MASTER

Evaluating a framework for sequential group music recommendations
a modular framework for dynamic fairness and coherence control

Hadash, Sophia

Award date:
2019

Technische Universiteit
Eindhoven
University of Technology

Department of Industrial Engineering & Innovation Sciences
Human-Technology Interaction Research Group

Evaluating a framework for sequential group music recommendations

A Modular Framework for Dynamic Fairness and Coherence control

Sophia Hadash

Supervisors:
Dr. ir. M.C. Willemsen, Eindhoven University of Technology
Dr. N. Tintarev, Delft University of Technology

v1.0.1

Eindhoven, February 2019

Abstract

In this thesis a theoretical and modular framework is proposed for sequential group music recommendation systems. The primary contribution of this framework is the addition of the satisfaction function module and track weighting module to the classical approach to group recommendation systems. The satisfaction function applies user weighting and is used for increasing the fairness of the recommender while the track weighting function applies track weighting and is used for reaching target characteristics of the playlist.

An implementation of a group music recommendation system was developed based on the framework and its modules. Four recommendation strategy module implementations were described and shown to accurately generate predictions for individuals while solving the so called disjoint set problem that occurs during group aggregation when the individual predictions are disjoint.

The usefulness of the satisfaction function module was evaluated using a focus group study. The study showed that the satisfaction function influences perceived fairness of the system and that affective state modelling in the satisfaction function can be used to increase its perceived fairness.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A recommendation system (RS) is a specific type of decision support or advice-giving system that guides users in a personalized way towards interesting options, where a large pool of such options are available (Felfernig, Boratto, Stettinger, & Tkalčič, 2018). It comes as no surprise that in the age of the Internet and streaming recommendation systems are applied to music. With access to huge collections of music it can be quite challenging to find the music you like. While today's music recommendation systems (MRS) are quite able to help users find their music, most MRS are focused on recommending single items to individual persons (e.g. Felfernig, Boratto, et al., 2018).

However, music is often not consumed by individuals but rather by groups of people. Examples of such scenarios are when friends put on music during a social gathering, when music is played in shops, or when a musical performance is presented to a group of people. The social aspect of music makes group music recommendation systems an interesting and valuable topic with many practical applications (e.g. Felfernig, Boratto, et al., 2018; Boratto & Carta, 2010; Masthoff, 2015).

Another aspect that makes music recommendation peculiar is related to the fact that musical items are quite short in duration. People compose playlists, often in an artistic fashion, such that music can be enjoyed over a longer duration. Consequently, practical music recommendation systems have to recommend sequences (i.e. playlists) instead of individual tracks. In such recommendation systems the holistic experience of the playlist ultimately determines whether the recommendation system is valuable to its users. This experience is not based on the simple sum of the relevance of its individual musical items, but is rather shaped by both the individual items and their sequence order. It is thus very valuable to understand how music recommendation systems can make use of sequence order in their recommendations.

Although both group music recommendation and sequential music recommendation are valuable topics in itself, practical applications aimed at groups of people are generally dealing with the combination. This combination is not straightforward and contains certain unique difficulties. For instance, psychological effects that are relevant in determining good sequence orders are often described at the individual level in sequential music recommendation research while these may work differently in group settings. It is thus important to study how theory from both group music recommendation and sequential music recommendation can be combined.

In this thesis theory and insights from group music recommendation and sequential music recommendation research are combined. Based on this combination a general modular framework for group music recommendation systems is proposed. An actual implementation of this framework is developed which makes use of Spotify's music library and user profiles to show its applicability and usefulness. To allow commercial user profile data to be effectively incorporated in the framework, four methodologies for determining individual item relevance based on limited knowledge of user preferences are described. In summary, we propose the following research question:

Research Question *How can theory and insights from sequential and group music recommendation systems structurally be combined such that they can be applied to practical group music* *recommendation systems.*

Chapter 2

Sequential Music Recommendation

Most of the music that people listen to is organized in a playlist. A playlist is defined as "a sequence of tracks (audio recordings)". This definition, as provided by Bonnin and Jannach (2014), corresponds to what is most commonly used in the literature. Traditionally playlists were created manually by disk jockeys or music producers. In the 80's the rising popularity of the mix tape allowed ordinary people to create their own playlists. Since then technological advances like the compact disc, mp3, and more recently streaming made the creation of personalized playlists easier and more accessible. Recommender systems can help people in the playlist creation process by either recommending a whole playlist, as in Automatic Playlist Generation (APG), or by dynamically extending an existing playlist, as in Automatic Playlist Continuation (APC).

The automatic playlist generation problem can crudely be summarized as the task of finding a useful sequence of tracks given some knowledge of the context and users. Automatic playlist continuation can be seen a special case of automatic playlist generation where the existing playlist serves as additional background knowledge (Schedl, Zamani, Deldjoo, Elahi, & Chen, 2018) and it is thus not particularly relevant for understanding sequential music recommendation. A summary of the challenges and state-of-the-art in the automatic playlist generation problem, on the other hand, is useful to understanding sequential music recommendation because it places the role of sequential music recommendation in perspective.

2.1 The automatic playlist generation problem

The automatic playlist generation problem is described as the following task: Given a set of available tracks (1), background knowledge (2), and a target characteristic (3), find a sequence of tracks that fulfills the target characteristic as good as possible (4) (Bonnin & Jannach, 2014). The following sections will elaborate on each of the components of the APG problem.

2.1.1 The set of tracks

While many commercial organizations have access to large libraries of music (e.g. Spotify, Last.fm, Pandora) (Kaminskas & Ricci, 2012), it is not a straightforward task to obtain access to large music libraries for academic purposes. Since the goal of most research in music recommendation systems is to generate personalized recommendations, a large enough pool of tracks is required in order to allow for personalization. Although this thesis will show how Spotify's music library can be used in academic research, a comprehensive summary of how sufficiently large track libraries can be obtained for academic purposes is outside the scope of this thesis.

2.1.2 Background knowledge

The next step in the APG problem is to obtain background knowledge on the musical pieces of the track library. Methods for obtaining this knowledge can be categorized as follows: musical features extracted from the audio signals; metadata and expert opinions; social web data like tags, ratings, and social graphs (for an application, see Konstas, Stathopoulos, & Jose, 2009); and usage data like popularity, listening behaviour (clicks, skip operations), and manually created playlists (Bonnin & Jannach, 2014).

When libraries of commercial platforms are used in academic research, certain forms of background knowledge are often readily available through the platform in use (e.g., Jehan & Desroches, 2004). While the accessibility to this data is beneficial, there are disadvantages to using it as well. Most importantly, it is usually not clear how the data is obtained. Commercial platforms often do not report on details of how metadata is generated or obtained. This has a negative impact on the reproducibility and external validity of academic research that uses metadata from such platforms.

2.1.3 Target characteristics

The third subtask of the APG problem entails identifying the target characteristics of the playlist. An important characteristic that is usually determined by the user(s) is the purpose of the playlist. Playlists can for instance be created for listening while studying, doing sports, relaxing, or partying. A playlist intended for studying obviously is quite different than one created for a party. While many playlists are created for listening purposes, playlists are also created for purposes like discovery and collection (Schedl, Knees, & Gouyon, 2017).

In their review Bonnin and Jannach (2014) identified three types of identifying target characteristics: Explicit preferences and constraints, past user preferences, and contextual and sensor information. We split contextual and sensor information into situational and psychological/physiological characteristics because both context and physiological state can distinctly be used for determining target characteristics.

Explicit preferences and constraints are short-term preferences set by the user(s) that the playlist must contain. They can be captured by asking the user for seed tracks or keywords but also by directly asking the user to set a preferred mood, genre, or context. Explicit preferences can also be obtained in the form of predefined checklists or by using real-time user feedback.

Past user preferences can be a broad range of types of information. Examples of information types are past ratings, user libraries of music, previously created playlists, and the user profile of a recommender system. Past user preferences should be seen as a long-term profile of the user containing information about general user preferences and stands in contrast to short-term, explicit preferences.

Another important aspect of the target characteristics of a playlist is the current or future context. Examples of contextual variables are location, time, activity, weather, and listening style (e.g. stereo, headplugs). Additionally, contextual variables can be the current social context (Schedl et al., 2018) or the session-based purpose of the users. Certain contextual characteristics may be more complicated to incorporate in a recommendation system compared to others. For instance location, time, and weather can implicitly be obtained (given that the user gave consent for using such data) while the current social context or purpose of the user(s) is subject to prediction or should be determined explicitly. For an elaborate overview of challenges in contextual recommendations, see Kaminskas and Ricci (2012).

The last type of target characteristic is psychological information. The user's psychological or physiological state can be incorporated as a target characteristic for a playlist. For example, Van Der Zwaag, Janssen, and Westerink (2013) used a skin-conductance sensor as a measurement for the user's mood and used it successfully to recommend sets of songs that changed the user's mood over time either positively, negatively, or dynamically neutral. This demonstrates that the purpose of a playlist can very well be related to a user's psychological or physiological state (for a short overview, see Schedl et al. (2018)).

2.1.4 Finding a good sequence

After a track library, background knowledge, and target characteristics are obtained the final task in the automatic playlist generation problem is to find a 'good' sequence. This is a challenging task that involves multiple subtasks. First of all the question of what a 'good' sequence is should be addressed. In this thesis the term 'playlist quality' is used as the umbrella term for describing the components that lead to a good playlist. The second step is comprised of employing certain algorithms for generating a playlist given the track library, background knowledge, and target characteristics. The final step is the evaluation of the generated playlist.

Playlist quality and the role of sequence order

To solve the APG Problem a sequence has to be found that fulfills the target characteristics as good as possible. However, a sequence is not merely a set of individually recommended items, but has an inherent sequential structure. Some tracks transition smoothly into each other, while some clearly do not (e.g. a popular pop song to a slow classical piece). But what determines which transitions are smooth and which are not? Hansen and Golbeck (2009) performed an empirical user study on this topic. They described three aspects of a playlist that make up for its quality: individual item value, co-occurrence interaction effects, and order interaction effects. Since individual item value is extensively studied in traditional MRS research (e.g., Felfernig, Boratto, et al., 2018) it is not discussed in this section.

Co-occurrence interaction effects are effects that either increase or decrease the value of the pair (or set) of tracks independent of the individual values. This can be illustrated by a system that matches clothing. A mediocre shirt could match good with a mediocre skirt or pants, and form an outstanding collection. This thus increases the total value of the collection more than the average of the individual items, hence it has a positive co-occurrence interaction effect. Similarly, if the items would clash there would be a negative co-occurrence interaction effect. This analogy also holds for the domain of music.

Co-occurrence interaction effects are not necessarily restricted to pairs of tracks, but may also occur within larger sequences or playlist (comparable to the whole outfit in the clothing analogy). Therefore, this is also a factor that influences the holistic quality of the playlist and is thus not restricted to quality of the sequence order.

The literature on co-occurrence interaction effects in the domain of MRS is limited. Some authors of APG systems incorporated these effects by analyzing user-created playlists (Baccigalupo & Plaza, 2006; Maillet, Eck, Desjardins, & Lamere, 2009; Jannach & Lerche, 2015). Some studies focused on the effect of track characteristics (e.g. valence, tempo) on sequence co-occurrence effects. Liebman, Saar-Tsechansky, and Stone (2015) indicated that incorporating those track characteristics in an algorithm can improve its accuracy. Another user study showed that the valence and energy characteristics of tracks influence the co-occurrence effect while tempo does not (Hadash, Willemsen, Tintarev, Knees, & Tkalčič, n.d.). In other words, they showed that the flow of track transitions as perceived by the users is heavily dependent on whether the valence attribute of the music changes.

Order-effects describe the effects that occur purely based on the ordering of tracks. Order-effects can be divided into relative ordering effects and placement position effects. An example of relative ordering effects is that a certain track A may transition smoothly into track B, while a transition from B to A does not. Placement position effects are about the placement of a track in a playlist. For instance, some tracks are great songs to start or end a playlist with while others are not. Both types of order-effects can co-occur and their presence is not mutually exclusive.

Hansen and Golbeck (2009) performed a user study where participants had to create playlists from several fixed set of songs and provide ratings for each song. Although no statistical tests were used, they found that the first position received a significantly higher rating than any other position, and that the ratings amongst the other positions show no differences. They suggest that in the order-effect the specific position of a song in a sequence is not very important aside from the first one. They also found, based on occurrence frequencies, that some songs are much more or less likely to be next to each other (co-occurrence) compared to the base rate.

It has also been suggested that the ending song is important (e.g., Masthoff, 2015) because listeners remember the end of the sequence most. This effect may be attributable to the well-known peak-end rule heuristic which states that the most intense (peak) and the last (end) moment-to-moment experiences contribute more to the overall experience compared to the average (e.g., Kahneman, Fredrickson, Schreiber, Donald, & Redelmeier, 1993; Wiechert, 2018). Wiechert (2018) compared a peak-end metric with an averaging metric and found that the peak-end metric explained a distinct part of the variance in playlist experience. However, the variance explained by the peak-end metric was significantly smaller than that explained by the averaging metric.

All by all the quality of the playlist, and thus also the overall experience of the user(s), is not simply determined by the individual relevance of the items in the playlist but also by the specific sequence order. Both co-occurence interaction effects and order-effects of the sequence may play a role in the overall quality of the playlist, but how precisely these effects contribute to the overall playlist quality remains unclear as results are ambiguous (i.e. Wiechert, 2018) or have limited external validity (i.e. Hadash et al., n.d.).

Algorithms

The final subtask in the APG problem entails using an algorithm to find the best possible playlist. One often used type of algorithms are similarity-based algorithms. This approach is based on the fact that playlists should be coherent (Logan, 2004; Knees, Pohle, Schedl, & Widmer, 2006; Fields, Rhodes, & Casey, 2008) and makes use of distance functions to calculate the closeness of two tracks. After the distance function is defined a strategy is employed to find the tracks that form the playlist. Often the tracks a user liked are used as seed songs corresponding to the typical content-based recommendation approach (Pampalk, Pohle, & Widmer, 2005; Gärtner, Kraft, & Schaaf, 2007).

Other approaches are collaborative filtering (which is solely based on community-provided ratings), frequent pattern mining, statistical models, case-based reasoning, discrete optimization, and hybrid algorithms. For an elaborate overview of these algorithms, see Bonnin and Jannach (2014).

Playlist evaluation

The last concept that involves 'finding a good sequence' in the last subtask of the APG problem involves measuring how good the generated playlist is. Music recommender systems have its roots in machine learning and information retrieval. In these fields it is common to work with quantitative evaluation metrics. These metrics are traditionally accuracy-like (e.g. mean absolute error, precision, recall), but more recently beyond-accuracy metrics are being used more often (e.g. novelty, serendipity) (Schedl et al., 2018). Usage of quantitative measures has several benefits, like reproducibility, but also has significant limitations. The most important goals of a generated playlist is to provide personalized guidance to the user and to provide a pleasurable experience by recommending playlists with quality. Answering the question whether the experience and guidance offered to the users is of good quality involves both objective and subjective user-centric evaluation methods. Knijnenburg, Willemsen, Gantner, Soncu, and Newell (2012) proposed a framework for user-centric evaluation of recommender systems in general which can be applied for evaluating music recommendation systems.

2.2 Sequential music recommendation in perspective

The automatic playlist generation problem as described in Section 2.1 illustrated the role of sequence order in recommending playlists. The sequence order contributes directly to the playlist quality given a pool of tracks, background knowledge, and target characteristics. The relevance of the sequence order to the playlist quality can be dependent on the target characteristics, but this is not necessarily so. During music exploration the sequence order of a playlist may not be particularly relevant for discovering new music because people are highly interactive during active listening. For more passive listening the sequence order may be more important and bad sequence orders can be perceived as disturbing. The sequence order contributes to the overall playlist quality in conjunction with the individual track relevance and may be dependent on the target characteristics of the recommendation system. It is therefore important that improvements to music recommendation systems based on sequence order effects are closely tied to the target characteristics set by the stakeholders of the system.

Chapter 3

Group Music Recommender Systems

While certain forms of listening behaviors are individual, music consumption is considered highly social on many occasions. For instance during concerts, performances, and parties music is simultaneously enjoyed by groups of people. Strictly speaking, a group is defined as “two or more individuals who are connected by and within social relationships” (Forsyth, 2014) where ‘social’ implies that the relationship among the members originates from the presence of other people and not merely from impersonal factors like proximity or origin. The meaningful social relationship distinguishes groups from people merely connected by impersonal factors.

