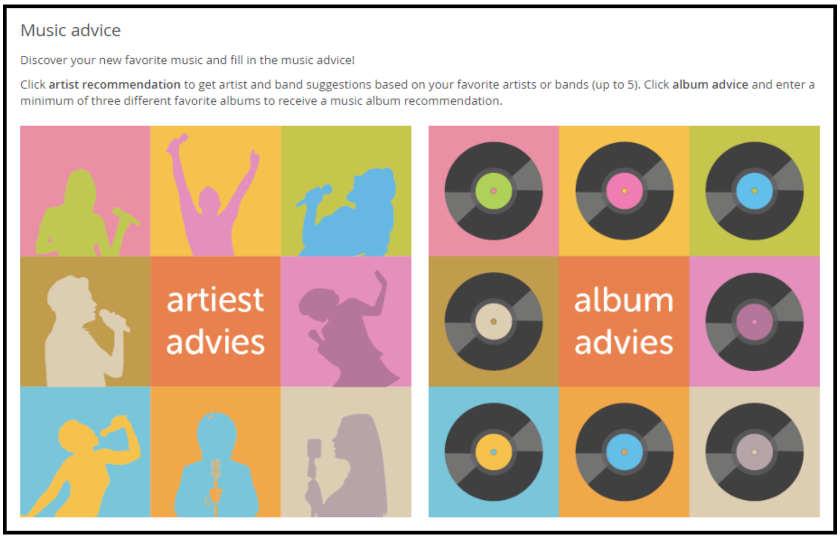


Muziekweb Playlist

The Muziekweb Playlist is intended as an introduction to music on specific topics

The Muziekweb Playlist is offered in cooperation with Buma and Sena.

Figure B.2: Muziekweb Playlist Page.

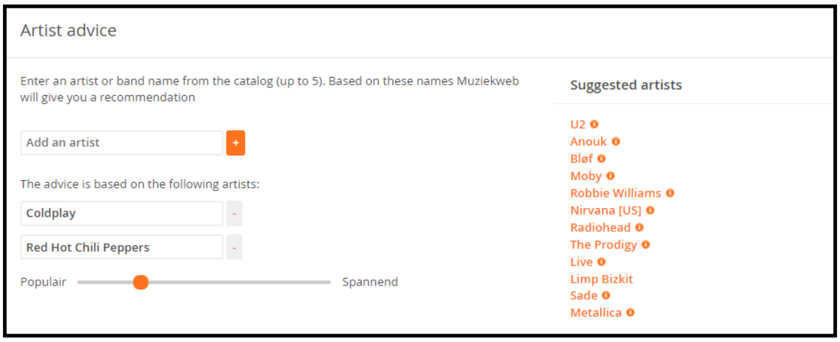


Music advice

Discover your new favorite music and fill in the music advice!

Click **artist recommendation** to get artist and band suggestions based on your favorite artists or bands (up to 5). Click **album advice** and enter a minimum of three different favorite albums to receive a music album recommendation.

Figure B.3: Muziekweb's Music Advice Page.



Artist advice

Enter an artist or band name from the catalog (up to 5). Based on these names Muziekweb will give you a recommendation

Add an artist +

The advice is based on the following artists:

Coldplay -

Red Hot Chili Peppers -

Populair Spannend

Suggested artists

- U2
- Anouk
- Bløf
- Moby
- Robbie Williams
- Nirvana [US]
- Radiohead
- The Prodigy
- Live
- Limp Bizkit
- Sade
- Metallica

Figure B.4: Muziekweb's Artist Advice Page.

C
Exploration Explorer’s Technologies

In Section 5.5, Exploration Explorer was presented. Supplementary to it, this appendix covers the technologies used to develop and run the website.

The website and server have mainly been developed in JavaScript (JS). More specifically, with ECMAScript 2019, which comes with next-generation JS features. We use JS because of our experience with the language and since it is a flexible, popular, and evergrowing language. Moreover, JS allows us to use Node package manager (npm), which, due to popularity, offers many packages that are of great use for web development. Furthermore, we have implemented our methods in Python, as the packages available for Python make the mathematical implementation easier.

The following enumeration covers which technologies have been used to develop and run the different parts of Exploration Explorer. We use:

- *Node.js* to run our server. It is JS without the browser; in other words, it is a runtime environment used to build JS applications. Along with its installation comes npm.
- *Express* to create the routing for the server. It is a relatively simple Node.js web application framework.
- *React* for frontend development. It is a JS library that is designed to update the browser DOM for us. It requires us to work with a ‘virtual’ DOM consisting of React elements, which are JS objects, on which operations will influence the ‘actual’ DOM.
- *SASS* for frontend design. It is a layer upon CSS that enables us to efficiently define frontend styles.
- *MongoDB* for database development. It allows for flexibility in database architecture design. Using the Mongoose node package, we can work with it in JS.

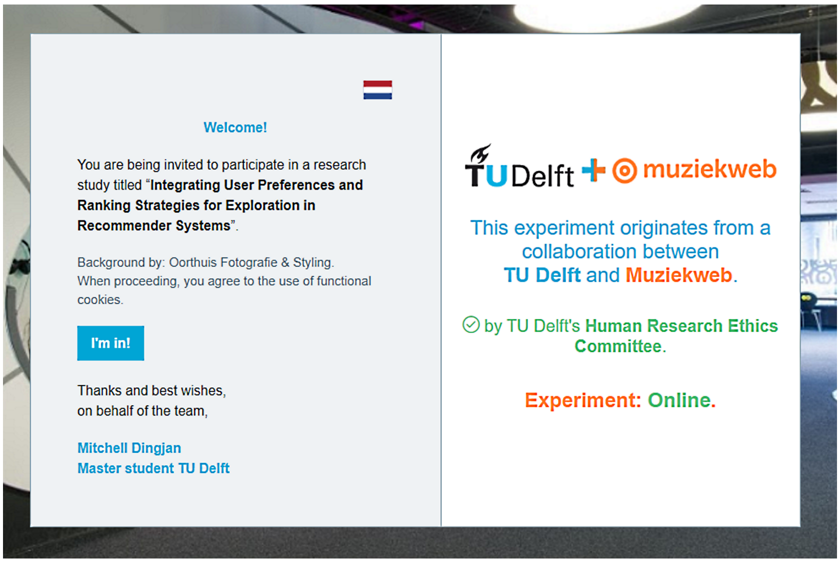
Together these technologies allow us to develop Exploration Explorer in an elegant way. We manage to get the different languages to work well together as well. Whenever the server needs to run the implementation of our methods, it calls a Python script and waits for it to finish. When the script completes its computations, the script returns the results to the active JS process, which in turn uses these to proceed with its computations.

(This page has been intentionally left blank.)

D

Exploration Explorer Walkthrough

This appendix contains a figure-by-figure walkthrough of Exploration Explorer.



Welcome!

You are being invited to participate in a research study titled "**Integrating User Preferences and Ranking Strategies for Exploration in Recommender Systems**".

Background by: Oorthuis Fotografie & Styling.
When proceeding, you agree to the use of functional cookies.

I'm in!

Thanks and best wishes,
on behalf of the team,

Mitchell Dingjan
Master student TU Delft

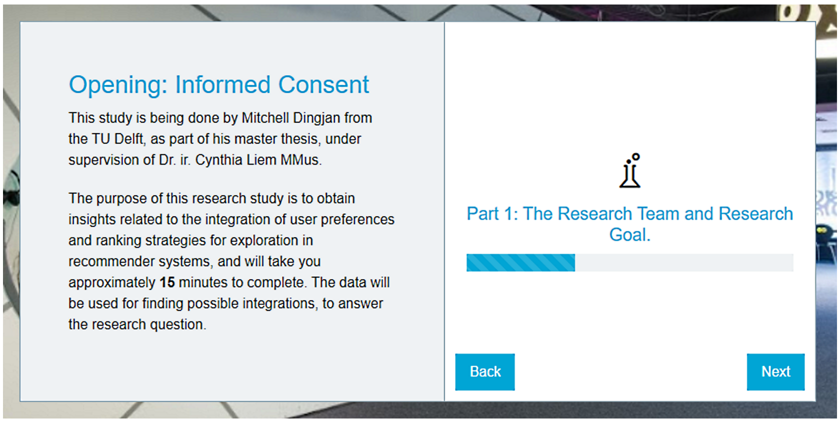
TU Delft + muziekweb

This experiment originates from a collaboration between **TU Delft** and **Muziekweb**.

✓ by TU Delft's **Human Research Ethics Committee**.

Experiment: Online.

Figure D.1: The Landing Page.