While a group should always have some form of meaningful social relationships, group music recommendation systems can be targeted for people with no meaningful relations. Examples of such target groups are people in public places like shops or elevators. Because of this the term ’group’ is used loosely in this thesis to account for both groups with and without meaningful social relationships.

Four group types are used in the literature that make the distinction between groups with and without meaningful social relationships (Boratto & Carta, 2010): established, occasional, random, and automatically identified groups. Established groups are groups explicitly formed by the members, whereas occasional groups denote groups formed by a shared (momentary) goal. Random groups are people that are connected purely by proximity and automatically identified groups are groups (profiles) formed based on algorithms.

The targeted group type can be used for classification of group music recommendation systems. Other factors that can be used to classify group music recommendation systems are whether individual preferences are developed or known, whether recommended items are presented or experienced by the users, whether the group interacts passively or actively with the system, whether sequences or individual items are recommended, and the type of preference aggregation strategy that is used (Masthoff, 2015; Boratto & Carta, 2010).

3.1 Preference aggregation strategies

A group recommender is inevitably faced with the task of aggregating data from individual profiles in order to generate group recommendations. The two strategies that allow recommendation systems to form group recommendations are the aggregated predictions and aggregated models strategies. With the aggregated predictions strategy recommendations are produced for each of the individuals of the group after which the total pool of recommendations are aggregated and ranked. An aggregated predictions strategy is useful when individual predictions are still desired, for example when predictions should stay close to the preferences of individuals.

The aggregated models strategy takes a different approach by aggregating the user profiles into a group profile after which recommendations are generated based on the group profile. This approach is particularly useful when the group should have the opportunity to influence and adapt the preferences of the group. Additionally, this approach may be beneficial in cases with privacy concerns since individual preferences do not need to be preserved after a group profile is created. In this situation only short-term explicit preferences are included by the recommendation system (Felfernig, Atas, et al., 2018).

3.2 Existing group music recommenders

Although most research on recommender systems focuses on recommending to individual users, research on group recommendation systems is becoming increasingly more popular (e.g., Felfernig, Boratto, et al., 2018; Boratto & Carta, 2010; Masthoff, 2015). Examples of existing group music recommendation systems existed in the late nineties (i.e. McCarthy & Anagnost, 1998) and several newer systems were developed since then.

MusicFX (McCarthy & Anagnost, 1998) is a GMRS that is employed at a fitness center. The system keeps track of the present users using a login system and allows members to set their genre preferences using surveys. A selection of broadcasting stations is classified by genre. Based on the preferences of the people who are currently present in the fitness center a broadcasting station is selected by averaging the genre preferences. The authors performed surveys and quantitative analysis based on behavior and preference logs to indicate that users were content with the music generated by their system.

Another group music recommendation system is Flytrap (Crossen, Budzik, & Hammond, 2002). Flytrap is a dynamic group music recommender system that actively keeps track of the present users by radio frequency ID badges and broadcasts music to people in a shared environment. The system uses a genre-similarity metric based on genre tags and user listening behavior to determine track relevance to the users. A stochastic aggregation strategy is used to determine the next track based on the track-user relevance scores. Additionally a virtual user is added that steers the system towards genre-coherent playlists and avoids artist repetitions.

Jukola (O'Hara et al., 2004) is an example of a group music recommendation system with active group participation. This system was installed at a local café bar in Bristol. People at the bar could obtain a handheld client after registration to participate in the decision process for deciding the music being played at the café. Participation occurred in the form of a voting system and each handheld device could place a vote prior to the selection of the next track. In practise handheld devices were used by established groups, but the decisions were shared over random groups (i.e. all participating groups in the café bar). The system primarily showed that the decision process in itself can be an important part of the experience of the users.

Adaptive Radio (Chao, Balthrop, & Forrest, 2005) is a system that broadcasts music to people in a shared environment. It focuses on improving recommendations by keeping track of the negative preferences of the users. It develops the user profiles over time and allows for some active interaction by providing a dislike button. When a track was disliked, all tracks of the corresponding album were excluded from future predictions for groups that included the user that disliked it. Tracks that belonged to albums of which none of its tracks were disliked by the group members were all included in the candidate set with equal weights.

More recent group music recommendation systems are GroupFun (Popescu & Pu, 2012) and the system developed by Piliponyte, Ricci, and Koschwitz (2013). Both systems were primarily used in user studies. GroupFun first obtained user preferences through user surveys. A collective local music database was used such that participants could select and rate songs that they liked. Based on these selections a candidate list for each user was obtained. Occasional groups were formed during the studies during which the user preferences were aggregated to a group profile. Using these group preferences and varying aggregation strategies playlists were recommended and rated by the users.

Piliponyte et al. (2013) developed a system that used user satisfaction balancing. This system used a local song database and matrix factorization collaborative filtering to compute track relevance scores for each user. Then they used an averaging aggregation strategy to aggregate the user predictions into a ranked list which is used as a candidate set. User satisfaction levels were predicted for each user by keeping track of the playback history and the impact each track had on each user. Finally, they picked tracks from the candidate set in such a way that the predicted user satisfaction levels change as little as possible.

3.3 Analysis of existing group music recommenders

Using the classification factors described earlier the existing examples of group music recommendation systems are classified in Table 1. The comparison shows that both groups with and without meaningful social relationships are targeted equally often. In situations where group music recommendation systems are located in shared environments, the systems need to deal with the uncertainty of who is present. Several solutions to this problem exist such as a login system (e.g. MusicFX), tokens and tags (e.g. Flytrap), and probabilistic mechanics. The probabilistic mechanics can use predictors like the time of the day or use more sophisticated predictors (e.g. visual recognition, voice recognition) (Masthoff, 2015).

Table 1: Classification of existing group music recommendation systems.

system	group type	user preferences	items experienced	group interaction	recommends	aggregated
MusicFX	random	known + developed	yes	passive	per item	predictions
Flytrap	random	developed	yes	passive	per item	predictions
Jukola	random + established	developed	yes	active	per item	predictions
Adaptive Radio	established	developed	yes	active	per item	predictions
GroupFun	occasional	known	yes	passive	sequences	models
Piliponyte et al. (2013)	occasional	known	yes	passive	sequences	predictions

Note. References: MusicFX (McCarthy & Anagnost, 1998), Adaptive Radio (Chao et al., 2005), Flytrap (Crossen et al., 2002), Jukola (O'Hara et al., 2004), GroupFun (Popescu & Pu, 2012).

Most systems develop user preferences over time by making use of explicit or implicit user feedback. A challenge that is inherent to group recommendation systems is that user feedback may be influenced by group effects like emotional contagion and conformity. Therefore, more uncertainty exists in the validity of the obtained user feedback and care should be taken to take group effects into account in the interpretation of the feedback. For example, user feedback for Jukola is obtained in a social environment where a single feedback elicitation device is used by established groups and the music is shared by a bigger random group. Some users of Jukola said "We've been fighting over what to vote for." (O'Hara et al., 2004, p. 149). This shows that feedback from this context should be interpreted as group preferences of the established groups and not as individual preferences of the registrant of the mobile device because the registrant may be in disagreement with the group decision. The obtained feedback should thus not directly be used to build long-term user profiles of the registrants but rather as short-term session-based preferences of the established groups.

All systems included in the list of examples let the user experience the recommended items. This is a benefit inherent to the domain of music, because musical items are usually of short duration and can therefore directly be experienced by the users. The music itself was usually obtained by maintaining a local database. The music included in these databases were user contributed (e.g., Adaptive Radio, Flytrap, GroupFun), maintained by staff and collected using Internet technologies (e.g., Jukola), or maintained by the researchers (e.g., Piliponyte et al., 2013).

MusicFX used broadcasting stations as the source for their music.

Some of the shown examples actively made use of group interactions. Jukola used a voting system to determine the next track and Adaptive Radio allowed group members to exclude albums of tracks they disliked. An important aspect of group recommendation systems is how to explain to the users how the decisions are made. Users of Jukola knew that the system used a voting system which probably made clear how the decisions were made. Adaptive Radio was less transparent to its users. Users could click a button to skip a track, but no information was given about what effect this action had. There are several benefits to group music recommenders that are transparent to the user. Users may more easily accept tracks that they dislike if it is shown that other users really like that particular track. Especially if the user also sees that the system also occasionally chooses tracks based on their own preferences. However, sometimes a person might not want others to know that a track was chosen based on their profile. Think of a scenario where an individual might be embarrassed by a particular track being associated to him or her. This leads to a conflict between the transparency and privacy of the system (Masthoff, 2015). An elaborate discussion on transparent group recommenders and the transparency and privacy conflict is outside the scope of this thesis (for more on explainable RS refer to Tintarev & Masthoff, 2015).

Most systems used some form of an aggregated predictions aggregation strategy. GroupFun is the only exception that used an aggregated models strategy instead. Although an aggregated models strategy was used the benefits associated to such a strategy were not exploited. For instance, there were no aggregated group preferences that the groups could interact with. The system was also not specifically designed to limit privacy concerns related to the collection of personal user profiles. Since the benefits of using group profiles were not exploited the system could just as well have been designed using an aggregated predictions aggregation strategy.

The group music recommender of Piliponyte et al. (2013) used a methodology different than the other examples. In the other examples users were always treated as equally important. The recommender of Piliponyte et al. (2013) dynamically balanced user importance based on predicted user satisfaction levels. Therefore, the relative importance of the users shifted during the operation of the system. Modelling of affective state is used more often in contemporary group recommendation research and can allegedly be used to increase the perceived fairness of the decision process (i.e. Masthoff, 2015; Masthoff & Gatt, 2006).

3.4 Modeling affective state

When recommending for a group, you cannot give everybody what they like all the time. Some tracks suit some users better than other users. To make sure that these sacrifices are fairly distributed among the users and ensure that no single user is completely left out, the affective state of the users can be modeled (see Masthoff, 2015; Masthoff & Gatt, 2006; Piliponyte et al., 2013).

Early attempts to model affective state used a summation over the user satisfaction with earlier items and the impact of a new item (Masthoff, 2004) to determine current user satisfaction levels. The impact was estimated using the individual item relevance scores (i.e. ratings) for the user. Several changes were used to improve the predictions: inclusion of low ratings, normalization, and a quadratic rather than linear estimation of item impact. The quadratic estimation makes the difference between ratings near the extremes more important than differences in the middle of the scale. While these changes were found to be an improvement, the affective state predicted using this method is only dependent on individual item relevance and not on the sequence order.

Later attempts improved the user satisfaction predictions by modeling several relevant psychological and social effects. Among these effects are the influence of mood on judgment, the difference between actual and retrospective experience, the influence of user expectations, the decay of emotion intensity over time, emotional contagion, and conformity Masthoff and Gatt (2006). The inclusion of these effects introduced a dependence of sequence order on the predicted user satisfaction levels. The user satisfaction models were evaluated using simulation studies and user studies in the learning domain, but not as part of an actual group recommendation system.

Piliponyte et al. (2013) implemented similar satisfaction models in an actual group music recommendation system. They used surveys prior to their study to build user profiles of the participants after which automatically identified groups were composed. Participants had to compose playlists for their group while having access to their groups' user profiles. Then they were presented with both a playlist submitted by their group member and a playlist generated by the recommendation system using one of several affective state models. The users had to evaluate both lists in a relative evaluation task while being unaware which list is user submitted and which is generated. Unfortunately, only descriptive statistics are reported and sufficient information for an interpretation of the results is not present.

The modeling of affective state is ultimately used to increase the perceived fairness of the system. It is meant to prevent that the recommendation system makes group members feel that others' preferences are not fairly represented in the recommendations. While several models were described no study measured whether such models actually affect how users perceive the fairness of the system. These studies do show that the importance weights of the group members during the operation of recommendation systems need not necessarily be equal. However, whether such models can actually increase the perceived fairness of the recommendation systems remains questionable.

Chapter 4

Recommending Sequences to Groups

While both sequential and group music recommendation systems are valuable topics in itself, we are ultimately interested in the combination. Practical group music recommendation systems are typically faced with the task of generating playlists and are thus inevitably dealing with sequential recommendations.

The combination of sequential and group recommendation poses additional challenges to those of group music recommendation and sequential music recommendation in itself. Firstly in shared environments where people may join and leave at arbitrary moments, the consumption of previous tracks is not shared among all members of the current group. This poses problems when the track history is used as part of the recommendation system. The track history is especially important for recommender systems that take sequence order into account. Secondly, some of the sequence order effects are purely individual. For example, placement position effects like the peak-end effect are different for each member and is determined by the time the member joins or leaves the environment. It is therefore relevant to examine related work on sequential group music recommendation systems.

4.1 Related work

While the research in the fields of sequential recommendation for single users and group recommendation are growing, there has been little effort in the area of sequential recommendation for groups (e.g. Felfernig, Boratto, et al., 2018; Ricci, Rokach, & Shapira, 2015).

Masthoff (2004) showed in the domain of news recommendation how previous items influence the impact of a new item on a group. The first study explored how the decision process during selection of a sequence of items for a group proceeds in humans. The study showed that some utilitarian aggregation strategies such as averaging and least misery are used and that people care about fairness and preventing misery.

The second study showed that mood and topical relatedness influenced ratings for future items, which Masthoff (2004) used to argue that sequential recommendation for groups can only be done dynamically such that the next item is decided shortly before it is presented and experienced.

Baccigalupo and Plaza (2006) implemented a web player that recommends and plays sequences of tracks to a group of people. The system aims to recommend sequences with variety, smooth track transitions, that are customized to the audience, and fairness in individual satisfaction. Following Masthoff's indication, the system also recommends tracks in real time based on the current audience. The goals of variety and smoothness are realized by disregarding tracks that were recently played and are not 'smooth' in relation to the previous track. Smoothness is determined using co-occurrence analysis based on existing playlists. The customization is based on user profiles and fairness is ensured by modeling user satisfaction. The satisfaction of the individual members are included in the aggregation algorithm by a satisfaction-weighted aggregation strategy, giving more weight to users that are less satisfied with the so far individually consumed playlist history. Baccigalupo and Plaza (2006) obtained some feedback from the users of the system. Some users indicated that some tracks were always followed by the same (other) track. Another problem was that the constraints that ensure smoothness over the playlist sequence interferes with the speed the system can adapt to a new audience. This feedback shows that the balancing of the components of such a system is crucial.

The system developed by (Piliponyte et al., 2013) also recommends music sequences to groups. The system used an aggregated predictions strategy and a matrix factorization collaborative filtering algorithm to determine item-user relevance. The individual predictions are then aggregated using an averaging strategy and user-weighted using a model for affective state.

4.2 Summary and conclusions of existing work

In Chapter 2 a summary of sequential music recommendation was given. The overview showed that the automatic playlist generation problem can be divided in a number of smaller problems. A track library should be obtained, background knowledge should be available or accumulated, target characteristics should be determined, and a sequence that fulfills the target characteristics as good as possible should be generated. The generation of the sequence is done using a certain algorithm and the playlist quality is both determined by individual item relevance and sequence order. The impact of sequence order on playlist quality can be dependent on the target characteristics, and improvements to recommendation systems based on the sequence order should therefore be tied to the target characteristics.

Chapter 3 addressed group music recommendation systems and contained an overview and discussion of previous work. Several literature examples were compared with each other. Most existing group music recommendation systems used or could have used an aggregated predictions aggregation strategy. A benefit of an aggregated predictions strategy is that such systems can build on existing recommendation algorithms used for individuals. Therefore, this strategy is more flexible and can more easily be used in a modular recommendation framework.

All group music recommendation system examples used some method to determine the relevance of the items to the users. MusicFX (McCarthy & Anagnost, 1998) classified their broadcasting stations by genre. Using known genre ratings per user the item (i.e. station) relevance per user was determined. Adaptive Radio (Chao et al., 2005) used a classification system to exclude albums of tracks that people skipped. Tracks of albums of which no tracks were skipped by a user were classified as relevant to the user, and all other tracks were marked as irrelevant. Flytrap (Crossen et al., 2002) used a tag-based genre similarity metric and user listening histories to determine user-item relevance scores. GroupFun (Popescu & Pu, 2012) only included items with user ratings and these ratings were used as the item-user relevance. Piliponyte et al. (2013) used matrix factorization collaborative filtering to determine the relevance of the items to the users.

Another method that all group music recommendation system examples used was aggregation. MusicFX (McCarthy & Anagnost, 1998) and Piliponyte et al. (2013) used an averaging aggregation function. Adaptive Radio (Chao et al., 2005) combined all item-user relevance scores to create a combined ranked list of items. For simplicity such a strategy will be termed a 'highest' aggregation strategy in this thesis. Flytrap (Crossen et al., 2002) used a stochastic aggregation strategy by using the item-user relevance score such that higher scores lead to higher probabilities for selecting the item. GroupFun (Popescu & Pu, 2012) used various aggregation strategies depending on the study condition.

Some existing group music recommendation systems used a method for steering the playlist towards target characteristics. Flytrap (Crossen et al., 2002) steered the playlist such that there were no drastic changes in genre. It also prevented tracks from the same artist to be played consecutively. Flytrap targeted a varied playlist with a coherent genre.

The system of Piliponyte et al. (2013) used unequal user importance to dynamically balance predicted user satisfaction levels. It is an implementation of affective state modelling (i.e. Masthoff

& Gatt, 2006) that aims to increase the perceived fairness of the recommendation system.

The existing work on sequential group music recommendation show that recommendations should dynamically be predicted on a per item basis. The affective state of the users are dependent on time and on the playback history. Additionally, groups can change dynamically over time. People can leave or join the group and group recommendation systems should be able to adapt to this.

In summary, an aggregated predictions aggregation strategy is preferred in most situations. Group music recommendation systems use an algorithm to determine item-user relevance scores. An aggregation strategy is used to combine the individual predictions. Target characteristics can be incorporated to steer the recommendations towards user or global preferences. These target characteristics should be dependable on the sequence order. Affective state modeling using dynamic balancing of user importance weights can be used for making the recommendations more fair. Such affective state modeling requires recommendations to be predicted dynamically and per item.

4.3 Framework for sequential group music recommendation

Based on the conclusions from the analysis of sequential and group music recommendation systems a modular framework for sequential group music recommendation systems is presented, see Figure 1. The framework contains a number of connected modules that each have their own role in the system. The framework is modular in the sense that varying implementations of the modules can be built. Module implementations can be changed in the framework without affecting the rest of the system. The rationale for choosing a modular framework is the increased external validity and flexibility of such systems. Additionally, the framework follows an aggregated predictions aggregation strategy which lends itself well for modularity. The rationale for choosing this aggregation strategy follows from that most current group music recommenders use this strategy and its relative simplicity.

4.3.1 User module

The user module is a source module that has no dependencies. In other words, the module provides information to the rest of the system, but the rest of the system has no influence on the user module. The purpose of the module is to keep track of the users of the system. It should know which users are active in the system. This can be similar to the currently present users but need not necessarily be so. For instance, people should be able to sign out from the system even if they remain present.

The secondary purpose of the module is to provide relevant information about the users to the system. This data is typically in the form of user profiles. Additionally, information about relationships between users can also be made available in this module. This can be relevant to other modules in the framework, but it depends on the type of implementation that is used in these modules.

While user profile information is necessary for some systems, this need not necessarily be the case in every implementation. The framework is meant to be a general framework that explains both simplistic and sophisticated sequential group music recommenders. In simplistic recommenders the user module may only keep track of the active users in the system and use minimalist implementations of the other modules such that user profiles and social data is not required.

4.3.2 Recommendation strategy module

The recommendation strategy module is tasked with determining the item-user relevance. For each active user it should generate a list of items with relevance scores. This is a basic functionality found in most recommendation systems and all group recommendation systems described

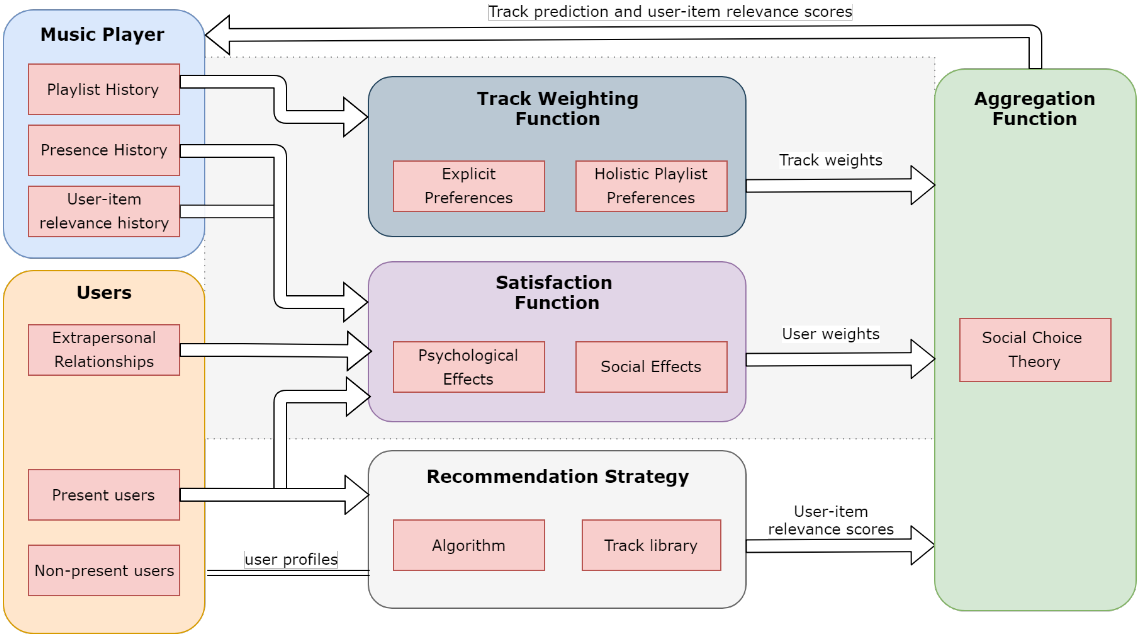


Figure 1: Framework for Sequential Group Music Recommendation. This model follows an aggregated predictions aggregation strategy. The light grey area shows the parts of the framework that deviate from the classical approach to group recommendation.

in Chapter 3 used some method to determine item-user relevance. It is similar to the recommendation algorithm used in recommendation systems aimed at individuals. Example strategies are collaborative filtering and content-based recommendation, but the module is not restricted to using such common algorithms. For instance, if user profiles consist primarily of genre ratings then tracks could be ranked based on their genres and the user-genre ratings.

The module is dependent on the user module, because it needs to know for which users it should generate predictions. If the implementation of the recommendation strategy module uses some form of user profiles, then it is also dependent on the user module for its user profiles. The recommendation strategy module should also have access to the track library and the track metadata.

The output of the recommendation strategy module is a matrix of item-user relevance scores. For sophisticated systems it is advised to have normalized relevance scores to make it more easy to balance the modules of the system.

4.3.3 Music player module

The music player module keeps track of the playback history of the system. It records which tracks were played when and for whom. It also records the user-item relevance scores that were used at the moment the decision for playing this track was made. It is therefore dependent on the output of the aggregation function module which makes the final decisions.

4.3.4 Track weighting function module

The track weighting module is used to steer the playlist towards target characteristics. These target characteristics can be explicit or implicit user preferences or global system preferences. Examples include genre coherence and preventing artist repetition (i.e. as in Crossen et al., 2002), preventing heavy fluctuations in track attributes (e.g., valence or danceability), and session-based purpose (explicitly asked or implicitly predicted).

The module depends on the music player module because it needs to know the playback history. Based on the history and target characteristics sequence order effects can be modeled in this module. This closely follows the conclusion from Chapter 2 on sequential recommenders that sequence order effects should be tied to target characteristics. The output of the module is a weight for each candidate track. To illustrate how track weighting modules can be implemented imagine the following scenarios.

Suppose that a group music recommendation system is used during a party at which people are dancing. The purpose of this system is to generate a playlist personalized to the group with danceable music. Suppose that each track has a danceability score as metadata that is generated using some audio analysis technique. Because the target characteristics require the system to recommend danceable music, the track weighting module can weight each track based on its danceability score. This technique does not depend on the playback history and is independent on the sequence order.

Imagine another scenario. Research has shown that people perceive the flow of a sequence of tracks to be better when there are no large fluctuations in the valence of tracks (i.e. Hadash et al., n.d.). Therefore, the developers of the system want to prevent track sequences with large fluctuations in valence. Suppose that each track has a valence score as metadata based on user tags. The module could look at the playback history to determine the valence of the current track. The candidate tracks are then weighted based on the absolute difference between the current valence and the candidate track's valence such that candidates with similar valence receive higher weights. Because this technique uses the playback history to determine the current valence it introduces an effect that is dependent on sequence order.

Not all group recommendation systems steer the playlist towards target characteristics. Simplistic implementations can use a track weighting function that simply returns homogeneous track weights. A slightly better but yet simplistic implementation can assign low or zero weights to tracks that have already been played (i.e. prevent track repetition). This can also be extended to prevent artist repetition by applying a low weight to tracks of artists of which tracks have already been played. Another extension might use an exponential function to gradually allow these tracks and artists to be repeated over time.

4.3.5 Satisfaction function module

The satisfaction function module is used to weight the importance of the active users. The most straightforward reason for introducing user weights is to increase the perceived fairness of the system by modeling affective state. The satisfaction function depends on the user module for the active users, user profiles, and social information. Additionally it depends on the music player module for the user presence history and item-user relevance scores used for deciding this history.

The satisfaction function can be used to introduce user weight balancing based on affective state modeling as in Masthoff and Gatt (2006); Piliponyte et al. (2013). Based on the item-user relevance scores of the history the individual satisfaction levels can be predicted and used to create user importance weights. This gives users with underrepresented musical interests a momentary increase in relative importance in deciding the next track.

Similar to the other modules, the satisfaction function does not need to be sophisticated for simpler recommendation systems. Systems that do not model affective state can simply use equal user weights.

4.3.6 Aggregation function module

The aggregation function module combines the recommendations for individuals produced by the recommendation strategy module into a ranked list of group predictions. The prediction with the highest score is then decided for the next track. The aggregation module depends on the item-relevance matrix produced by the recommendation strategy module, the track weights produced by the track weighting function module, and the user weights produced by the satisfaction function module.

The most straightforward implementation multiplies the track weights and user weights with the item-user relevance scores prior to aggregation (i.e. linear weighting). Then, an aggregation strategy based on social choice theory can be used to decide the next track. Examples of such strategies are averaging, 'highest' (as in Chao et al., 2005), stochastic (as in Crossen et al., 2002), and least misery. The module both returns the decided track and the corresponding unweighted item-user relevance scores. These scores are stored by the music player module because they are required by some of the other modules.

4.4 Rationale for the framework

Without the track weighting function and satisfaction function modules the framework is similar to the classical aggregated predictions aggregation strategy structure (Felfernig, Atas, et al., 2018). A drawback of this classical structure is that such systems decide the next track without taking previous decisions into account. For each track it recommends as if there is no playback history for the current session. This can result in scenario's where tracks are repeated consecutively, where users feel left out, or where the music in the playlist has no coherence.

The introduction of the track weighting function module is aimed to implement target characteristics and sequence order effects in group recommenders. Target characteristics can be set by the developers of the system based on stakeholder analysis or predicted during each session using either implicit or explicit user feedback. The module can also use a combination of persistent and session-based preferences.

The satisfaction function module is introduced to give affective state modeling a place in group music recommendation systems. The module introduces user weights to the system which can either be fair, persistently unfair, or dynamic.

The modular structure of the framework allows for varying implementations of modules. This is useful when variations in algorithms are compared to each other. For instance, affective state modeling can be toggled on and off by implementing two versions of the satisfaction function. The first version would then use affective state modeling and the second version would apply equal weights to all users. Then, while keeping the other modules and thus the rest of the system the same the impact of affective state modeling on user experience can be studied. This technique can be used for all modules to evaluate the effect of any particular algorithm or methodology.

Table 2 illustrates how the methodologies of existing group recommendation systems from the literature fit the framework. Each existing group music recommendation system described in Chapter 3 is classified in terms of the framework modules. This illustrates how the existing systems could fit the modular structure of the framework and shows the practical applicability of the framework.

Table 2: Illustration of how existing group music recommendation systems could fit within the framework.

system	user profiles	recommendation strategy	track weighting function	satisfaction function	aggregation function
MusicFX	genre ratings	broadcasting stations classified by genre and compared with genre ratings	equal weights	equal weights	averaging
Flytrap	user listening histories	genre-similarity metric based on tags	genre coherence and artist repetition prevention	equal weights	stochastic
Adaptive Radio	skip history	binary relevance based on skip history, similarity based on albums	equal weights	equal weights	highest
GroupFun	track ratings	uses ratings for determining relevance	equal weights	equal weights	various
Piliponyte et al. (2013)	track ratings	matrix factorization collaborative filtering	equal weights	various, including affective state modeling	averaging

Note. References: MusicFX (McCarthy & Anagnost, 1998), Adaptive Radio (Chao et al., 2005), Flytrap (Crossen et al., 2002), GroupFun (Popescu & Pu, 2012). Jukola (O'Hara et al., 2004) is not included because the details are not described in the publication.

4.5 Numerical example of the framework

The relevance of the introduction of the track weighting and satisfaction function modules to the classical approach to an aggregated predictions aggregation strategy is best illustrated using an example. Suppose that there is an existing recommendation strategy that is able to provide item-user relevance scores. Four users are active in the system and the recommendation strategy has computed the item-user relevance scores for these users and the candidate tracks, see Table 3. The aggregation function used by the system uses an averaging strategy which decides T_1 to be picked as the first track. The aggregation strategy picked a track that Erik and Tim like, but that does not score very well for Anne and Silvie. Nearly before the first track finishes playing the next track has to be decided. If no adaptations are made to the system after deciding the first track, the system will simply recommend the same track again.

One thing that can be done to prevent this is to simply remove a track from the track library that was previously played, but this is not a very good strategy considering that people actually like hearing tracks that they know. Another approach that fits the structure of the framework is to apply a weight of zero to the item. In Table 4 a weight of zero is applied to T_1 after this track was recommended to the group to prevent repetition. The table illustrates how this facilitates the aggregation algorithm for recommending another item. In more sophisticated track weighting functions the weight of T_1 can gradually increase over time to allow for repetition. The next track that is recommended in our example is now T_2.

This second track pleases Erik and Tim again and is disliked by Silvie and Anne. Since Erik and Tim were pleased with the previous track as well it would have been more fair if a track that Silvie and Anne liked was picked. Track T_3 would be a better option since it scores just slightly below T_2 but pleases both Silvie and Anne. This can be accomplished by a satisfaction function that models satisfaction levels of the users. Suppose such a satisfaction is simplistically predicted by taking the inverse of the previous track's score (normalized to a range between zero and one) for that user. The introduction of this satisfaction function makes T_3 win over T_2 as displayed in Table 5.

Table 3: Scores for Anne, Erik, Silvie, and Tim used for picking the first track. No weighting is applied.

User	T_1	T_2	T_3	T_4	T_5
Anne	4	2	8	3	1
Erik	7	8	0	1	9
Silvie	4	5	7	9	2
Tim	9	6	4	2	8
Average	7.25 WIN	5.25	5	3.75	5

Note. T = Track, WIN indicates the track in the corresponding column is chosen by the aggregation function.

Table 4: Scores for Anne, Erik, Silvie, and Tim for the second track after track weighting is applied.

User	T_1	T_2	T_3	T_4	T_5
Anne	4	2	8	3	1
Erik	7	8	0	1	9
Silvie	4	5	7	9	2
Tim	9	6	4	2	8
Track weights	0	1	1	1	1
Average	0	5.25 WIN	5	3.75	5

Note. T = Track, WIN indicates the track in the corresponding column is chosen by the aggregation function.

Table 5: Scores for Anne, Erik, Silvie, and Tim for the second track after track and user weighting.

User	User Weights	T_1	T_2	T_3	T_4	T_5
Anne	0.6	2.4	1.2	4.8	1.8	0.6
Erik	0.3	2.1	2.4	0	0.3	2.7
Silvie	0.6	2.4	3.0	4.2	5.4	1.2
Tim	0.1	0.9	0.6	0.4	0.2	0.8
Track weights		0	1	1	1	1
Average		0	7.2	9.4 WIN	7.7	5.3

Note. T = Track, WIN indicates the track in the corresponding column is chosen by the aggregation function.

This simplistic example illustrates how track weighting can be used to steer the playlist towards target characteristics (e.g. prevent repetition) and how user weighting can steer the playlist towards fairness and preventing misery. By choosing more elaborate track weighting and satisfaction functions more sophisticated kinds of target characteristics and fairness balancing can be implemented. Examples target characteristics that can be implemented range from simple repetition prevention to genre coherence and session-based playlist purpose characteristics.