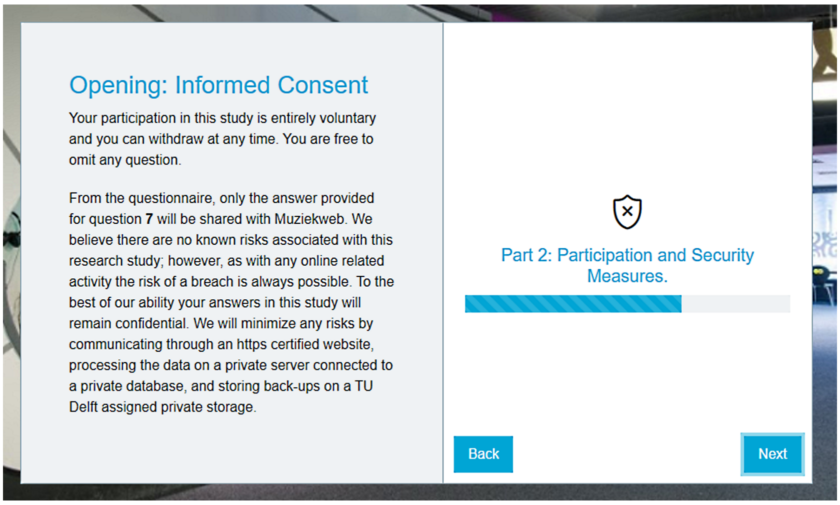
Opening: Informed Consent

This study is being done by Mitchell Dingjan from the TU Delft, as part of his master thesis, under supervision of Dr. ir. Cynthia Liem MMus.

The purpose of this research study is to obtain insights related to the integration of user preferences and ranking strategies for exploration in recommender systems, and will take you approximately **15** minutes to complete. The data will be used for finding possible integrations, to answer the research question.

Part 1: The Research Team and Research Goal.

Figure D.2: Informed Consent (1/3).

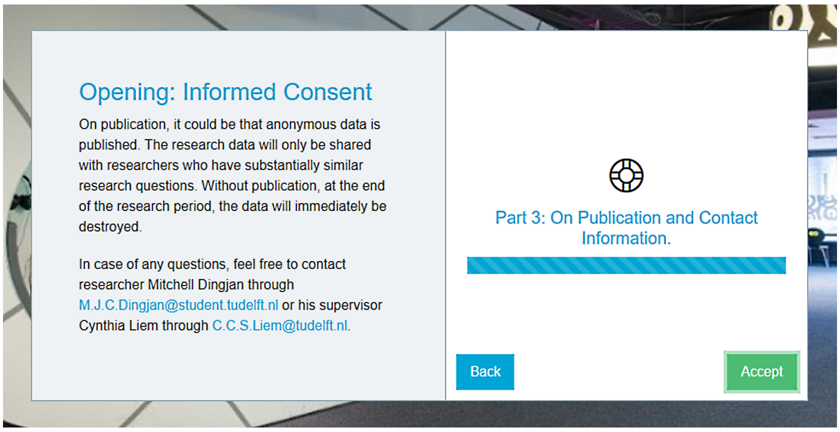


Opening: Informed Consent

Your participation in this study is entirely voluntary and you can withdraw at any time. You are free to omit any question.

From the questionnaire, only the answer provided for question **7** will be shared with Muziekweb. We believe there are no known risks associated with this research study; however, as with any online related activity the risk of a breach is always possible. To the best of our ability your answers in this study will remain confidential. We will minimize any risks by communicating through an https certified website, processing the data on a private server connected to a private database, and storing back-ups on a TU Delft assigned private storage.

Figure D.3: Informed Consent (2/3).

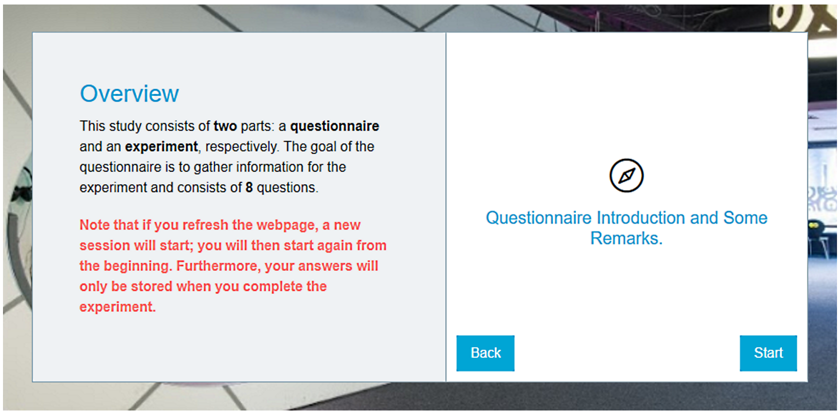


Opening: Informed Consent

On publication, it could be that anonymous data is published. The research data will only be shared with researchers who have substantially similar research questions. Without publication, at the end of the research period, the data will immediately be destroyed.

In case of any questions, feel free to contact researcher Mitchell Dingjan through M.J.C.Dingjan@student.tudelft.nl or his supervisor Cynthia Liem through C.C.S.Liem@tudelft.nl.

Figure D.4: Informed Consent (3/3).

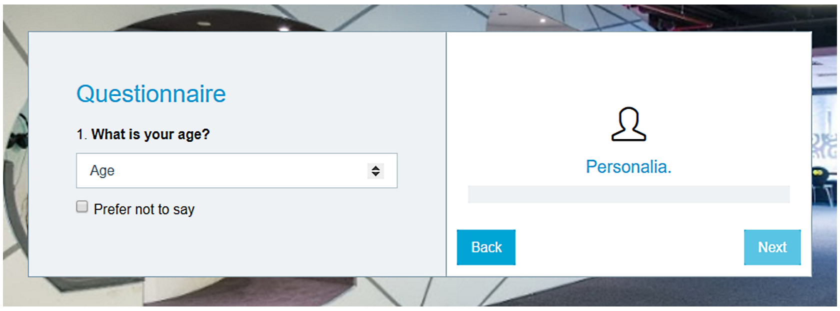


Overview

This study consists of **two** parts: a **questionnaire** and an **experiment**, respectively. The goal of the questionnaire is to gather information for the experiment and consists of **8** questions.

Note that if you refresh the webpage, a new session will start; you will then start again from the beginning. Furthermore, your answers will only be stored when you complete the experiment.

Figure D.5: Questionnaire Introduction and Some Remarks.



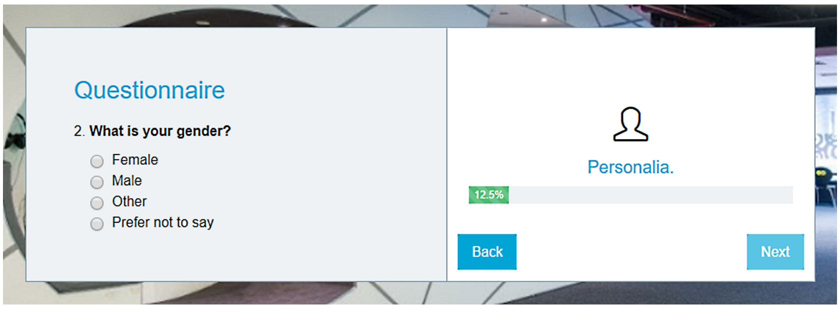
Questionnaire

1. What is your age?

Age

□ Prefer not to say

Figure D.6: Survey Question 1 - Age (Personalia).

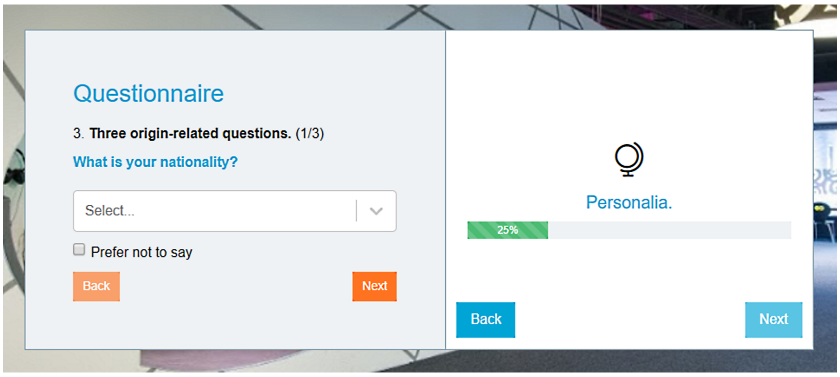


Questionnaire

2. What is your gender?

○ Female
○ Male
○ Other
○ Prefer not to say

Figure D.7: Survey Question 2 - Gender (Personalia).



Questionnaire

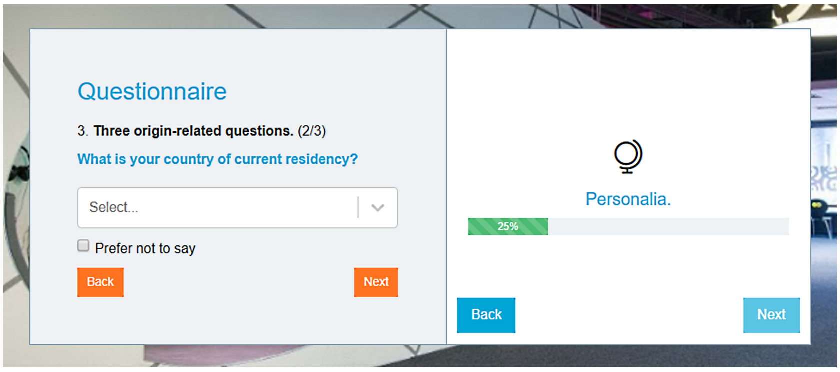
3. Three origin-related questions. (1/3)

What is your nationality?

Select...

□ Prefer not to say

Figure D.8: Survey Question 3a - Nationality (Personalia).