4.6 Evaluation of the framework

The main contribution of the framework is the introduction of the track weighting function and satisfaction function modules to the classical aggregated predictions approach towards group recommendation systems. Although the introduction of these modules is based on examples in the literature, the effectiveness of these modules in sequential group music recommenders is not yet fully understood. Especially the usefulness of the satisfaction function module remains uncertain. While the satisfaction function is aimed to enhance the perceived fairness of group recommendation systems it has not been shown that user satisfaction modeling can actually achieve this. Masthoff and Gatt (2006) evaluated their satisfaction functions in the domain of learning and not in the domain of group recommender systems. Baccigalupo (2009) evaluated his GMRS, which included both track and user weighting by only describing user experiences and quantitative metrics. Also, some studies evaluated specific parts of the satisfaction function like the social factors. Quijano-Sanchez, Recio-Garcia, Diaz-Agudo, and Jimenez-Diaz (2013) evaluated group recommenders with social factors in the domain of movies.

To evaluate the usefulness of the satisfaction function module an existing implementation of the framework is required. Therefore, an implementation was developed with numerous module implementations to show the practical applicability of the framework. Using this implementation the usefulness of the satisfaction function module for increasing the perceived fairness of group music recommendation systems is explored in a user study. Because affective state modeling depends on the accuracy of the recommendation strategy module, the effectiveness of the implemented recommendation strategies is evaluated prior to the study such that the results are not confounded by inaccurate recommendation strategies. We propose the following hypothesis:

H1 *A satisfaction function that uses affective state modeling increases the perceived fairness of group music recommendation systems compared to a function that uses fair user weighting.*

Chapter 5

A practical implementation of the framework

Based on the framework proposed in Chapter 4 a practical implementation of a sequential group music recommendation system is developed. For each module several varying implementations were developed. These implementations are often common methodologies used in the literature (e.g. averaging aggregation) or with analogies to methods in the literature.

Since the framework requires a track library and user profiles the implementation is built upon the commercial platform of Spotify. This allowed the framework to be used with millions of tracks and existing user profile data. This gives the framework richer user data compared to using preference elicitation in the form of surveys prior to each user study. There are two reasons for choosing to use the user profiles of Spotify. The first reason is that Spotify has a large user-base in the country where we will conduct our study. In the Netherlands Spotify had a monthly reach of 32% in 2017 and the trend is increasing over the years, with a total of 6.2 million users in 2018 (Statista, 2019). The second reason is that Spotify has an open API which offers user profile data that can be used in the framework.

Disadvantages of using Spotify is the limited usability of the profiles. Spotify provides the top tracks and artists for a given user, but does not provide any item-user relevance scores. These scores are essential if we want to use the Spotify's user profiles the framework. Therefore several recommendation strategies are developed that are able to generate item-user relevance scores based on ranked lists of top tracks and artists (i.e. the user profile data available through Spotify).

5.1 Recommendation strategies

The recommendation strategies have to generate item-user relevance scores based on the user profiles (i.e. available representation of user preferences) and track library and metadata. In this implementation the user profiles consist of ordered lists of top tracks and artists. These lists are available over short, medium, or long time frames. Additionally, an existing black-box content-based or hybrid recommendation strategy is available within Spotify's API that is able to recommend similar tracks based on a seed track.

Four methodologies are developed that generate item-user relevance scores based on the available user profile data. The first methodology exploits the ordered nature of the top tracks and uses the seed recommender to find tracks related to the top tracks of a user. The second methodology extends the first methodology for the top artists. The third methodology uses features from Spotify's audio analysis to find similar tracks using a Gaussian mixture clustering model. The fourth methodology uses an operationalization of genre distance to obtain a list of top genres for a user and uses this to recommend tracks with genres the user likes. Finally, the methodologies can be combined for a hybrid approach towards generating the user-track ratings.

These methodologies are chosen because they produce a diverse set of predictions. The track and artist distance methodologies generate user-track ratings for a small subset of the total pool of tracks relatively close to the users' taste. While this certainly increases the accuracy of the recommendations for the individual users, it not very helpful when aggregating over the group, because it is very likely that the recommendations for each user will be disjoint sets.

The audio feature methodology makes use of a limited number of clusters and each user can have scores for multiple clusters. Therefore, the proportion of the total pool of tracks for which a nonzero score is available is much larger compared to the track and artist distance methodologies. This increases the likelihood that tracks can be found for which multiple users have nonzero scores, which improves the system's capacity to aggregate individual predictions to group predictions.

The genre distance methodology is also able to generate relevance scores for a large proportion of the total pool of tracks, although the proportion is a bit smaller compared to the audio feature methodology. The genre distance methodology will have user-track relevance scores for all tracks of which the artists contain genres that are in the set of top genres and their first order connections to other genres.

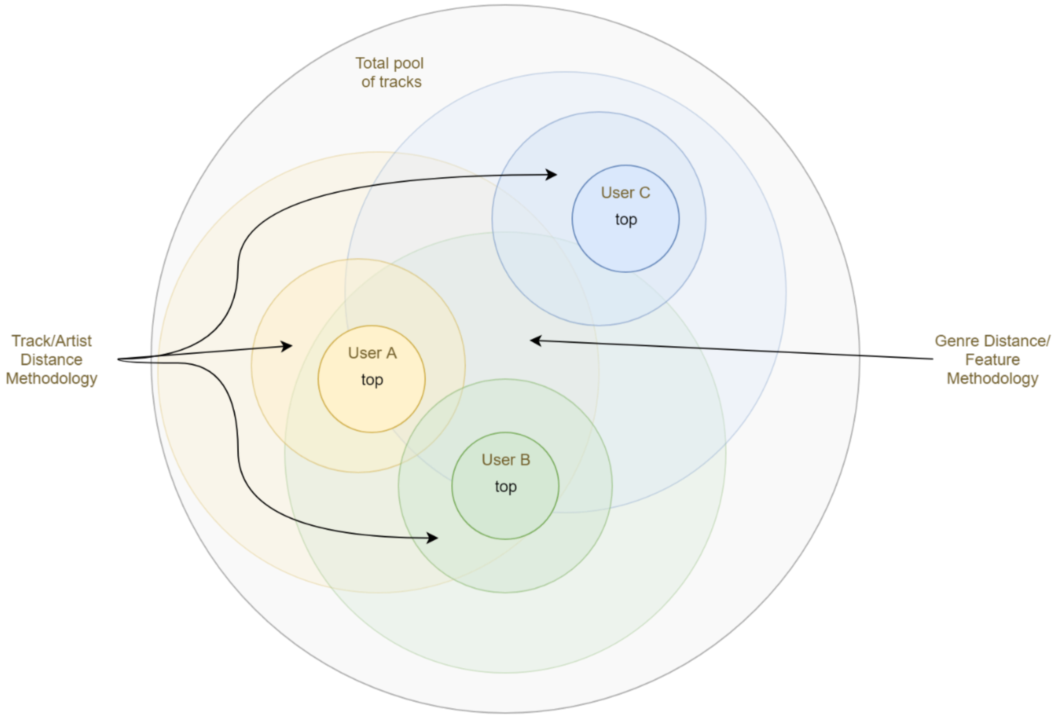


Figure 2: Visualization of the sets of recommendations that each methodology can generate. The space represents the concept of musical taste.

In summary, the track and artist methodologies find recommendations close to the user's top tracks and artists. The genre distance methodology finds recommendations for whole clusters of genres resulting in recommendations for a substantial subset of the total pool of tracks. The audio feature methodology typically generates user-track ratings for a very large proportion of the total pool of tracks. Figure 2 visualizes the difference in spread (i.e. relatedness to personal taste) between the various methodologies.

5.1.1 The disjoint set problem

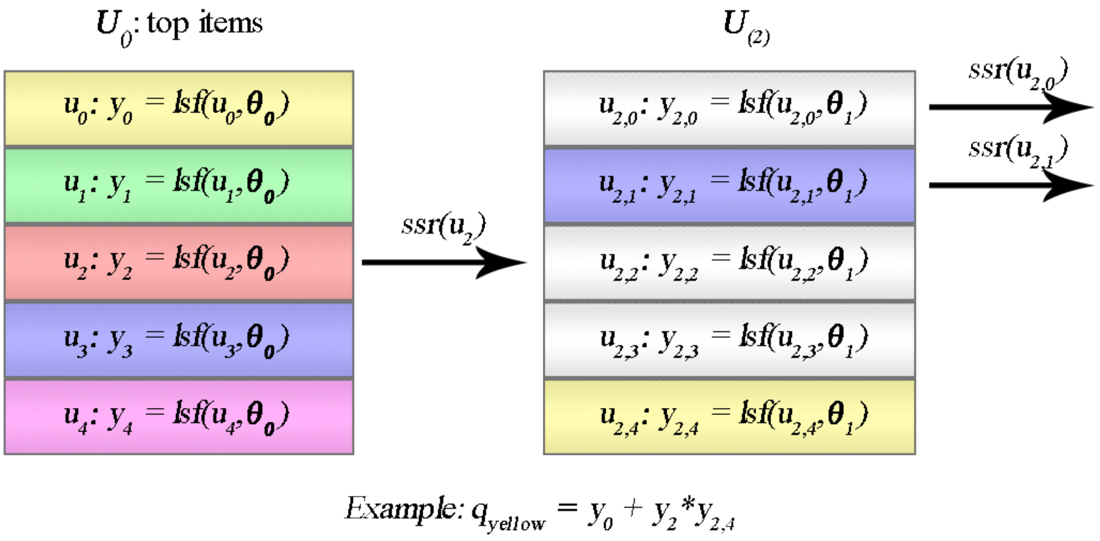
Usage of high-spread methodologies like the genre distance methodology and the audio feature methodology is essential when aggregation functions other than fairness are used in a group recommendation context because these strategies requires that predictions for individuals exist for the same set of items. While the audio feature and genre methodologies are able the generate predictions for more tracks this presumably comes at the cost of a less accurate prediction compared to the low-spread methodologies. The genre and audio feature methodologies are therefore not intended to be used separately but rather in conjunction with the low-spread methodologies in order to increase the spread of the recommendations.

In other words, the high-spread methodologies are intended to solve the problem of aggregating predictions for individuals to group predictions when the individual predictions are disjoint sets. This problem will be referred to as 'the disjoint set problem' in this thesis. While the high-spread methodologies solve this problem they are also hypothesized to decrease recommendation accuracy. The evaluation of these methodologies is therefore a balancing question: what is the cost in terms of accuracy loss for using the high-spread methodologies in conjunction with the low-spread methodologies. Whether the genre distance methodology or the audio feature methodology is the better choice is yet to be determined.

H2 *Using the genre or audio feature methodology in conjunction with the track and artist distance methodologies decreases the accuracy of the predicted user-item relevance.*

5.1.2 Track distance methodology

The basic idea of this methodology is to exploit the ordered nature of the top-tracks of a user profile. We assign a linear score to each top track based on its position in the list. Then we use Spotify's seed recommender to obtain lists of tracks related to the users' top tracks and assign scores to them in a similar fashion. We multiply the score of the top-track with the scores assigned to the related list. Then, recursively, we repeat this process with the tracks in the related lists returned by the seed recommender up to a depth D because tracks can occur multiple times in these lists. This is illustrated in Figure 3.



U_0: top items
u_0: $y_0 = lsf(u_0, \theta_0)$
u_1: $y_1 = lsf(u_1, \theta_0)$
u_2: $y_2 = lsf(u_2, \theta_0)$
u_3: $y_3 = lsf(u_3, \theta_0)$
u_4: $y_4 = lsf(u_4, \theta_0)$
$ssr(u_2)$
$U_{(2)}$
$u_{2,0}$: $y_{2,0} = lsf(u_{2,0}, \theta_1)$
$u_{2,1}$: $y_{2,1} = lsf(u_{2,1}, \theta_1)$
$u_{2,2}$: $y_{2,2} = lsf(u_{2,2}, \theta_1)$
$u_{2,3}$: $y_{2,3} = lsf(u_{2,3}, \theta_1)$
$u_{2,4}$: $y_{2,4} = lsf(u_{2,4}, \theta_1)$
$ssr(u_{2,0})$
$ssr(u_{2,1})$
Example: $q_{yellow} = y_0 + y_2 * y_{2,4}$

Figure 3: Visual explanation of the Track Distance Methodology. Colors represent unique tracks, indicating that a track can have multiple occurrences across all lists.

Let us first start with a few definitions:

$$U_k = (u_{k,i})_{i=1}^{N_k} \tag{1}$$

$$U_{(k,i)} = ssr(u_{k,i}) \tag{2}$$

Where u indicates an item (in this methodology a track), U indicates an item sequence, and k is the sequence identifier. For example, $k = ()$ indicates the top list, $k = (1)$ indicates the list from seed item u_1, and $k = (1, 3)$ indicates the list from seed $u_{1,3}$. N_k is the number of items in sequence k, and ssr is Spotify's seed recommendation system.

Now, let us define our scoring functions:

$$lsf(u_{k,i}, \boldsymbol{\theta}_{|k|}) = \frac{N - i}{N}(s_{max} - s_{min}) + s_{min} \tag{3}$$

$$\boldsymbol{\theta} = \{N, s_{min}, s_{max}\} \tag{4}$$

Where lsf is our linear scoring function, $\boldsymbol{\theta}_{|k|}$ is the set of parameters at depth $|k|$. s_{min} and s_{max} are the scores assigned to the last and first item in the sequence respectively. Then, we define a score tree such that we can score each possible item position in the tree.

$$Y_{(k,i)} = (lsf(u_{k,i}, \boldsymbol{\theta}_{|k|}))_{i=1}^{N_k} \tag{5}$$

Finally, for each unique element present in the set of items up to depth D, the score Q is equal to:

$$U = \{\bigcup_k U_k, |k| \leq D\} \tag{6}$$

$$Q = \{\sum_{k,i} \prod_{q=k_1}^{k_{1:|k|}} y_{q,i} : v \in U, u_{k,i} = v\} \tag{7}$$

Where U is the total set of recommendations, v is an element from this set, $y_{(q,i)}$ is an element from the score tree, q is a subsequence of k. For example if $k = (1, 3, 5)$, then $q = (1)$, $(1, 3)$, and $(1, 3, 5)$ in the product. Q is then the set of scores generated using this methodology.

This concludes the track distance methodology. A relevance score is generated for a set of tracks closely related to a profile's top tracks based on Spotify's seed recommender.

5.1.3 Artist distance methodology

This methodology applies the track distance methodology to the list of top artists for the user after which the artist scores are used to assign track scores. All tracks that have an artist for which a score is available will have a nonzero score in this methodology. A track can have multiple artists, and we assume that a track for which scores for multiple artists are available is a better recommendation than a track for which only a single artist has a score given that the track has multiple artists and the scores are all equal. Using this assumption the artist scores of each track are added and divided by the number of artists. The final scores are thus:

$$Q_A = \{\frac{1}{|t|}\sum_{a}^{|t|} q'_A : t \in T, a \in t, a \in U_A\} \tag{8}$$

Where T is the total set of tracks, t is the set of artists belonging to a track, a is an artist, U_A is the set of artists for which a score is available using the algorithm from the previous section applied to the artists, q'_A is the artist score, and Q_A are the aggregated track scores based on the artists.

5.1.4 Audio feature methodology

The third methodology makes use of Spotify's audio analysis of tracks and is a form of content-based recommendation. Spotify has performed audio analysis on their tracks and derived a number of features. The features are danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. A more detailed description of their analysis and

Table 6: Example calculation of the Audio Feature Methodology.

	$p(T_i \in C_1)$	$p(T_i \in C_2)$	$p(T_i \in C_3)$	
Top tracks				
Track 1	.8	.1	.1	
Track 2	.6	.2	.2	
Track 3	.3	.3	.4	
Track 4	.4	.6	0	
Track 5	.7	.2	.1	
Total	2.8	1.4	.8	
Other tracks				*Score* $p(A_i\|C)$
Track 1	.1	.3	.6	.236
Track 2	.8	.2	0	.504
Track 3	.5	.3	.2	.396

Note. $p(T_i \in C_k)$ indicates the probability of track T_i (i.e. the rows) to belong to cluster C_k, $p(A_i|C)$ indicates the probability that the user likes track i (the row) given the probabilities of liking each cluster (top-part of the table). Track 2 receives the highest score because there is a high probability that it belongs to cluster 1 and that the user likes this cluster.

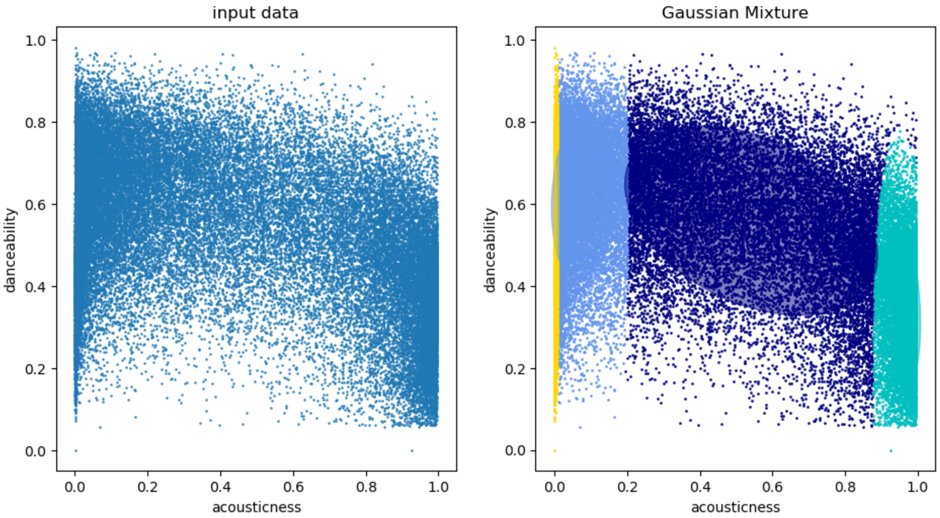


Figure 4: Visualization of two-dimensional Gaussian mixture clustering primarily used for illustration purposes. The colors indicate the clusters, where a track is labeled to a cluster when it has, according to the fitted model, the highest probability of belonging to the cluster. The number of components is user-specified.

the features is available in Jehan and Desroches (2004). The general idea behind this methodology is that tracks that have similar features to tracks that you like are better recommendations compared to tracks that you do not like. A good example is a user that primarily listens to classical music. One can imagine that tracks that are high in acousticness or instrumentalness are better recommendations compared to other tracks for this user.

First, we build a large database of tracks and their audio features ($n_{initial} \approx 500000$) by fetching them from Spotify. Then, we fit a Gaussian mixture clustering model on the data by means of the expectation-maximization (EM) algorithm, i.e. the model components are estimated based on the track metadata. This is done multiple times for varying component numbers and the fitted model with the lowest BIC criterion is selected and stored. This model should be recomputed when the number of tracks in the database has grown for a significant amount.

Then, we predict for each track in the database the probability of belonging to each cluster using the model. For each user account, we take its top tracks and their predicted cluster probabilities and aggregate this to cluster probabilities for the user account using Equation 9, see Appendix A for proof and assumptions.

$$p(user\ likes\ cluster\ i) = \frac{1}{N_{top}} \sum_{j=1}^{N_{top}} p(track\ j\ belongs\ to\ cluster\ i) \tag{9}$$

Then, track-user scores are generated by multiplying the user-cluster probabilities with the track-cluster probabilities, see Table 6 for an example.

5.1.5 Genre distance methodology

The fourth methodology is based on the concept of genres and the idea that users can like certain genres and dislike others. The goal is to generate user-track ratings based on the genres that the user likes. In Spotify, artists can have a number of genre classifications, while tracks do not. Therefore, the user's top genres are determined by looking at the user's top artists. This methodology is identical to the method used by Lamere (2012) to generate a genre similarity visualization, see Figure 5.

Genre scoring based on top artists

We apply the same method as in the Track Distance Methodology to benefit from the ordered nature of the top artists list. First, each artist receives a linear score based on its position in the list. Then, the genres associated to the artist copy the score of the artist and we count the total scores for each genre. In this way, we take the fact that a user likes certain artists more than other artists into account in the calculation. As a result, we have genre scores for the set of genres contained in the user's top artists list. In short,

$$\boldsymbol{G}_u = \{\{g : g \in a\} : a \in \boldsymbol{A_T}\}, \quad \boldsymbol{G}_u \subseteq \boldsymbol{G} \tag{10}$$

$$S'_{u,k} = \begin{cases} \sum\{\{lsf(a, \boldsymbol{\theta}_0) : g \in a, g \in k\} : a \in \boldsymbol{A_T}\} & k \in \boldsymbol{G}_u \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Where $\boldsymbol{G}_u$ is a subset of the total set of genres $\boldsymbol{G}$, $\boldsymbol{S}'_u$ are the genre-scores for user u as a column vector with cardinality $|G|$, g and k are genres, a is an artist, and $\boldsymbol{A_T}$ is the set of top artists. A limitation of this methodology is that we only have nonzero scores for a subset of genres, which makes it more difficult to bridge the gap between distinct user profiles. Therefore we introduce a method to extrapolate scores for the whole set of tracks based on this subset.

Genre similarity metric

We make use of a similarity metric between the genres by means of co-occurrence analysis (as in Pachet, Westermann, & Laigre, 2001). This analysis assumes that artists tend to produce music within a small cluster of genres with not a lot of diversity.

We implemented this by iterating over all known artists in our database (consisting of Spotify's data; $n \approx 80.000$ prior to the pilot study) and counting the co-occurrences of genres within artists in a genre-genre matrix. The resulting symmetric similarity matrix is normalized and stored as $\boldsymbol{D}$.

Genre score extrapolation

To generate genre scores for genres not included in the users' top tracks and artists we apply the similarity metric to the genre scores that we already have for the user.

$$\boldsymbol{S}_u = (\boldsymbol{D} + \boldsymbol{I})\boldsymbol{S}'_u\boldsymbol{1} \tag{12}$$

Where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{1}$ is a $|\boldsymbol{S}_u| \times 1$ sum vector.

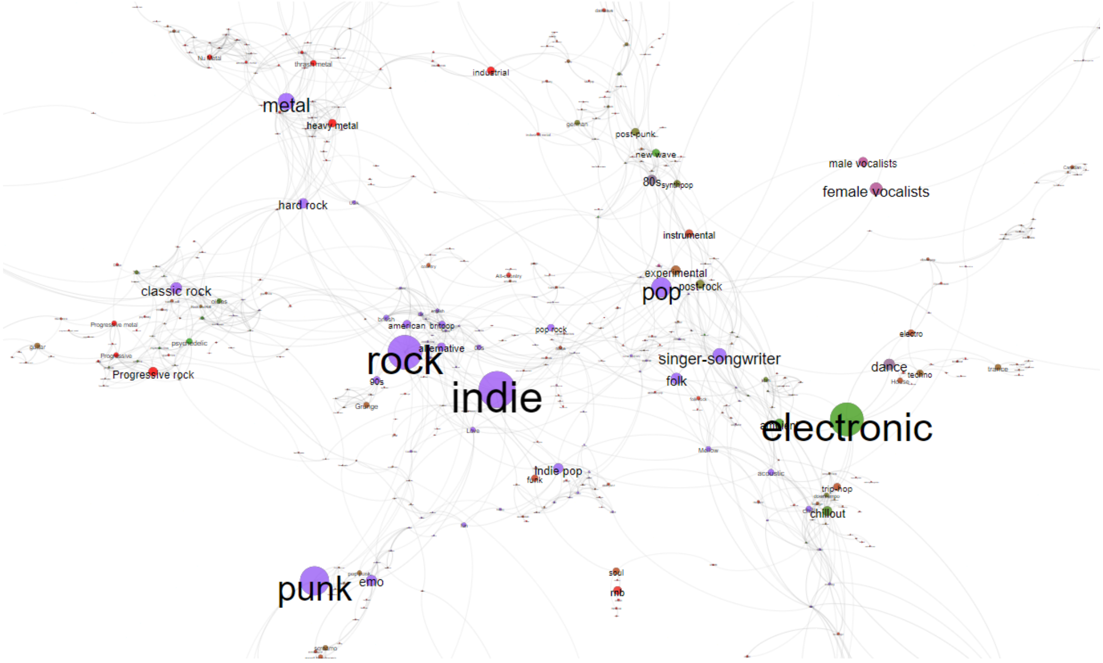


Figure 5: Visualization of the genre similarity co-occurrence analysis. Dot size indicates how often a genre occurs in general and connections are made where genres co-occur amongst artists. Copied with permission from Lamere (2012).

5.2 Aggregation functions

Implementations of the aggregation function module includes averaging, 'highest', and fairness strategies. All modules multiply the item and user weights with the item-user relevance matrix prior to aggregation. The averaging aggregation function computes the average item relevance scores over all active users after which the item with the highest average is recommended. The 'highest' aggregation module ranks the recommendations by maximum relevance score. The fairness module cycles between the active users and recommends the best prediction for the user whose turn it is.

5.3 Track weighting functions

The track weighting module implementations includes an equal weights module, a persistent no repeat module which prevents track repetition, an exponential repeat module that prevents track and artist repetition but reduces weight penalties over time, and a genre filter. The genre filter assigns a weight to tracks of particular genres and another weight to tracks that do not have those genre classifications.

5.4 Satisfaction functions

Three satisfaction functions are implemented: an ’equal’ function which uses equal weights for all users, a ’delta’ function that assigns more weight to a particular user, and an ’emotional decay’ function that uses affective state modeling. The ‘emotional decay’ satisfaction function used the following formula to model user satisfaction

$$impact(u, i) = (score(u, i) - 0.5)^2 \tag{13}$$

$$sat_{i+1}(u) = (1 + \epsilon)\ \delta\ sat_i(u) + (1 - \epsilon)\ impact(u, i), \quad \text{with } 0 \leq \delta \leq 1,\ 0 \leq \epsilon \leq 1 \tag{14}$$

Where i represents a recommended item, $score(i)$ is the algorithm score computed for user u for item i, $impact(i)$ represents the impact item i has on user u independent on the current emotional state of the user. δ models emotional decay such that lower values indicate that satisfaction decays more rapidly over time. Parameter ϵ models the extent to which the user’s current satisfaction influences the impact a new item has (Masthoff & Gatt, 2006). Then, the user satisfactions are converted to weighting scores dynamically after each new recommendation using an exponential conversion.

$$w(u) = e^{\frac{-sat(u)\ \lambda}{\max_{v \in u} sat(v)}} \tag{15}$$

Where λ is the decay factor, indicating to what extent changes between user satisfactions among the group members are differentiated in weight differences.

Chapter 6

Evaluation of the recommendation strategy module implementations

In Chapter 4 a modular framework for group music recommendation systems was introduced and a practical implementation of the framework was described in Chapter 5. Using this implementation the usefulness of the satisfaction function module in the framework can be evaluated. However, the satisfaction function module is dependent on the accuracy of the recommendation strategy. Therefore, a user study is conducted that aims to answer the following hypothesis:

H3 *The recommendation strategy methodologies generate item-user relevance scores such that a higher score indicates a higher user likeability.*

In group recommendations it is important to solve the disjoint set problem when aggregating predictions for individuals to group predictions. This study will additionally compare the genre distance methodology with the audio feature methodology in terms of how well these methodologies are able to solve the disjoint set problem without introducing a too large penalty to accuracy (Hypothesis 2).

6.1 Method

6.1.1 Participants

Participants in this study included 59 people, of which 54 were participants of the JF Schouten database of Eindhoven University of Technology. Participation occurred individually, not in groups. The sample consisted of 31 males and 28 females. The age of the participants ranged from 19 to 64 ($M = 25.6, SD = 8.8$). Participants were required to have a Spotify account (free or Premium) and to have used this account prior to taking part in the study.

6.1.2 Design

This study used a within-subjects experimental design in the form of an online survey. The study consisted of two levels. The first level compared playlists as a whole. Dependent variables on this level were perceived accuracy, perceived diversity, and perceived attractiveness. These variables were measured such that the recommendation strategies could be compared with each other in terms of attractiveness and accuracy. Diversity is measured as a control variable to ensure than any effects that we might find are not mediated by playlist diversity. The independent variable was the recommendation strategy algorithm used to generate the playlist (the condition). Three conditions were used: an equally weighted combination of the track and artist distance methodologies (base), an equally weighted combination of the track, artist, and genre methodologies (genre), and an equally weighted combination of the track, artist, and audio feature methodologies (gmm). The track and artist distance methodologies were used together because they are both low-spread methodologies and we were interested in comparing the low-spread with the high-spread methodologies. The high-spread methodologies were used in combination with the track and artist distance methodologies rather than separately because the high-spread methodologies were not intended to be used separately. Therefore, the combination of the high-spread methodologies with the track and artist distance methodologies would give results with more practical applicability.

The second level measured how well the predicted item-user relevance scores are able to predict user likeability. The dependent variables were likeability (i.e. user ratings) and perceived personalization scores. The independent variable was the user-item relevance score as generated by the algorithm. Each playlist contained lower rating predictions besides the top 3 best predictions such that the predictive accuracy of the item-user relevance score could be evaluated over a broader range. The condition order and recommendation item order were randomized.

6.1.3 Materials

We made use of a variety of scales for measuring the dependent variables in this study. Additionally, we measured music sophistication and demographic variables as control variables.

On the item level, one question was used for measuring likeability: "Rate how much you like the song". For measuring perceived personalization we used the following item: "Rate how well the song fits your personal music preferences". Both questions were answered on a 5-point with halves (i.e. 10 actual options) visual scale containing star and heart icons, respectively.

On the condition level, we used the playlist evaluation scale presented in Table 7 to measure perceived attractiveness and perceived diversity. The items were adapted from Willemsen, Graus, and Knijnenburg (2016) to fit the context of a playlist instead of a list of movies. Items with low factor loadings were removed to keep the questionnaire short.

Table 7: The items of the playlist evaluation scale. Adapted from Willemsen et al. (2016).

Concept	Item	Factor Loading	Specific Variance	Communality
Perceived attractiveness Alpha: .94	The playlist was attractive	1.08	0.14	1.18
	The playlist showed too many bad items	-1.15	0.17	1.34
	The playlist matched my preferences	1.11	0.15	1.24
Perceived diversity Alpha: .85	The playlist was varied	0.92	0.26	0.85
	The tracks differed a lot from each other on different aspects	0.87	0.17	0.76
	All the tracks were similar to each other	-0.87	0.26	0.76

Note. Negative factor loadings indicate a negative framing. Factor loadings are calculated using a principal component analysis without rotation. PCA was appropriate for this scale as shown by Kaiser-Meyer-Olkin's $MSA = 0.73$ and Bartlett's test of sphericity $\chi^2(15) = 867, p < .001$ (Williams et al., 1996).

Music sophistication was measured using the general scale of the Goldsmith Music Sophistication Index (MSI) (Müllensiefen, Gingras, Stewart, & Ji, 2013). Music sophistication is measured for a possible use as a control variable. The demographic variables that we measured contained gender, age, and the amount of time listening to Spotify. The items are presented in Table 8.

A custom platform was developed that is capable of taking the questionnaires, playing playlists, and that makes use of an implementation of the framework, see Figure 7. The server was hosted at Delft University of Technology and a secure connection was used to protect the data of the participants.

Table 8: Demographic variables scale.

Concept	Item	Options
gender	What is your gender?	Male, Female, I'd rather not say
age	What is your age?	
Spotify usage	I listen to Spotify for __ hours a week.	0, 1-3, 4-6, 7-10, 11-15, 16-20, more than 21

6.1.4 Procedure

Participants received a link to the online study per e-mail. After clicking the link, participants were directed to the start of the online study, and a consent form was presented. After explicitly accepting the consent form, participants could continue to the study.

A login screen (Figure 6) was presented which prompted the user for signing in with Spotify. After the user successfully signed in with his/her Spotify account, the user could continue to the next screen. In the background we validated whether the account was actually in use with Spotify. If the account was new or hardly used, a prompt was displayed indicating that this is the case. The participants were directed back to the login screen and requested to either choose a different account or return at a later time after having used the account with Spotify.

Then, consecutively the demographic and MSI scales were presented while in the background the recommendations for the first condition were calculated for the participant. These recommendations were calculated in the background because the high-spread methodologies had a significant computation time (i.e. several minutes) due to the vast amount of tracks for which predictions were to be computed ($n \approx 0.6$ million). After having filled in both scales, an information screen was presented. Participants were informed about having to listen to music samples and having to rate the individual tracks. Additionally, participants were informed about the possible need to wait a small amount of time for the playlist generation to finish.