Questionnaire

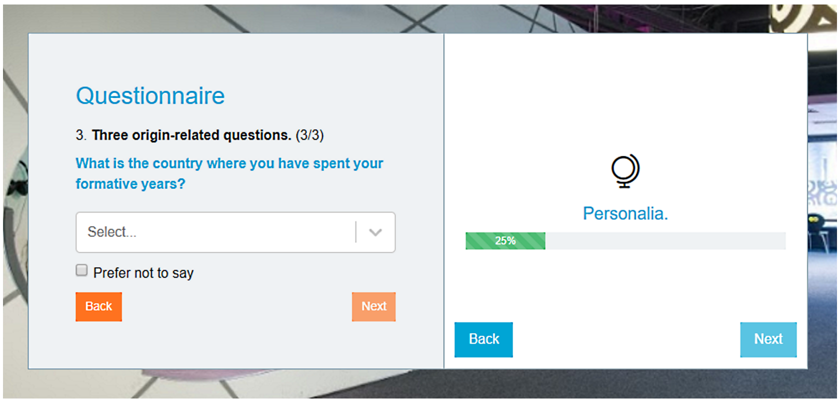
3. Three origin-related questions. (2/3)

What is your country of current residency?

Select...

□ Prefer not to say

Figure D.9: Survey Question 3b - Country of Current Residency (Personalia).



Questionnaire

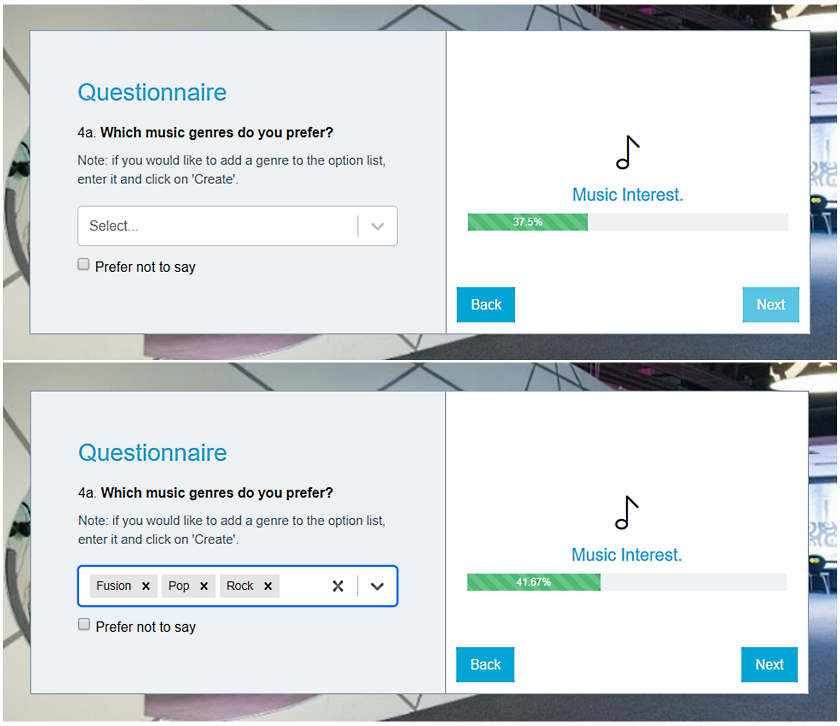
3. **Three origin-related questions.** (3/3)

What is the country where you have spent your formative years?

Select...

□ Prefer not to say

Figure D.10: Survey Question 3c - Country of Formative Years (Personalia).



Questionnaire

4a. **Which music genres do you prefer?**

Note: if you would like to add a genre to the option list, enter it and click on 'Create'.

Select...

□ Prefer not to say

Questionnaire

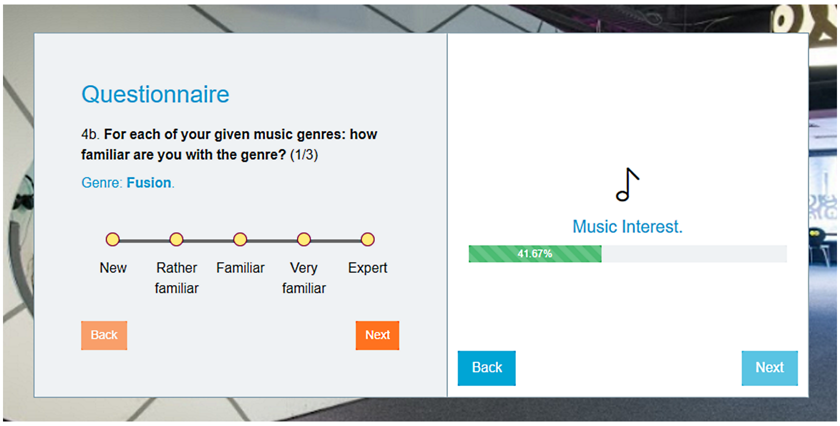
4a. **Which music genres do you prefer?**

Note: if you would like to add a genre to the option list, enter it and click on 'Create'.

Fusion × Pop × Rock ×

□ Prefer not to say

Figure D.11: Survey Question 4a - Preferred Music Genres (Music Interest).



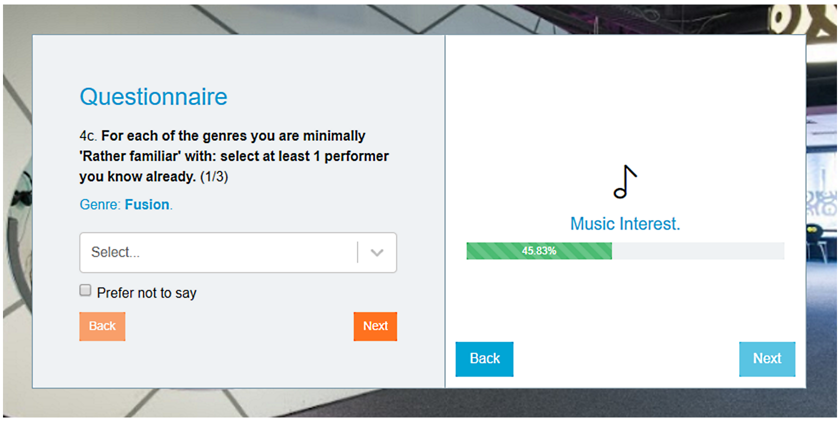
Questionnaire

4b. **For each of your given music genres: how familiar are you with the genre?** (1/3)

Genre: **Fusion**.

○ New ○ Rather familiar ○ Familiar ○ Very familiar ○ Expert

Figure D.12: Survey Question 4b - Familiarity With the Selected Genres (Music Interest).



Questionnaire

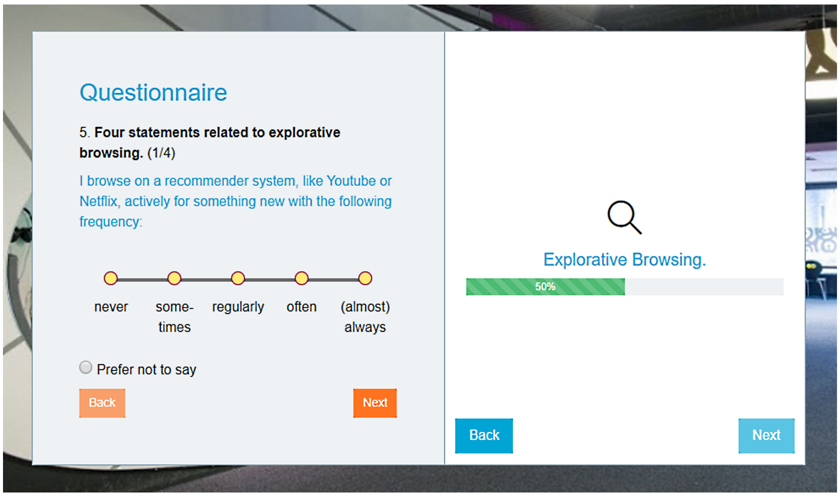
4c. **For each of the genres you are minimally 'Rather familiar' with: select at least 1 performer you know already.** (1/3)

Genre: **Fusion**.

Select...

□ Prefer not to say

Figure D.13: Survey Question 4c - Familiar Performers of the Selected Genres (Music Interest).



Questionnaire

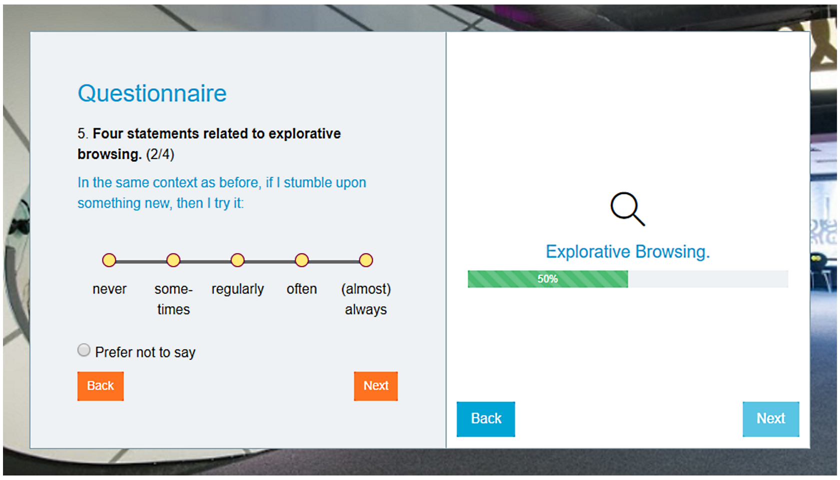
5. **Four statements related to explorative browsing.** (1/4)

I browse on a recommender system, like Youtube or Netflix, actively for something new with the following frequency:

○ never ○ some-times ○ regularly ○ often ○ (almost) always

○ Prefer not to say

Figure D.14: Survey Question 5a - Frequency Statement 1 (Explorative Browsing).