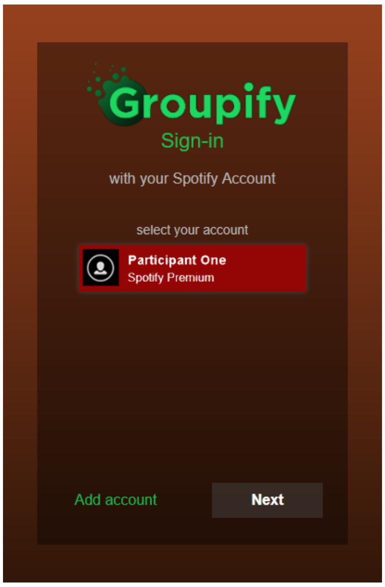
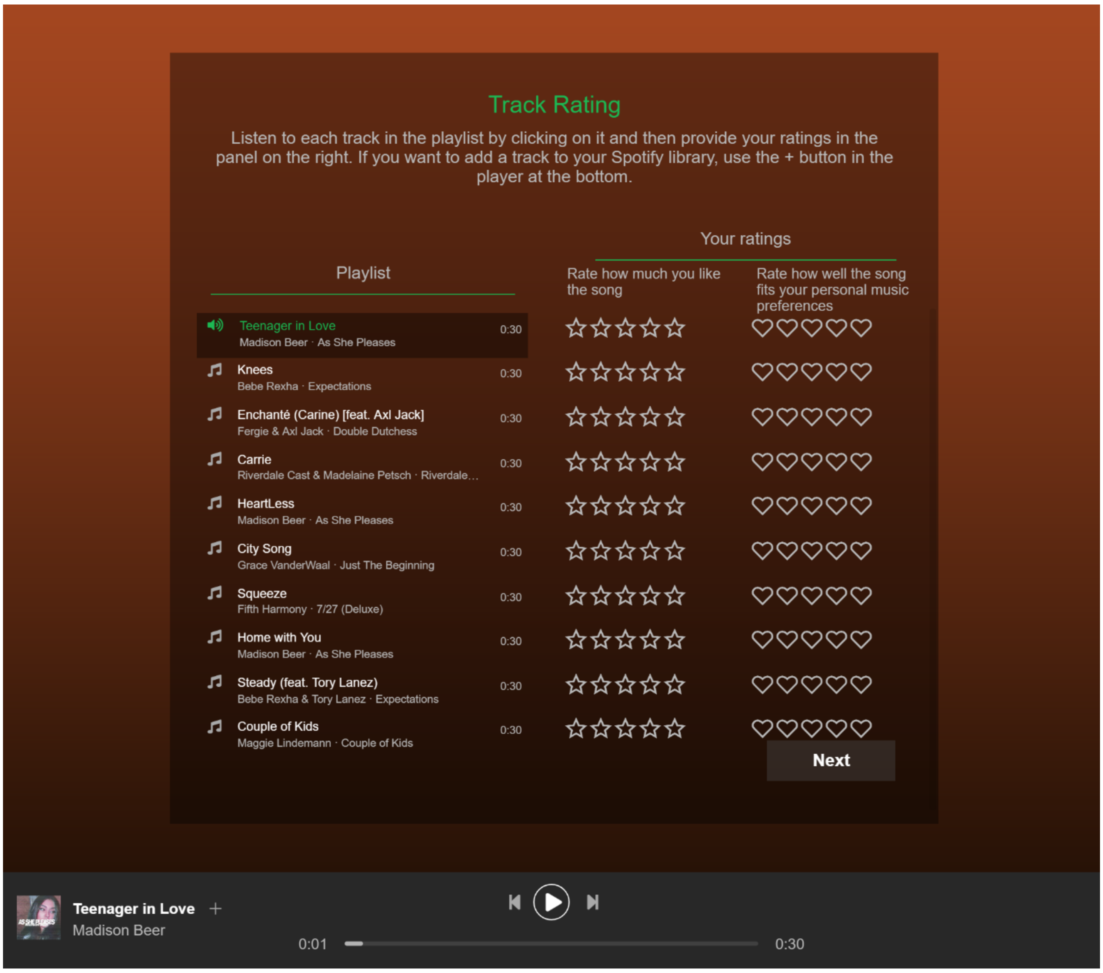


Figure 6: Screenshot of the login screen presented to participants in study 1. Clicking 'Add account' would direct users to Spotify, at which they were requested to grant Groupify permission for using their data for personalization purposes. After this permission is granted users are directed back to the login screen. Their user card then shows up and the 'next' button lights up, allowing them to continue with the study.

Then, the track rating screen was presented (Figure 7) for the first condition, followed by a the playlist evaluation scale (Table 7). During the rating of the tracks and filling in of the questionnaire, the recommendations for the next condition were computed. After having completed all three track rating and questionnaire phases, the participant was thanked for his time and effort. If a participant identifier was not present, i.e. the participant was not from the JF Schouten database, the participant was prompted for his/her e-mail in order to receive the possible reward.



Track Rating

Listen to each track in the playlist by clicking on it and then provide your ratings in the panel on the right. If you want to add a track to your Spotify library, use the + button in the player at the bottom.

Playlist		Your ratings Rate how much you like the song	Your ratings Rate how well the song fits your personal music preferences
Teenager in Love Madison Beer · As She Pleases	0:30	☆☆☆☆☆	♡♡♡♡♡
Knees Bebe Rexha · Expectations	0:30	☆☆☆☆☆	♡♡♡♡♡
Enchanté (Carine) [feat. Axl Jack] Fergie & Axl Jack · Double Dutchess	0:30	☆☆☆☆☆	♡♡♡♡♡
Carrie Riverdale Cast & Madelaine Petsch · Riverdale...	0:30	☆☆☆☆☆	♡♡♡♡♡
HeartLess Madison Beer · As She Pleases	0:30	☆☆☆☆☆	♡♡♡♡♡
City Song Grace VanderWaal · Just The Beginning	0:30	☆☆☆☆☆	♡♡♡♡♡
Squeeze Fifth Harmony · 7/27 (Deluxe)	0:30	☆☆☆☆☆	♡♡♡♡♡
Home with You Madison Beer · As She Pleases	0:30	☆☆☆☆☆	♡♡♡♡♡
Steady (feat. Tory Lanez) Bebe Rexha & Tory Lanez · Expectations	0:30	☆☆☆☆☆	♡♡♡♡♡
Couple of Kids Maggie Lindemann · Couple of Kids	0:30	☆☆☆☆☆	♡♡♡♡♡

Figure 7: Preview of the track rating screen as displayed to the participants during the study. The panel consists of a rating panel (user input) on the right, and an interactive playlist to the left. Interaction with the music occurs by clicking items in the playlist or by making use of the control buttons or seek bar at the bottom.

6.2 Results

The study took place between the 9th of January and 1st of February of 2019. Prior to the analysis, participants with incomplete submissions were removed from the data ($n = 55$). These participants mostly stopped with the experiment prior to the first track rating condition but after the MSI and demographic questionnaires. These were also predominantly participants from the first few days of the study. In these days a technical problem occurred which may have prevented some participants from continuing with the study. Figure 8 displays the distributions of hours participants listened to Spotify per week and their Music Sophistication scores.

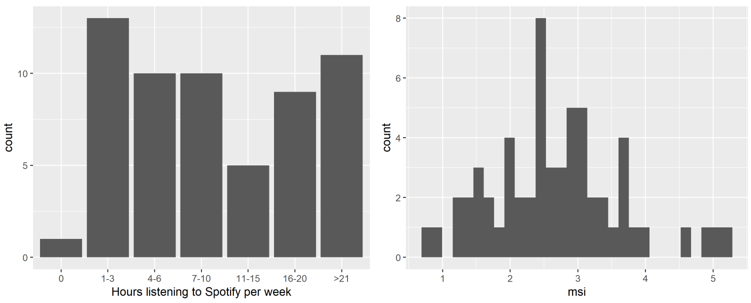


Figure 8: Left: number of hours participants were listening to Spotify (self-reported). Right: music sophistication scores on a scale from 0 to 6.

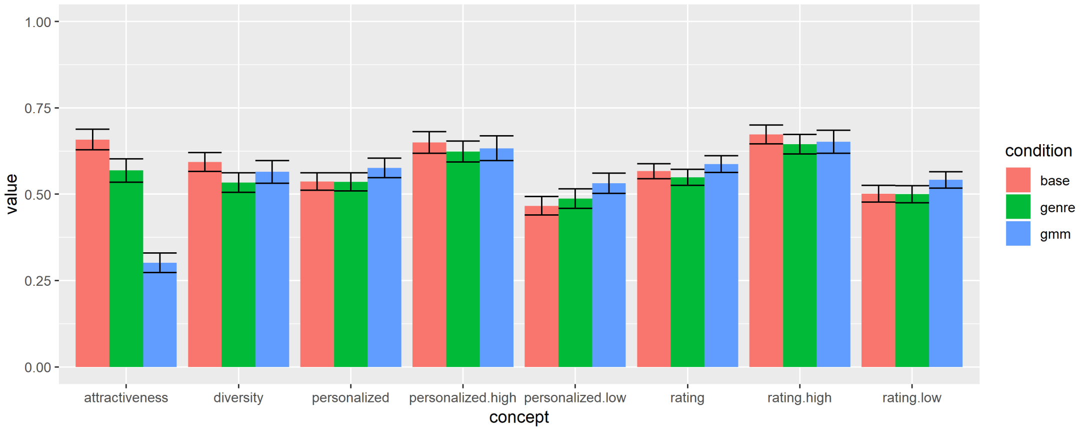


Figure 9: Bar chart of the measured concepts by condition. The value indicates the mean and the interval illustrates the standard error. The high and low indicators of the rating and personalization scores indicate that only the top 3 recommendations or the lower ranked (20th to 300th) recommendations are included.

The predicted user-item relevance scores are normalized between 0 and 1. User ratings were therefore scaled to the same scale, such that 5-star responses correspond to a score of 1 and 0.5-star responses correspond to a score of 0. Figure 9 displays the summary statistics of list attractiveness, list diversity, item rating scores, and item personalization scores per condition. Item ratings scores and item personalization scores are higher for the top 3 recommendations as compared to the worse (20th to 300th) recommendation in each condition. Perceived list diversity does not differ much per condition, whereas list attractiveness is highest for the 'base' condition, a bit lower for the 'genre' condition, and much lower for the 'gmm' condition.

An initial type II MANOVA examined attractiveness, diversity, personalization, and rating as dependent variables; condition as independent variable; and MSI scores and number of hours listening to Spotify per week as covariates. The multivariate result for MSI scores and Spotify usage were non-significant, Pillai's Trace $= .027, F(4, 166) = 1.16, p = .329$ and Pillai's Trace $= .014, F(4, 166) = 0.57, p = .686$, respectively.

The multivariate result for condition was significant, Pillai's Trace $= .375, F(8, 334) = 9.62, p < .001$, indicating differences in list attractiveness, list diversity, item rating, and/or item personalization between the conditions. The univariate F tests and Tukey's multiple comparisons of means for the significant F statistics are reported in Table 9. These results are in line with Hypothesis 2 because they show that the track and artist methodologies generate significantly more attractive playlists compared to these same methodologies in conjunction with the genre or audio feature methodologies. However, the penalty that the audio feature methodology introduces to playlist

Table 9: Comparisons between algorithm conditions for the various dependent variables.

				Tukey's Multiple Comparisons of Means					
	ANOVA			genre-base		gmm-base		gmm-genre	
dependent	*F*	*df*	*p*	*diff*	*p*	*diff*	*p*	*diff*	*p*
user ratings	5.69	2, 2137	.003**	-.030	.082	.017	.430	.047	.003**
good	.701	2, 624	.497	-.030	.469	-.018	.763	.012	.888
bad	2.86	2, 1092	.058	-.020	.528	.025	.376	.045	.045*
personalization	6.09	2, 2137	.002**	-.017	.483	.034	.053	.051	.002*
good	.721	2, 624	.487	-.033	.461	-.020	.748	.013	.892
bad	4.98	2, 1092	.007**	-.002	.991	.054	.020*	.056	.014*
diversity	1.01	2, 171	.367	-.238	.333	-.114	.773	.124	.742
attractiveness	36.8	2, 171	***	-.358	.100	-1.43	***	-1.07	***

Note. Illustrates the results of univariate F tests on condition for each dependent variable in the MANCOVA. Stars indicate the following significance levels: $^*.05,^{**}.01,^{***}.001$. The 'good' subgroup for item attractiveness (user ratings) and item personalization indicate that only the top 3 recommendations are included in the analysis. Likewise, the 'bad' subgroups indicate that only the worse recommendations (20th to 300th) were included.

Table 10: Correlations and $RMSE$ of user rating scores and personalization scores with predicted ratings by the algorithms.

			Breusch-Pagan		NCV	
dependent	r	$RMSE$	BP	p	χ^2	p
user ratings	.384	.380	26.05	***	18.24	***
good	.403	.351	21.22	***	16.84	***
bad	.244	.405	5.25	.022*	3.40	.065
personalization	.416	.373	17.76	***	11.63	***
good	.425	.344	21.58	***	15.10	***
bad	.317	.396	2.20	.138	1.33	.248

Note. r = Pearson's correlation, $RMSE$ = Root Mean Squared Error, BP = Breusch-Pagan test statistic. Only pairwise complete data is included. Stars indicate the following significance levels: $^*.05,^{**}.01,^{***}.001$. The 'good' subgroup for item attractiveness (user ratings) and item personalization indicate that only the top 3 recommendations are included in the analysis. Likewise, the 'bad' subgroups indicate that only the worse recommendations (20th to 300th) were included. The table includes the Breusch-Pagan test and the NCV test for heteroskedasticity (Breusch & Pagan, 1979) for a linear model containing the user ratings as dependent variable and the algorithmic ratings as predictor. $n = 2140, n_{good} = 627, n_{bad} = 1095$.

attractiveness is significantly more severe compared to the genre distance methodology while both methodologies achieve the goal of preventing disjoint sets of individual recommendations. Therefore the genre distance methodology seems to be the better choice between the two high-spread methodologies.

To answer Hypothesis 3 we analyzed the performance of the predicted user-item relevance by comparing them with the user ratings. Table 10 illustrates the Pearson correlations and $RMSE$ for the user rating and personalization scores. The $RMSE$ should be interpreted carefully because the user ratings are not yet transformed using a linear regression model. Since the correlations appear to vary between the 'good' (top 3) and 'worse' (20th to 300th) recommendations, the relation between user ratings and predicted user-item relevance is visualized in Figure 10. The relation is visualized for the 'good' and 'worse' recommendations separately as well.

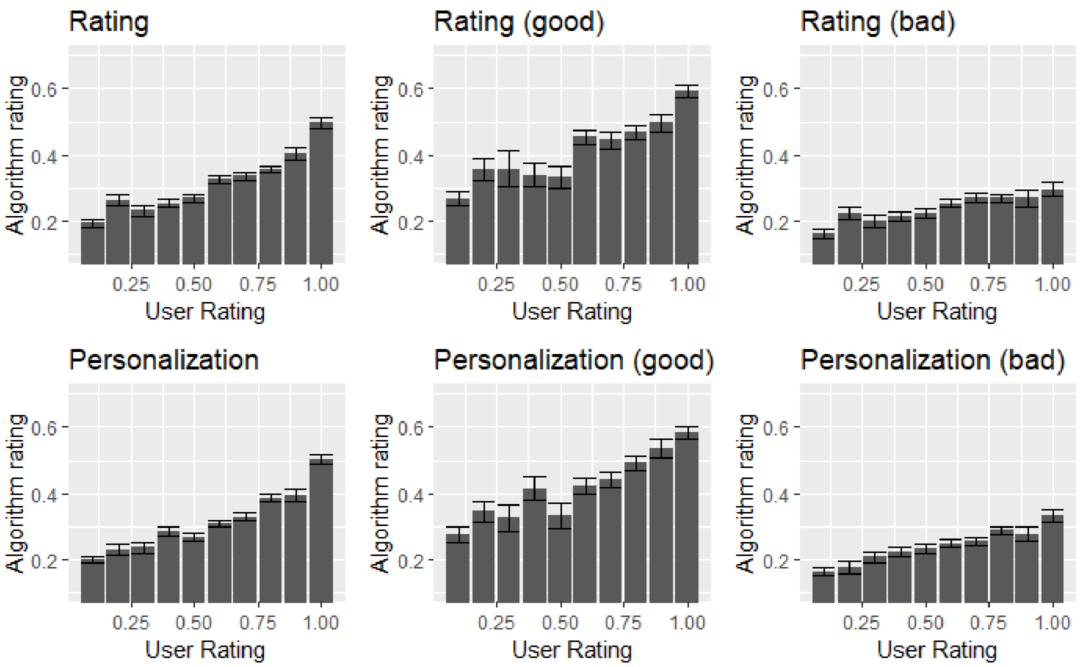


Figure 10: Bar charts of the item rating scores, item personalization scores, and the ‘good’ and ‘bad’ recommendation subgroups against the predicted item-user relevance scores (algorithm rating). The error bars indicate standard errors. $n = 2140, n_{good} = 627, n_{bad} = 1095$.