Questionnaire

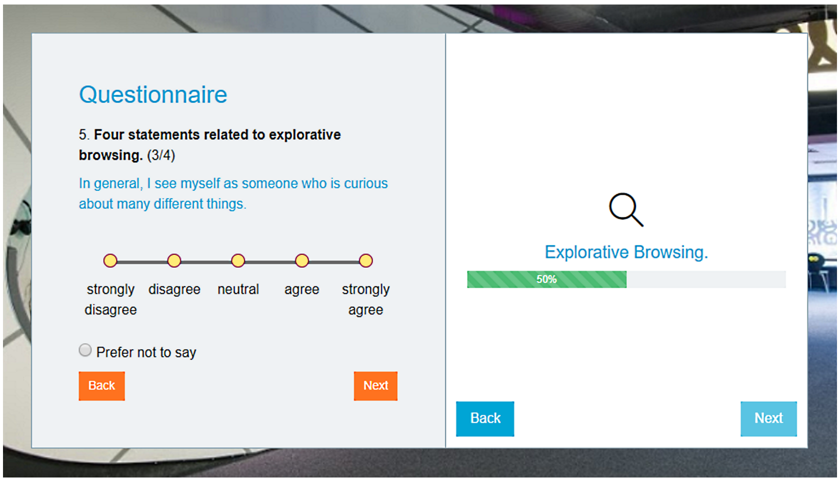
5. **Four statements related to explorative browsing.** (2/4)

In the same context as before, if I stumble upon something new, then I try it:

○ never ○ some-times ○ regularly ○ often ○ (almost) always

○ Prefer not to say

Figure D.15: Survey Question 5b - Frequency Statement 2 (Explorative Browsing).



Questionnaire

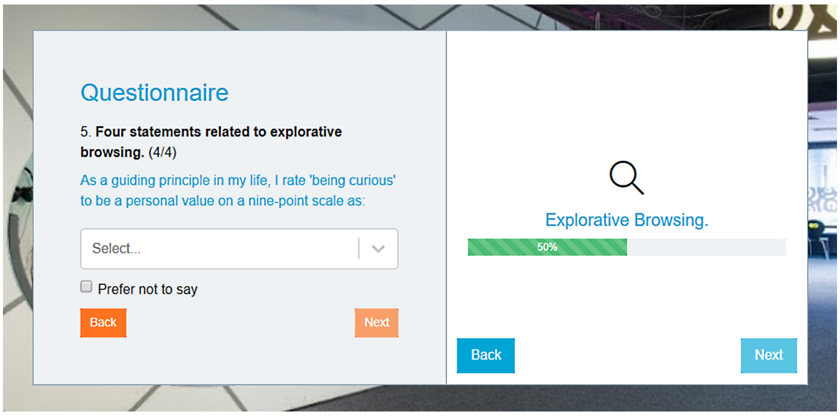
5. **Four statements related to explorative browsing.** (3/4)

In general, I see myself as someone who is curious about many different things.

○ strongly disagree ○ disagree ○ neutral ○ agree ○ strongly agree

○ Prefer not to say

Figure D.16: Survey Question 5c - Personality Trait Statement (Explorative Browsing).



Questionnaire

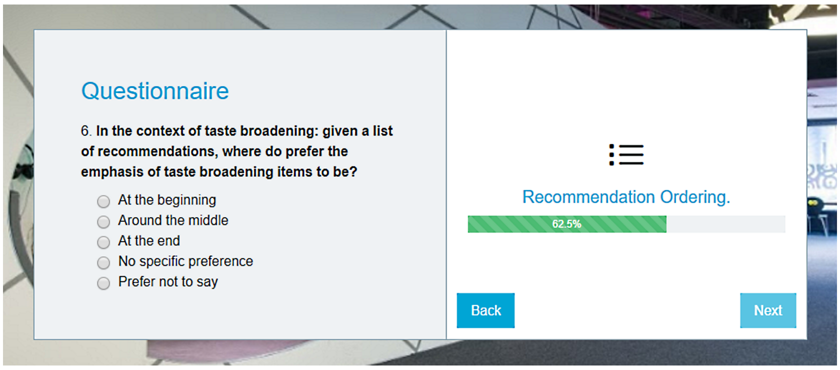
5. **Four statements related to explorative browsing.** (4/4)

As a guiding principle in my life, I rate 'being curious' to be a personal value on a nine-point scale as:

Select...

□ Prefer not to say

Figure D.17: Survey Question 5d - Personal Value Statement (Explorative Browsing).

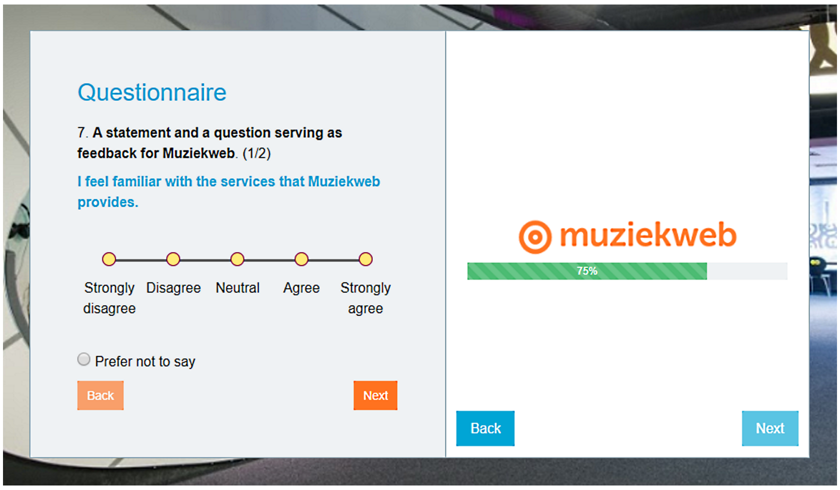


Questionnaire

6. **In the context of taste broadening: given a list of recommendations, where do prefer the emphasis of taste broadening items to be?**

○ At the beginning
○ Around the middle
○ At the end
○ No specific preference
○ Prefer not to say

Figure D.18: Survey Question 6 - Recommendation Ordering.



Questionnaire

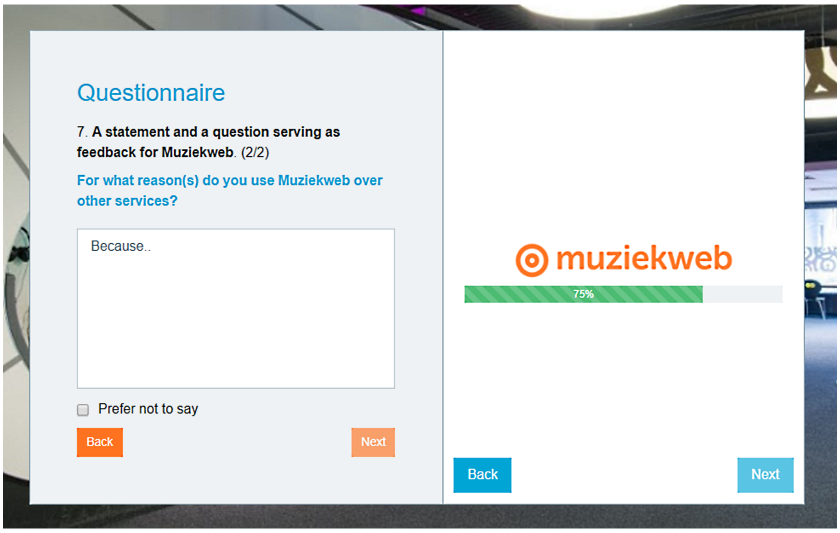
7. A statement and a question serving as feedback for Muziekweb. (1/2)

I feel familiar with the services that Muziekweb provides.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

○ Prefer not to say

Figure D.19: Survey Question 7a - Familiarity With Muziekweb (Feedback for Muziekweb).



Questionnaire

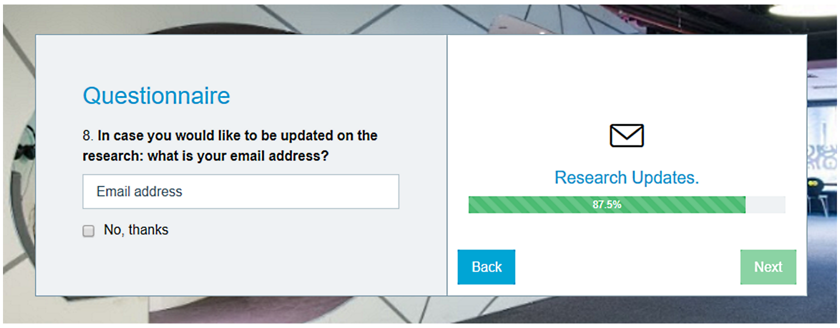
7. A statement and a question serving as feedback for Muziekweb. (2/2)

For what reason(s) do you use Muziekweb over other services?