The relation between user rating and algorithm rating seems to be mostly linear, although 5-star user ratings are accompanied by relatively higher algorithm rating scores compared to the linear trend. Therefore, we analyzed the skedasticity of the residuals of the linear trend, see Figure 12. The Breusch-Pagan and NCV tests for heteroskedasticity are reported as well in Table 10 (Breusch & Pagan, 1979). The results indicate that items with high predictions (algorithm rating > 0.7) for the lower ranked recommendations (20^{th} to 300^{th} recommendation as compared to the top 3) received user rating scores lower than expected by the linear trend.

Linear regression models for predicting the user ratings are reported in Table 11 and illustrated in Figure 11. These models show that the slope between the item-user relevance and user ratings is positive regardless off condition and recommendation quality which is in line with Hypothesis 3. The regression lines for the base condition lie above those of the other conditions which is in line with Hypothesis 2, because it shows that the accuracy of the low-spread methodologies is highest. The slope for the ’worse’ predictions in the genre condition is significantly higher than those of the other conditions. There is no theoretical explanation for this finding.

Another method for evaluating how well item-user relevance predicts user ratings is to classify both scores into two groups: positive prediction or negative prediction, and liked by user or disliked by user. After the ratings are classified, the confusion matrix metrics (Figure 13a), accuracy metrics (Figure 13b), and ROC-curves of the three algorithms (Figure 14) are computed.

There are various small differences in the classification metrics between the conditions. A false negative (type II error) is worse than a false positive (type I error) because it is important to prevent recommending items that people do not like. If an item is recommended that has a low predicted item-user relevance for a particular user there probably is a good reason for it (e.g. it is aggregated with other profiles). If this recommendation then turns out to be liked (i.e. false negative) than this is not necessarily bad. This means that recall of the algorithms is significantly more important than precision in this application.

The track and artist distance methodologies have the lowest number of false negatives over all classification thresholds, see Figure 13a. The genre and audio feature methodologies have a

Table 11: Linear regression models for predicting user rating scores and user personalization scores. Independent variables include the algorithm scores and the condition.

dependent	F	*df*	R^2_{adj}	x	β	t	p
user ratings	129.5	3, 1750	.180	score	.510	7.83	***
				genre	-.147	-5.46	***
				gmm	-.108	-3.97	***
				score*genre	.163	1.98	.048*
				score*gmm	.001	.010	.995
				constant	.507	26.1	***
good	51.46	3, 554	.214	score	.581	5.93	***
				genre	-.081	-1.42	.157
				gmm	-.204	-3.34	***
				score*genre	-.014	-.104	.917
				score*gmm	.084	.666	.506
				constant	.498	12.7	***
bad	28.15	3, 829	.089	score	.367	2.27	.024*
				genre	-.183	-4.28	***
				gmm	-.089	-2.02	.044*
				score*genre	.351	1.88	.061
				score*gmm	-.007	-.039	.969
				constant	.536	16.0	***
personalization	147.7	3, 1750	.201	score	.612	9.06	***
				genre	-.130	-4.63	***
				gmm	-.094	-3.34	***
				score*genre	.107	1.25	.211
				score*gmm	-.036	-.445	.657
				constant	.461	22.9	***
good	57.66	3, 554	.234	score	.673	6.56	***
				genre	-.070	-1.18	.240
				gmm	-.210	-3.29	.001**
				score*genre	-.061	-.446	.656
				score*gmm	.062	.474	.636
				constant	.447	10.9	***
bad	41.43	3, 829	.127	score	.501	2.98	.003**
				genre	-.175	-3.95	***
				gmm	-.109	-2.39	.017*
				score*genre	.302	1.56	.119
				score*gmm	.095	.493	.622
				constant	.490	14.1	***

Note. *df* = degrees of freedom, x = predictor, genre = condition 'genre', gmm = condition 'gmm'. Stars indicate the following significance levels: *.05,** .01,*** .001. The 'good' subgroup for item attractiveness (user ratings) and item personalization indicate that only the top 3 recommendations are included in the analysis. Likewise, the 'bad' subgroups indicate that only the worse recommendations (20th to 300th) were included. The independent variable 'score' is the predicted user-item relevance.

slightly higher proportion of false negatives. They differ mostly in the proportion of false negatives in the high classification threshold region. This directly influences the recall of the algorithms. The recall reaches one at a threshold of approximately 0.63, 0.75, and 0.95 for the 'base', 'genre', and 'gmm' conditions respectively.

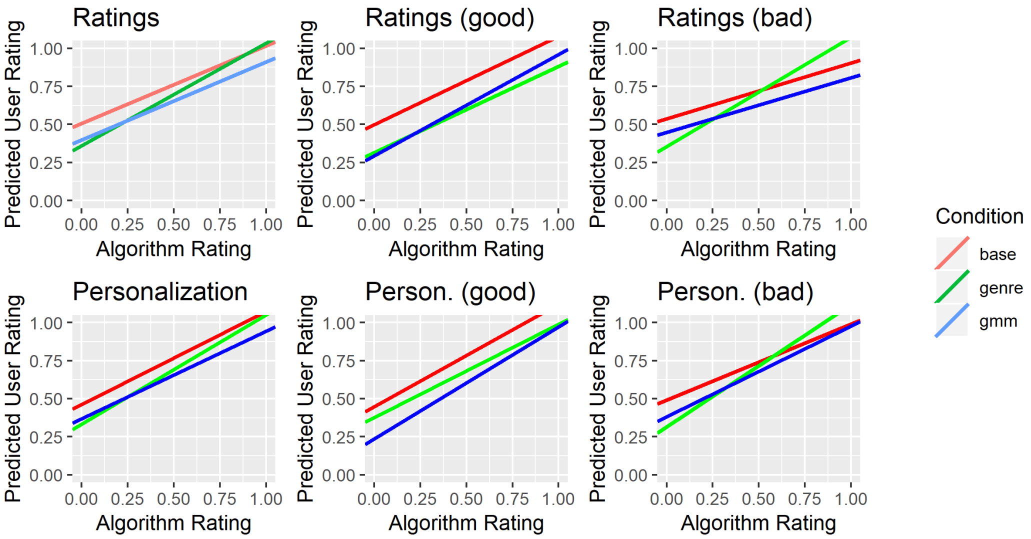


Figure 11: Linear regression models for predicting user rating scores (top) and user personalization scores (bottom). Independent variables include the algorithm scores and the condition. The 'good' subgroups indicate that only the top 3 recommendations are included in the analysis. Likewise, the 'bad' subgroups indicate that only the worse recommendations (20^{th} to 300^{th}) were included.

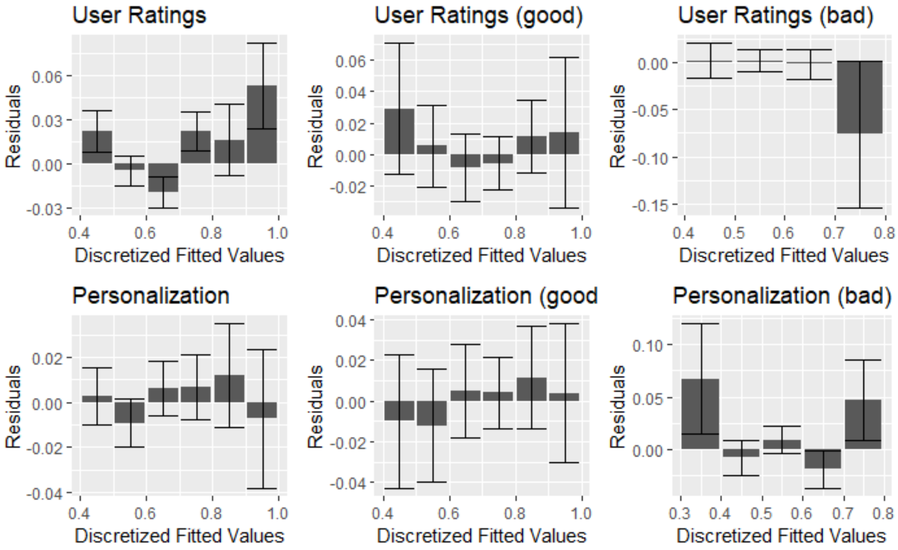
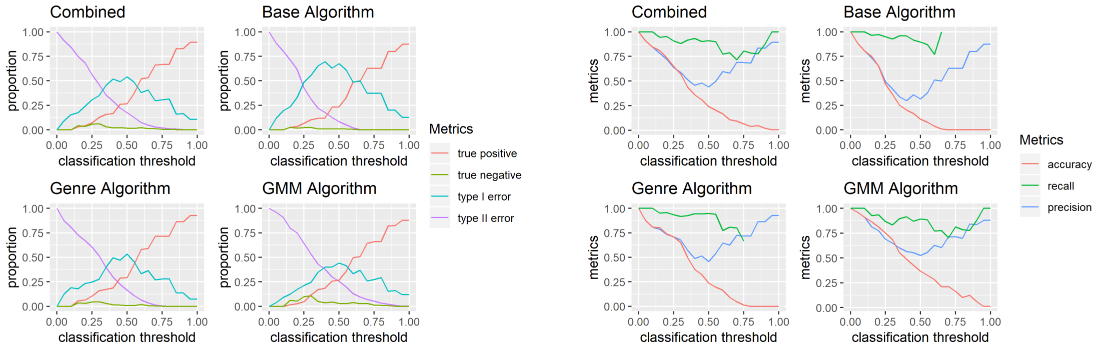


Figure 12: Bar charts displaying mean and standard deviations of the residuals against the predictions of linear models for the item rating scores and item personalization scores. $n = 2140, n_{good} = 627, n_{bad} = 1095$.



(a) base metrics

(b) derived metrics

Figure 13: Classification metrics of the recommendation strategies for varying classification thresholds.

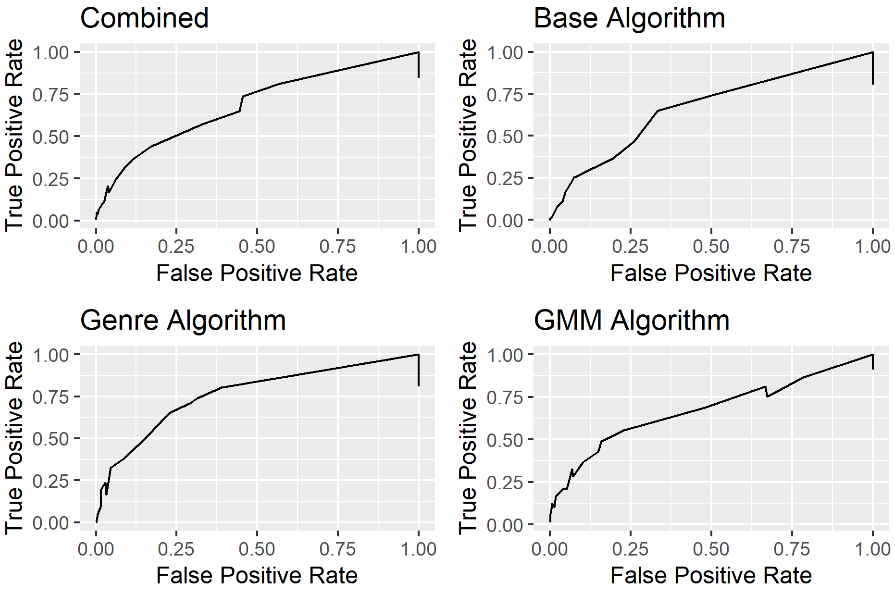


Figure 14: ROC-curves of the recommendation strategies used in the conditions for varying classification thresholds.

6.3 Discussion

The first interesting result is the difference in perceived attractiveness between the conditions. The condition with the track and artist distance methodologies (i.e. the low-spread methodologies) generated playlists with the highest perceived attractiveness. The addition of the genre distance methodology (in conjunction with the track and artist methodologies) significantly decreased the perceived accuracy of the recommendations, but not as severely as the audio feature methodology did. The purpose of the high-spread methodologies is to solve the problem that occurs when predictions for individuals are disjoint sets and have to be aggregated to group predictions. Both the genre distance methodology and the audio feature methodology are able to prevent disjoint sets in the individual recommendations, but the genre distance methodology seems to do this with a much smaller penalty to accuracy compared to the audio feature methodology. Therefore, the genre distance methodology appears to be the preferred high-spread methodology.

One limitation of this study is that there was no measure of whether tracks were known by the users. The mere-exposure effect hypothesizes that a list containing more items that a user knows will be more attractive compared to lists with fewer known items. Therefore, this may have confounded the comparison between the audio feature and genre distance methodologies. The genre distance might have recommended more items that the user knew compared to the audio feature methodology. However, independent of the reason it still holds that the genre distance methodology solved the disjoint set problem while only introducing a small penalty to perceived attractiveness.

The second finding is that in all conditions and for both higher and lower ranked recommendations there is a very clear linear relation between predicted item-user relevance and the user ratings. Items with a higher predicted item-user relevance receive higher user ratings on average. The relation is not one-on-one in the sense that an algorithm rating of 0 does not receive a user rating of 0 on average. The results indicate that given enough data, the predicted item-user relevance scores could be mapped to the corresponding average user ratings by a linear function.

The linear relation between predicted item-user relevance and user ratings seems to vary between the conditions, see Figure 11. The track and artist distance methodologies receive higher user ratings overall, but the slope between item-user relevance and user rating is nearly identical between the conditions. The genre distance methodology is an exception, since the slope seems to be significantly steeper, especially for the lower ranked recommendations. This may be an artifact of the used parameter values (e.g. various weight values), but this could also indicate that people have a tendency to like recommendations based on genre. The top 3 recommendations have a higher probability of containing tracks that also score well in the track and artist distance methodologies, which may explain why the slope of the 'genre' condition is not steeper for the higher ranked recommendations. The probability of recommendations being attributable to the genre distance methodology in the 'genre' condition is higher for the lower ranked recommendations.

The linear relations between item-user relevance and user ratings are not homoscedastic. A possible explanation could be that the item-user relevance available for the users are not uniformly distributed. Certain users may only have a small number of very good recommendations, a bunch of good recommendations, and a ton of mediocre recommendations. Therefore, the number of samples within certain regions of item-user relevance may vary greatly. This can have an impact on the scedasticity of the linear models. Another explanation could be that certain regions of the item-user relevance scores predict user ratings with more or less uncertainty compared to ratings in other regions. Given the results, especially the item-user relevance between 0.9 and 1.0 seem to have greater uncertainty in predicting user ratings. Overall, however, a linear model tends to underestimate the user ratings in these regions.

6.4 Conclusion

In all conditions higher predicted item-user relevance scores lead to higher average user ratings. Therefore, the methodologies proposed in Chapter 5 can be said to predict item-user relevance correctly. Both the genre distance methodology and the audio feature methodology can be used to solve the disjoint set problem when used in conjunction with the track and artist distance methodologies. However, the genre distance methodology does so while introducing a much smaller penalty to perceived playlist attractiveness compared to the audio feature methodology.

Chapter 7

Evaluation of the satisfaction function module

In Chapter 4 a framework for sequential group music recommendation systems was proposed. The main additions to the classical approach to group recommendation systems are the track weighting function and the satisfaction function. The track weighting function steers the playlist towards target characteristics while the satisfaction function is aimed to increase the perceived fairness of the system. However, whether such a satisfaction function actually influences the perception of fairness in the distribution of musical taste of group members within a music playlist is not known.