Because..

□ Prefer not to say

Figure D.20: Survey Question 7b - Reasons for Using Muziekweb Over Other Services (Feedback for Muziekweb).



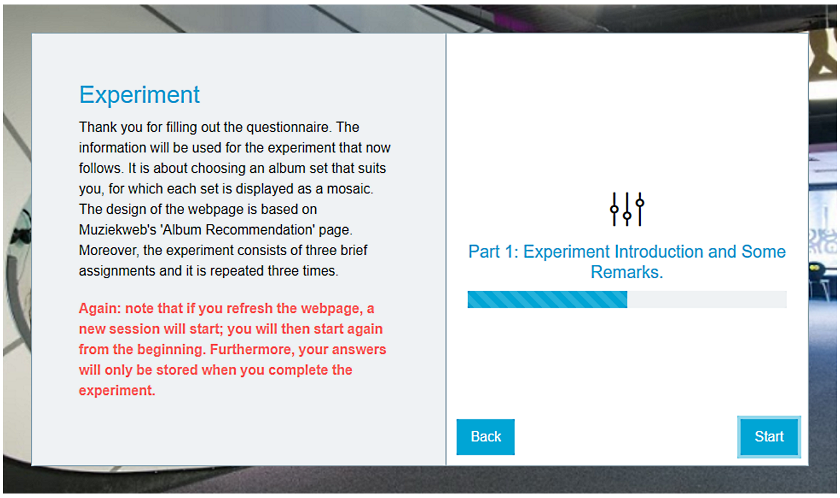
Questionnaire

8. In case you would like to be updated on the research: what is your email address?

Email address

□ No, thanks

Figure D.21: Survey Question 8 - Research Updates.

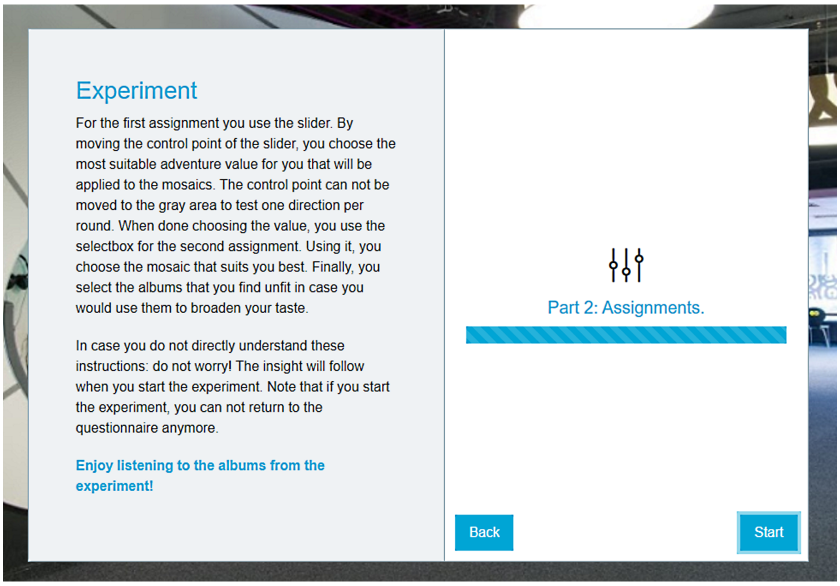


Experiment

Thank you for filling out the questionnaire. The information will be used for the experiment that now follows. It is about choosing an album set that suits you, for which each set is displayed as a mosaic. The design of the webpage is based on Muziekweb's 'Album Recommendation' page. Moreover, the experiment consists of three brief assignments and it is repeated three times.

Again: note that if you refresh the webpage, a new session will start; you will then start again from the beginning. Furthermore, your answers will only be stored when you complete the experiment.

Figure D.22: Part 1: Experiment Introduction and Some Remarks.



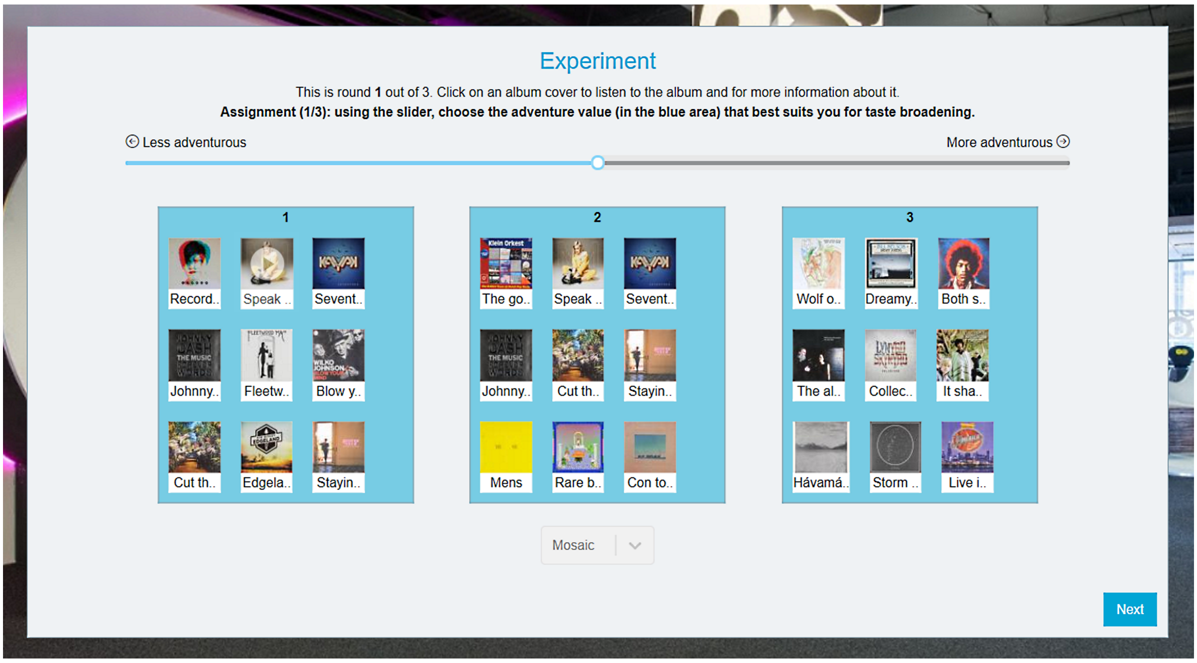
Experiment

For the first assignment you use the slider. By moving the control point of the slider, you choose the most suitable adventure value for you that will be applied to the mosaics. The control point can not be moved to the gray area to test one direction per round. When done choosing the value, you use the selectbox for the second assignment. Using it, you choose the mosaic that suits you best. Finally, you select the albums that you find unfit in case you would use them to broaden your taste.

In case you do not directly understand these instructions: do not worry! The insight will follow when you start the experiment. Note that if you start the experiment, you can not return to the questionnaire anymore.

Enjoy listening to the albums from the experiment!

Figure D.23: Part 2: Assignments.

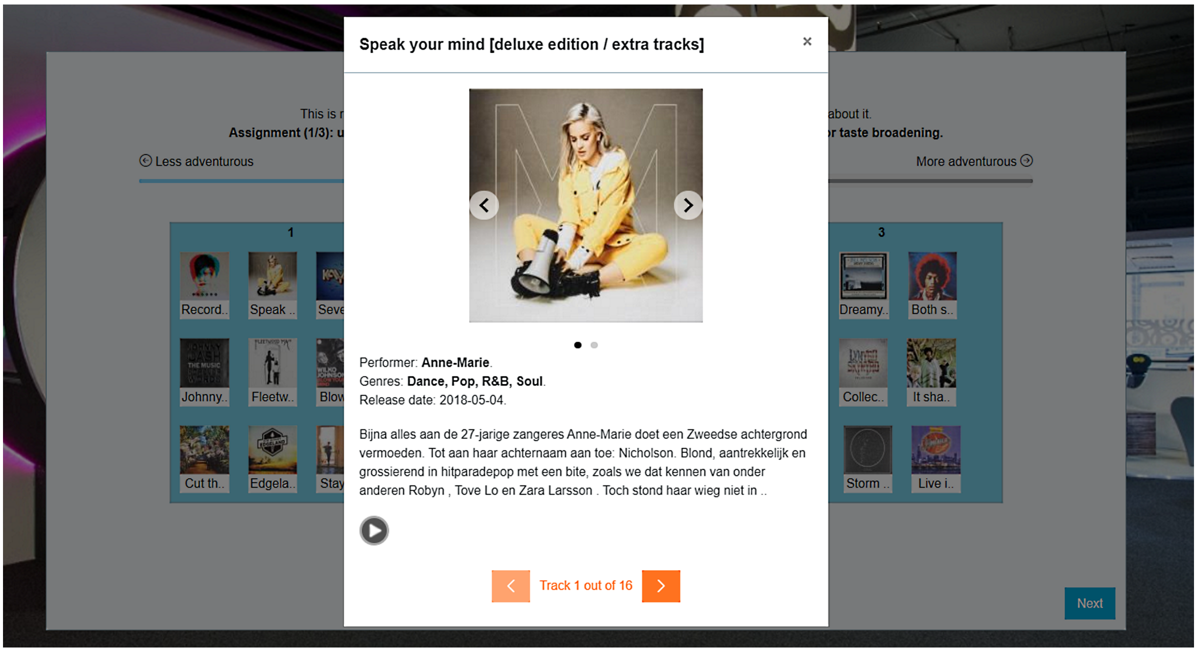


Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (1/3): using the slider, choose the adventure value (in the blue area) that best suits you for taste broadening.

Figure D.24: Experiment Assignment 1 out of 3 - Slider Value Selection (Round 1 out of 3).

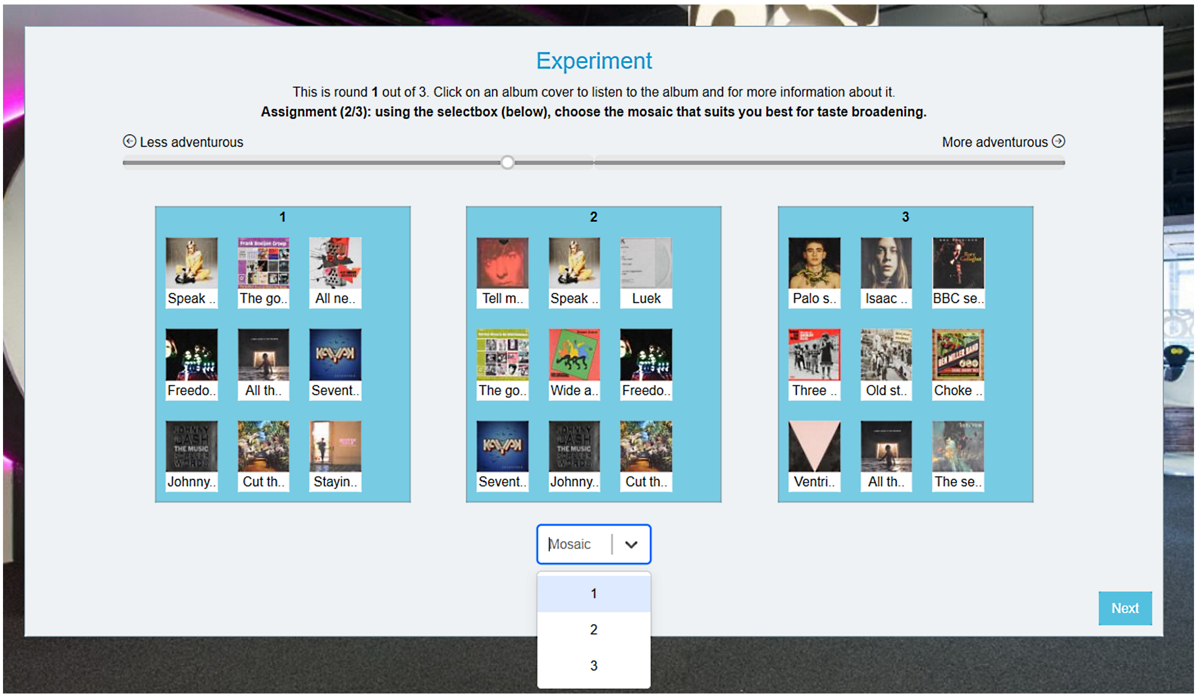


Speak your mind [deluxe edition / extra tracks]

Performer: **Anne-Marie**.
Genres: **Dance, Pop, R&B, Soul**.
Release date: 2018-05-04.

Bijna alles aan de 27-jarige zangeres Anne-Marie doet een Zweedse achtergrond vermoeden. Tot aan haar achternaam aan toe: Nicholson. Blond, aantrekkelijk en grossierend in hitparadepop met een bite, zoals we dat kennen van onder anderen Robyn , Tove Lo en Zara Larsson . Toch stond haar wieg niet in ..

Figure D.25: Album Modal Opened by Clicking on an Album.

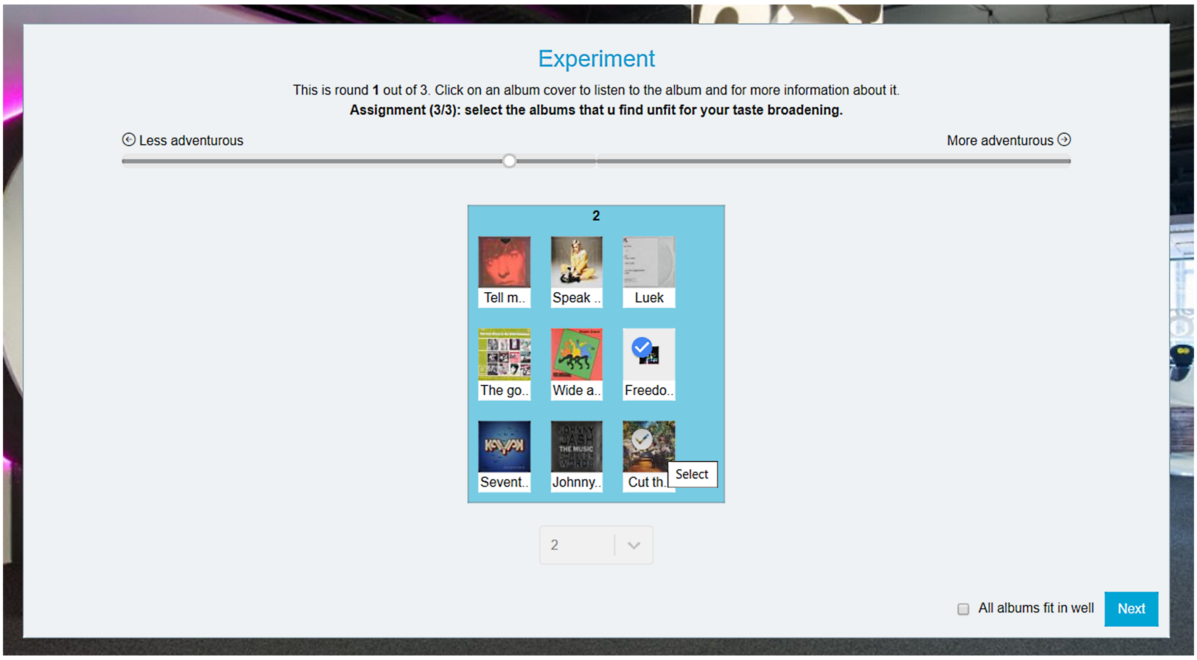


Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (2/3): using the selectbox (below), choose the mosaic that suits you best for taste broadening.

Figure D.26: Experiment Assignment 2 out of 3 - Mosaic Selection (Round 1 out of 3).

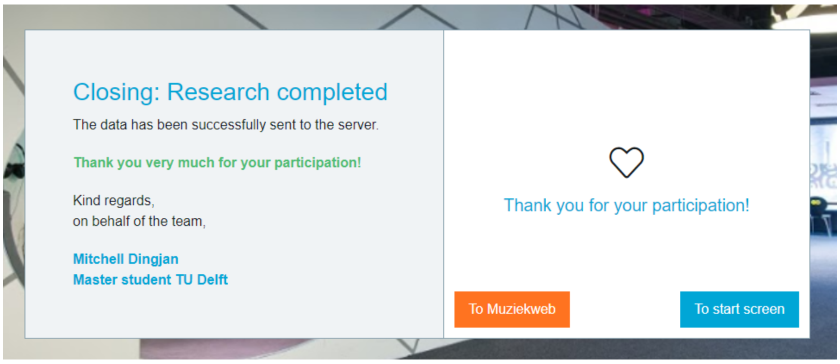


Experiment

This is round **1** out of 3. Click on an album cover to listen to the album and for more information about it.

Assignment (3/3): select the albums that u find unfit for your taste broadening.

Figure D.27: Experiment Assignment 3 out of 3 - Unfit Album Selection (Round 1 out of 3).



Closing: Research completed

The data has been successfully sent to the server.

Thank you very much for your participation!

Kind regards,
on behalf of the team,

Mitchell Dingjan
Master student TU Delft

Thank you for your participation!

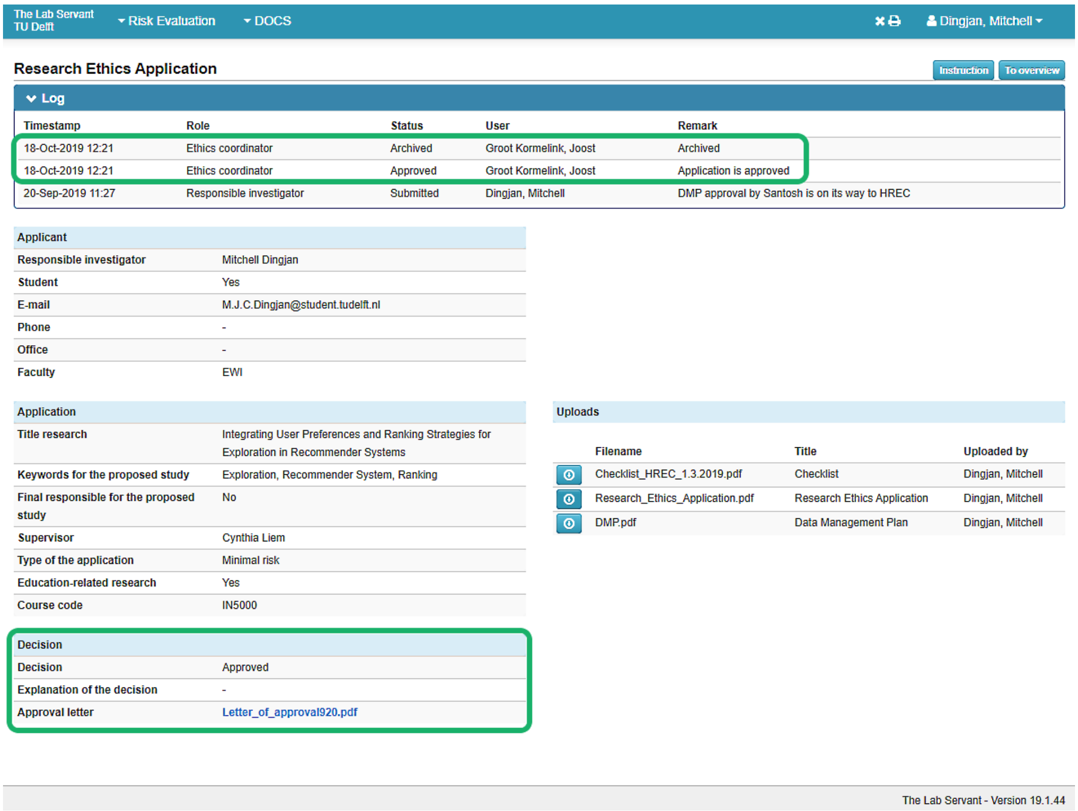
Figure D.28: Closure on Completion of Round 3.

(This page has been intentionally left blank.)

E

HREC Approval

This appendix contains the research approval by TU Delft's HREC, considering our quantitative online user study. See Figure E.1; the relevant parts of the figure that state their approval have been marked.



Research Ethics Application

Timestamp	Role	Status	User	Remark
18-Oct-2019 12:21	Ethics coordinator	Archived	Groot Kormelink, Joost	Archived
18-Oct-2019 12:21	Ethics coordinator	Approved	Groot Kormelink, Joost	Application is approved
20-Sep-2019 11:27	Responsible investigator	Submitted	Dingjan, Mitchell	DMP approval by Santosh is on its way to HREC

Applicant	
Responsible investigator	Mitchell Dingjan
Student	Yes
E-mail	M.J.C.Dingjan@student.tudelft.nl
Phone	-
Office	-
Faculty	EWI

Application	
Title research	Integrating User Preferences and Ranking Strategies for Exploration in Recommender Systems
Keywords for the proposed study	Exploration, Recommender System, Ranking
Final responsible for the proposed study	No
Supervisor	Cynthia Liem
Type of the application	Minimal risk
Education-related research	Yes
Course code	IN5000

Uploads

Filename	Title	Uploaded by
Checklist_HREC_1.3.2019.pdf	Checklist	Dingjan, Mitchell
Research_Ethics_Application.pdf	Research Ethics Application	Dingjan, Mitchell
DMP.pdf	Data Management Plan	Dingjan, Mitchell

Decision	
Decision	Approved
Explanation of the decision	-
Approval letter	Letter_of_approval920.pdf

The Lab Servant - Version 19.1.44

Figure E.1: Research Approval by HREC Through The Lab Servant.

(This page has been intentionally left blank.)

Bibliography

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2312–2320, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

[2] ACM. CHIIR 2020's call for contributions poster, 2019. URL http://sigir.org/chiir2020/cfp.pdf. [Online; accessed 2 September 2019].

[3] ACM. RecSys 2019's call for contributions page, 2019. URL https://recsys.acm.org/recsys19/call/. [Online; accessed 1 August 2019].

[4] ACM. SIGIR 2020's call for contributions page (full paper version), 2019. URL https://sigir.org/sigir2020/call-for-full-papers/. [Online; accessed 1 December 2019].

[5] Panagiotis Adamopoulos and Vilma Todri. Personality-based recommendations: Evidence from amazon.com. In *9th ACM Conference on Recommender Systems Poster Proceedings*, United States, 2015. Association for Computing Machinery (ACM).

[6] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control Optim.*, 33(6):1926–1951, November 1995. ISSN 0363-0129. doi: 10.1137/S0363012992237273. URL https://doi.org/10.1137/S0363012992237273.

[7] Rajeev Agrawal. Sample mean based index policies by o(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995. doi: 10.2307/1427934.

[8] Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard. The non-stationary stochastic multi-armed bandit problem. *International Journal of Data Science and Analytics*, 3(4):267–283, Jun 2017. ISSN 2364-4168. doi: 10.1007/s41060-017-0050-5. URL https://doi.org/10.1007/s41060-017-0050-5.

[9] Kumaripaba Athukorala, Alan Medlar, Antti Oulasvirta, Giulio Jacucci, and Dorota Glowacka. Beyond relevance: Adapting exploration/exploitation in information retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, IUI '16, pages 359–369, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4137-0. doi: 10.1145/2856767.2856786. URL http://doi.acm.org/10.1145/2856767.2856786.

[10] Peter Auer and Chao-Kai Chiang. An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 116–120, 2016.

[11] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, page 322, USA, 1995. IEEE Computer Society. ISBN 0818671831.

[12] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002. ISSN 0885-6125. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

[13] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003. ISSN 0097-5397. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

[14] Adelchi Azzalini and Antonella Capitanio. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):579–602, Aug 1999. ISSN 1467-9868. doi: 10.1111/1467-9868.00194. URL http://dx.doi.org/10.1111/1467-9868.00194.

[15] Adelchi Azzalini and Alessandra Dalla Valle. The multivariate skew-normal distribution. *Biometrika*, 83(4):715–726, 12 1996. ISSN 0006-3444. doi: 10.1093/biomet/83.4.715. URL https://doi.org/10.1093/biomet/83.4.715.

[16] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012. URL http://arxiv.org/abs/1204.5721.

[17] Iván Cantador, Ignacio Fernández-Tobías, Alejandro Bellogín, Michal Kosinski, and David Stillwell. Relating personality types with user preferences multiple entertainment domains. In *CEUR workshop proceedings*, volume 997, 01 2013.

[18] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, May 1997. ISSN 0004-5411. doi: 10.1145/258128.258179. URL https://doi.org/10.1145/258128.258179.

[19] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 151–159, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/chen13a.html.

[20] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 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, pages 659–666, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390446. URL http://doi.acm.org/10.1145/1390334.1390446.

[21] Richard Combes, Stefan Magureanu, and Alexandre Proutiere. Minimal exploration in structured stochastic bandits. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1761–1769, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[22] Paul Costa and Robert McCrae. The revised neo personality inventory (neo-pi-r). *The SAGE Handbook of Personality Theory and Assessment*, 2:179–198, 01 2008. doi: 10.4135/9781849200479.n9.

[23] John M. Digman. Personality structure: Emergence of the five-factor model. *Annual Review of Psychology*, 41(1):417–440, 1990. doi: 10.1146/annurev.ps.41.020190.002221. URL https://doi.org/10.1146/annurev.ps.41.020190.002221.

[24] Soude Fazeli, Hendrik Drachsler, Marlies Bitter-Rijpkema, Francis Brouns, Wim v. d. Vegt, and Peter B. Sloep. User-centric evaluation of recommender systems in social learning platforms: Accuracy is just the tip of the iceberg. *IEEE Transactions on Learning Technologies*, 11(3):294–306, July 2018. ISSN 2372-0050. doi: 10.1109/TLT.2017.2732349.

[25] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504. URL https://doi.org/10.1006/jcss.1997.1504.

[26] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, pages 1–9, April 2010. doi: 10.1109/DYSPAN.2010.5457857.

[27] Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 218–227. JMLR.org, 2015.

[28] Nicolas Galichet. *Contributions to Multi-Armed Bandits : Risk-Awareness and Sub-Sampling for Linear Contextual Bandits*. Theses, Université Paris Sud - Paris XI, September 2015. URL https://tel.archives-ouvertes.fr/tel-01277170.

[29] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems, 2008.

[30] Sébastien Gerchinovitz and Tor Lattimore. Refined lower bounds for adversarial bandits. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1198–1206, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[31] Jennifer Golbeck and Eric Norris. Personality, movie preferences, and recommendations. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 1414–1415, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2240-9. doi: 10.1145/2492517.2492572. URL http://doi.acm.org/10.1145/2492517.2492572.

[32] Frédéric Guillou. *On Recommendation Systems in a Sequential Context.* Theses, Université Lille 3, December 2016. URL https://tel.archives-ouvertes.fr/tel-01407336.

[33] Alan Hanjalic. New grand challenge for multimedia information retrieval: bridging the utility gap. *International Journal of Multimedia Information Retrieval*, 1, October 2012. doi: 10.1007/s13735-012-0019-z.

[34] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 144–150. AAAI Press, 2016. URL http://dl.acm.org/citation.cfm?id=3015812.3015834.

[35] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 355–364, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3209981. URL http://doi.acm.org/10.1145/3209978.3209981.

[36] Katja Hofmann, Shimon Whiteson, and Maarten Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Inf. Retr.*, 16(1):63–90, February 2013. ISSN 1386-4564. doi: 10.1007/s10791-012-9197-9. URL http://dx.doi.org/10.1007/s10791-012-9197-9.

[37] Abid Hussain and Yousaf Shad Muhammad. Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. *Complex & Intelligent Systems*, Apr 2019. ISSN 2198-6053. doi: 10.1007/s40747-019-0102-7. URL https://doi.org/10.1007/s40747-019-0102-7.

[38] Nicole Immorlica, Karthik A. Sankararaman, Robert Schapire, and Aleksandrs Slivkins. Adversarial bandits with knapsacks. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 202–219, Nov 2019. doi: 10.1109/FOCS.2019.00022.

[39] Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 325–333, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[40] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu. Adversarial attacks on stochastic bandits. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3644–3653, Red Hook, NY, USA, 2018. Curran Associates Inc.

[41] Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7 (1):2:1–2:42, December 2016. ISSN 2160-6455. doi: 10.1145/2926720. URL http://doi.acm.org/10.1145/2926720.

[42] Peter Knees, Markus Schedl, Bruce Ferwerda, and Audrey Laplante. User awareness in music recommender systems. In *Personalized human-computer interaction*, De Gruyter Textbook. De Gruyter Open, 2019. ISBN 9783110552485.

[43] Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem, 2015.

[44] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems – a survey. *Knowledge-Based Systems*, 123:154 – 162, 2017. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2017.02.009. URL http://www.sciencedirect.com/science/article/pii/S0950705117300680.

[45] Audrey Laplante. Improving music recommender systems: What can we learn from research on music tastes? In *ISMIR*, pages 451–456, 2014.

[46] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press (preprint), 2019.

[47] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 661–670, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772758. URL https://doi.org/10.1145/1772690.1772758.

[48] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 539–548, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2911548. URL http://doi.acm.org/10.1145/2911451.2911548.

[49] Yu Liang and Martijn C. Willemsen. Personalized recommendations for music genre exploration. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '19, pages 276–284, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6021-0. doi: 10.1145/3320435.3320455. URL http://doi.acm.org/10.1145/3320435.3320455.

[50] Huafeng Liu, Jingxuan Wen, Liping Jing, and Jian Yu. Deep generative ranking for personalized recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 34–42, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6243-6. doi: 10.1145/3298689.3347012. URL http://doi.acm.org/10.1145/3298689.3347012.

[51] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 114–122, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599. doi: 10.1145/3188745.3188918. URL https://doi.org/10.1145/3188745.3188918.

[52] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bounds and optimal algorithms, 2014.

[53] Sandy Manolios, Alan Hanjalic, and Cynthia C. S. Liem. The influence of personal values on music taste: Towards value-based music recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 501–505, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6243-6. doi: 10.1145/3298689.3347021. URL http://doi.acm.org/10.1145/3298689.3347021.

[54] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: Personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 31–39, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5901-6. doi: 10.1145/3240323.3240354. URL http://doi.acm.org/10.1145/3240323.3240354.

[55] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, page 1097–1101, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932984. doi: 10.1145/1125451.1125659. URL https://doi.org/10.1145/1125451.1125659.

[56] Muziekweb. Welcome page, 2019. URL https://www.muziekweb.eu/en/Muziekweb/Informatie/Welkom. [Online; accessed 3 September 2019].

[57] Muziekweb. Organization and policy page, 2019. URL https://www.muziekweb.eu/en/Muziekweb/Informatie/OverDeCDR. [Online; accessed 3 September 2019].

[58] Muziekweb. Muziekweb music advice page, 2020. URL https://www.muziekweb.nl/en/Muziekweb/Advies/. [Online; accessed 28 February 2020].

[59] Muziekweb. Muziekweb intro's page, 2020. URL https://www.muziekweb.nl/en/Muziekweb/Intros. [Online; accessed 28 February 2020].

[60] Muziekweb. Muziekweb playlist page, 2020. URL https://www.muziekweb.nl/en/Muziekweb/Playlist. [Online; accessed 28 February 2020].

[61] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 677–686, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2568012. URL http://doi.acm.org/10.1145/2566486.2568012.

[62] Santiago Ontañón. Combinatorial multi-armed bandits for real-time strategy games. *J. Artif. Int. Res.*, 58(1):665–702, January 2017. ISSN 1076-9757.

[63] Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin UK, 2011. ISBN 1594203008, 9781594203008.

[64] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, and Dan Pei. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 3–11, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6243-6. doi: 10.1145/3298689.3347000. URL http://doi.acm.org/10.1145/3298689.3347000.

[65] Bruno L. Pereira, Alberto Ueda, Gustavo Penha, Rodrygo L. T. Santos, and Nivio Ziviani. Online learning to rank for sequential music recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 237–245, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6243-6. doi: 10.1145/3298689.3347019. URL http://doi.acm.org/10.1145/3298689.3347019.

[66] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 784–791, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390255. URL http://doi.acm.org/10.1145/1390156.1390255.

[67] Anshuka Rangi, Massimo Franceschetti, and Long Tran-Thanh. Unifying the stochastic and the adversarial bandits with knapsack. *CoRR*, abs/1811.12253, 2018. URL http://arxiv.org/abs/1811.12253.

[68] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press. ISBN 978-0-9749039-5-8. URL http://dl.acm.org/citation.cfm?id=1795114.1795167.

[69] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. Springer Publishing Company, Incorporated, 2nd edition, 2015. ISBN 1489976361, 9781489976369.

[70] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. URL http://www.projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.bams/1183517370.

[71] Sonia Roccas, Lilach Sagiv, Shalom Schwartz, and Ariel Knafo-Noam. The big five personality factors and personal values. *Personality and Social Psychology Bulletin*, 28:789–801, 06 2002. doi: 10.1177/0146167202289008.

[72] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Found. Trends Mach. Learn.*, 11(1):1–96, July 2018. ISSN 1935-8237. doi: 10.1561/2200000070. URL https://doi.org/10.1561/2200000070.

[73] Javier Sanz-Cruzado, Pablo Castells, and Esther López. A simple multi-armed nearest-neighbor bandit for interactive recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 358–362, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6243-6. doi: 10.1145/3298689.3347040. URL http://doi.acm.org/10.1145/3298689.3347040.

[74] Shalom H. Schwartz. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In Mark P. Zanna, editor, *Advances in Experimental Social Psychology*, volume 25, pages 1 – 65. Academic Press, 1992. doi: https://doi.org/10.1016/S0065-2601(08)60281-6. URL http://www.sciencedirect.com/science/article/pii/S0065260108602816.

[75] Steven L. Scott. A modern bayesian look at the multi-armed bandit. *Appl. Stoch. Model. Bus. Ind.*, 26(6): 639–658, November 2010. ISSN 1524-1904. doi: 10.1002/asmb.874. URL https://doi.org/10.1002/asmb.874.

[76] Aleksandrs Slivkins. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272, 2019. URL http://arxiv.org/abs/1904.07272.

[77] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* A Bradford Book, USA, 2018. ISBN 0262039249, 9780262039246.

[78] Liang Tang, Yexi Jiang, Lei Li, and Tao Li. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, page 73–80, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450326681. doi: 10.1145/2645710.2645732. URL https://doi.org/10.1145/2645710.2645732.

[79] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *j-BIOMETRIKA*, 25(3/4):285–294, December 1933. ISSN 0006-3444 (print), 1464-3510 (electronic). doi: https://doi.org/10.2307/2332286. URL http://www.jstor.org/stable/2332286.

[80] Aristide C. Y. Tossou, Christos Dimitrakakis, and Devdatt Dubhashi. Thompson sampling for stochastic bandits with graph feedback. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2660–2666. AAAI Press, 2017.

[81] Alexandra L. Uitdenbogerd and Ron G. van Schyndel. A review of factors affecting music recommender success. In *ISMIR 2002, 3rd International Conference on Music Information Retrieval*, pages 204–208. IRCAM-Centre Pompidou, 2002.

[82] Hastagiri P. Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 225–232, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645733. URL http://doi.acm.org/10.1145/2645710.2645733.

[83] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 109–116, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0683-6. doi: 10.1145/2043932.2043955. URL http://doi.acm.org/10.1145/2043932.2043955.

[84] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.*, 45(3):35:1–35:33, July 2013. ISSN 0360-0300. doi: 10.1145/2480741.2480752. URL http://doi.acm.org/10.1145/2480741.2480752.

[85] Huazheng Wang, Sonwoo Kim, Eric McCord-Snook, Qingyun Wu, and Hongning Wang. Variance reduction in gradient exploration for online learning to rank. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 835–844, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6172-9. doi: 10.1145/3331184.3331264. URL http://doi.acm.org/10.1145/3331184.3331264.

[86] Huasen Wu and Xin Liu. Double thompson sampling for dueling bandits. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 649–657, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[87] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538 – 1556, 2012. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2011.12.028. URL http://www.sciencedirect.com/science/article/pii/S0022000012000281. JCSS Special Issue: Cloud Computing 2011.

[88] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–10–II–18. JMLR.org, 2014.

[89] Masrour Zoghi, Zohar Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 307–315, Cambridge, MA, USA, 2015. MIT Press.