To evaluate the satisfaction function an implementation of a group recommendation system in which the satisfaction function can be embedded is necessary. In Chapter 5 such an implementation based on the framework is described. The accuracy of the implemented recommendation strategies was evaluated in Chapter 6 because affective state modelling using the satisfaction function module depends on the accuracy of the recommendation strategy. The conclusion of this evaluation is that the recommendation strategies produce item-user relevance scores that represent user likeability sufficiently. Therefore, these recommendation strategies can be used while evaluating the satisfaction function module.

In this chapter a study is described that explored the perception of fairness in the distribution of musical taste of group members within a music playlist. A focus group study was conducted to learn whether different satisfaction functions lead to different evaluations of fairness within a group music recommendation system. Additionally, it evaluated the implementation of the proposed framework in terms of usability and user satisfaction.

7.1 Method

7.1.1 Participants

Participants in this study included 18 people, of which all of them were students of the master course Creative Thinking and Innovative Design at the Jheronimus Academy of Data Science in Den Bosch, The Netherlands. Participants were invited to voluntarily sign-up for the study in groups of three during the course. The sample consisted of 12 males and 6 females. The age of the participants ranged from 22 to 29 ($M = 23.3, SD = 1.81$). Participants were required to have a Spotify account (free or Premium) and to have used this account prior to taking part in the study.

7.1.2 Design

This study followed a semi-structured focus group design with groups of three people each. Prior to the study we composed an interview plan with the questions and topics that we want to cover, see Appendix B. During the study, we could ask questions not in the interview plan when needed. Satisfaction with the recommended music, ease of use, interaction adequacy, interface adequacy, transparency, intention of use, and usefulness were measured for evaluating the usability of the system.

Additionally, the focus groups were requested to interact with our sequential group music recommendation system in three different within-subjects conditions. The conditions were three different satisfaction functions and all other system components were fixed during the study. The satisfaction functions (see Section 5.4) were the 'equal' satisfaction function which used fixed user weights, the 'delta' satisfaction function which emphasized a particular user, and the 'emotional decay' satisfaction function which dynamically balanced user importance based on affective state modelling. The order of the conditions was randomized prior to each session. After each condition we measured individual and group evaluation of perceived fairness and satisfaction with the recommended music.

Modules in the group recommendation system

The modules other than the satisfaction function module of the group recommendation system were fixed. For an overview and description of the implemented modules, refer to Chapter 5.

The recommendation strategy was an equally weighted combination of the Track Weighting Methodology and the Artist Weighting Methodology as recommendation strategy (as in the 'base' condition used in the study described in Chapter 6). This recommendation strategy was chosen because it was the strategy that recommended the most attractive recommendations and because it is the fastest in terms of computation time.

The track weighting function used an exponential decay function that prevents track (and later on also artist) repetition, but gradually allows repetition over time by increasing the weights using an exponential. The exponential used a halftime of 45 minutes and applies a weight penalty of 1 to newly recommended tracks and a weight penalty of 0.8 to artists of the newly recommended track. These parameters were tuned based on the session time of the study and by means of trial and error and inspection.

The satisfaction function varied based on the condition. In the 'equal' condition user weights of 1 were used. In the 'delta' condition a weight of 0.8 was used for the overly represented person and a weight of 0.3 for the other users. These values were chosen such that the amount of unfairness was presumably large enough to be noticed but was not completely removing the user importance of the other users. The 'emotional decay' condition used a decay factor of $\delta = 0.8$. A large value was chosen such that predicted user satisfaction decayed relatively slowly over time. The effect of user satisfaction on impact was not modelled $\epsilon = 0$ for simplicity. A unity decay factor was used $\lambda = 1$.

The aggregation function that we used was the 'highest' module which combined all items for which the recommendation strategy produced item-user relevance scores and recommended the item with the highest relevance score.

7.1.3 Materials

After each condition each participant individually filled in a recommendation system evaluation scale measuring accuracy, novelty, diversity, and perceived fairness of the recommended items on a 5-pt Likert scale which was created for the purpose of this study. The purpose of this scale was primarily to privately measure perceived fairness of the satisfaction function used in the condition. The scale is presented in Table 12.

Participants also filled in a general system evaluation scale measuring interface adequacy, interaction adequacy, ease of use, transparency, usefulness, overall satisfaction, and intent of use which was created for this study. This scale was only filled in once by the participant and was aimed for measuring the usability of the group recommender. The scale is presented in Table 13. Additionally, participants filled in the demographics scale, see Table 8 such that the participants of the study could better be described.

Table 12: Recommendation system evaluation scale.

concept	item	framing
recommendation accuracy	The items recommended to us matched my interests.	+
	The items recommended to us matched my group's interests.	+
recommendation novelty	The items recommended to us are novel.	+
	The recommender system helped us discover new music.	+
recommendation diversity	The items recommended to us are diverse.	+
fairness	The recommended items matched the interests of the other group members more than my own.	−
	All group members' interests were fairly expressed in the recommended items.	+
	My personal musical taste was represented in the recommended items.	+
	Some group members might be dissatisfied with the recommended items.	−

Note. A '+' indicates a positive framing and a '−' indicates a negative framing.

An interview guide was created for note taking during the sessions. The guide contained the following implicit measurements: whether any participants had a leading role in the group, and whether participants requested help during the login phase. The guide also contained a number of group questions that were to be asked after the group interacted with each condition: "*Did the recommender overly represent the musical interests of certain group members over those of others? If yes, please indicate in what order your musical tastes were represented*", and "*If you would grade this recommender based on how it is able to play music that all group members enjoy listening to, what would it be? Please use a scale from 1 to 10*". Additionally, the guide contained the following topic points: useful contexts/situations for the system; intent of use regarding the discovery of their friends' musical preferences; characteristics about the system they liked, disliked, or missed; and ideas for improvements.

The system was running on a virtual server located at Delft University of Technology and a client connection was instantiated prior to each session on a laptop. The laptop was connected to a large touch screen display visible and accessible to the focus group. The laptop was also connected to two speakers through which the music could be played.

7.1.4 Procedure

Participants arrived in groups of three at the location of the study. Prior to the study, all participants were required to sign a consent form. After the forms were signed, they were introduced to the study. Participants were told they were to interact with the group music recommendation system during the study and they were encouraged to share their thoughts. Then, we asked for permission to make an audio recording of the session, and the recording was started after we received their consent.

After the participants were briefed and the recording was setup, we asked participants to login to the system without providing any further information. Only when participants were unable to login and when they were not helping each other, we provided them with further instructions. Participants had to login by browsing to https://groupify.pw/ on their mobile device (see Figure 15), entering a PIN code displayed on the screen (see Figure 16) and signing-in with their Spotify account. Additionally, participants could download an Android or iOS app by scanning a QR-code or searching in the app stores. However, when participants did so we directed them back to browsing to the website instead as the apps were not yet functional.

After all group members were signed-in to the system, we requested them to interact with the

Table 13: General system evaluation scale.

concept	item	framing
interface adequacy	The layout of the recommender interface is attractive.	+
	The layout of the recommender interface is adequate.	+
	The icons and labels used in the recommender interface are clear.	+
interaction adequacy	The recommender allows me to tell what I like/dislike.	+
	I found it easy to tell the system what I like/dislike.	+
	I found it easy to inform the system if I like/dislike the recommended item	+
ease of use	I became quickly familiar with the recommender system.	+
	The recommender system was easy to use.	+
transparency	I understood why the items were recommended to us.	+
usefulness	The recommender helped me discover the musical tastes of my group members.	+
	The recommender helped us find music that we all like.	+
	The recommender is useful to find music we like.	+
overall satisfaction	Overall, I am satisfied with the recommender.	+
	I enjoyed using the recommender.	+
use intentions	I will use this recommender again.	+
	I will use this recommender frequently.	+
	I will tell my friends about this recommender.	+

Note. A '+' indicates a positive framing and a '−' indicates a negative framing.

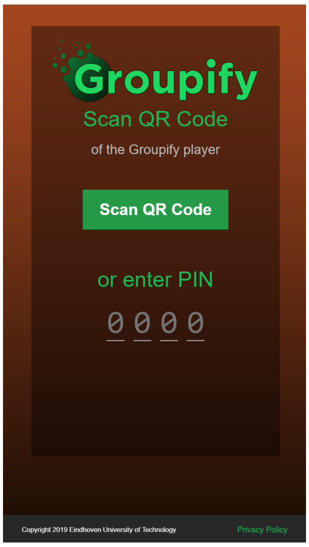
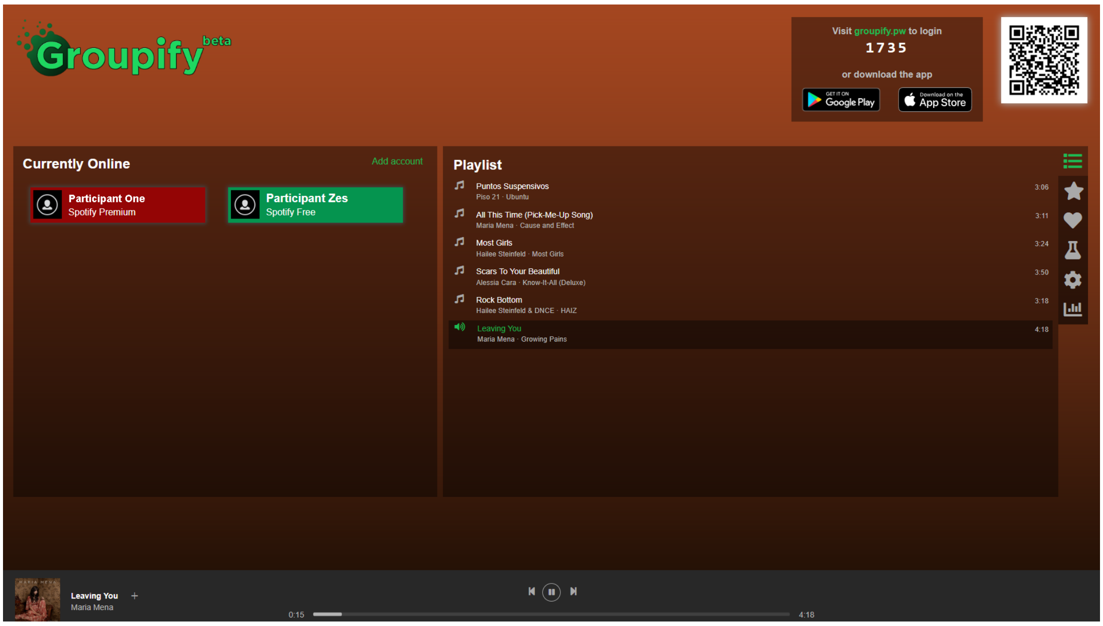


Figure 15: Screenshot of the login website displayed on the mobile devices of the participants.

system and to listen to and talk about the music for about 10 minutes after which participants received individual forms with the recommendation system evaluation scale. When they finished, we asked the group whether they felt the music represented the musical tastes of all group members fairly. If they thought the music was not fairly distributed, they were asked to indicate which group members received more music than others. Then, we asked how they would grade the music as a group based on how much they like the music. When they came up with a consensus, we changed the condition of the system and asked them to interact with the system again. This is repeated for all three conditions.

After these steps were repeated for all three conditions, we asked several questions to give rise to group discussions. First, we asked in which contexts or situations they would find this recommendation system useful. Then, we asked whether they would actually like to use the system to discover their friends' musical tastes. Finally, we asked what they liked, what they missed, and whether they have ideas for improvements.

After the discussions were finished we handed the participants general system evaluation scale and the demographics scale after which they were debriefed. The participants received their reward and were thanked for their time and effort.



Visit groupify.pw to login
1735
or download the app

Currently Online

Participant One
Spotify Premium

Participant Zes
Spotify Free

Playlist

Puntos Suspensivos Piso 21 · Ubuntu	3:06
All This Time (Pick-Me-Up Song) Maria Mena · Cause and Effect	3:11
Most Girls Hailee Steinfeld · Most Girls	3:24
Scars To Your Beautiful Alessia Cara · Know-It-All (Deluxe)	3:50
Rock Bottom Hailee Steinfeld & DNCE · HAIZ	3:18
Leaving You Maria Mena · Growing Pains	4:18

Leaving You +
Maria Mena
0:15 4:18

Figure 16: Screenshot of the recommendation system used in the focus group study.

7.2 Results

The study took place between the 4th of February and the 8th of February of 2019. In this section we first describe the qualitative results that we obtained during the study. Then, we shortly summarize the quantitative results. Only descriptive statistics and basic quantitative analysis are presented due to the limited limited number of participants and thus power.

7.2.1 Qualitative analysis

We analyzed the results of the qualitative data by tabulating for each concept/question the notes taken and the quotes of the participants. Then, a summary is composed of the notes and quotes after which the summary is interpreted. The analysis is reported in Tables 14 and 15.

Table 14: Qualitative analysis performed for study 2. Part 1 of 2.

Question	Notes/Quotes	Summary	Interpretation
	Login Ease of Use		
Observation of login task	Group identified the web address rapidly and proceeded to login One participant logged in rapidly and proceeded with helping the others One group started scanning the QR codes and downloading the app. After specific instructions they entered the web address instead "Can I login with Facebook? Do I need to login with the url" Most participants did not need to fill in their Spotify credentials	In most groups, at least one person identified a correct login method. This person seems to be able to help the others with signing-in. Additionally, most participants did not need to memorize Spotify credentials.	The login method seems intuitive and easy to use.

Note. The notes and quotes are based on note-taking done by an assistant during the focus group sessions. This is part 1 of 2 of the analysis.

The main implication of the login task is that the login procedure of the system is sufficiently clear and straight forward. All participants managed to login to the system. Usually one participant quickly learned how to login and helped the others with the login task afterwards. While most participants initially tried to login using the PIN system, some participants attempted to download the apps. While this is an intended login procedure, these participants were directed to using the PIN system as the apps were not yet fully functional.

Another implication is that the recommendation system can be used for at least two different purposes: exploration of friends' musical interests and background music. Most participants mentioned that they would not like to use the system for background music because it was too much of a distraction. Some participants thought this was because it elicits discussion. While most participants mentioned the exploration purpose, some participants did thought it was best suited for use as background music. This shows that different participants may have different purposes in mind when using a group music recommendation system. When participants envisioned using the system for exploration purposes they mostly mentioned social gatherings as contexts of usage. As a group, participants showed intention of using the system.

The results of what users liked or disliked is reported in Figure 17. All groups indicated that they would like to be able to set explicit preferences. They mainly mentioned they would like to select a genre, but some participants would also like to set a context, exclude genres, or set an attribute (like tempo). One group mentioned they want to be able to tell the system to play

Table 15: Qualitative analysis performed for study 2. Part 2 of 2.

Question	Notes/Quotes	Summary	Interpretation
Contexts/Situations of Use			
Would you use this recommender system as a group, for example while working on a project?	“It can be too distracting because of the change of genres, and also because of people would talk about the music” ”Would be messy”, ”I won’t use it”, ”Not for studying”, ”for dinner”, ”Mood would be available”, ”music should change according to mood” One group said they would use it. Another said they’d only use it for background music, and yet another said they’d use it if the music represents everyone well.	Many participants think it is best suited for social gatherings like parties, dinner, and drinking beers. Some indicate this is because it elicits discussion. A smaller number of participants mention they’d use it as background music.	There are two different purposes for the system: exploration of your friends’ musical taste and background music. When the system is used for the purpose of exploration, it is best suited in social gatherings.
In what context or situation would you find this recommender to be most useful?	”Not for party or work. To drink a beer or in a situation where you are not caring to discuss about the songs in the playlist” ”social, hanging at home with friends, like background” ”contexts that do not require attention, like a party” Two groups mentioned both parties and projects ”certain activities with friends, background music, with friends when having drinks” ”ambient music, work, group project, restaurant, bar, break, chilling with group”		
Intent of Use			
Would you use this recommender system with your friends to discover each other’s music tastes and preferences?	One group would not use it, because they would prefer to check their friends’ Spotify profiles when they want to know their musical tastes. All other groups did want to use the system. ”yes, it is fun”, ”If the app is working”, ”I would like to use it”, ”Maybe I’ll try” ”Yes, as long as it indicates a taste of my friends.”, ”Yeah, I would like to try to use in a group.”, ”It’s fun”	Most participants indicated they would want to use the recommendation system.	The recommendation system is attractive such that participants indicate intent of use.

Note. The notes and quotes are based on note-taking done by an assistant during the focus group sessions. This is part 2 of 2 of the analysis.

original versions of songs instead of remixes. Some groups mentioned artist repetition occurred too often and some groups thought the music contained too many different styles.

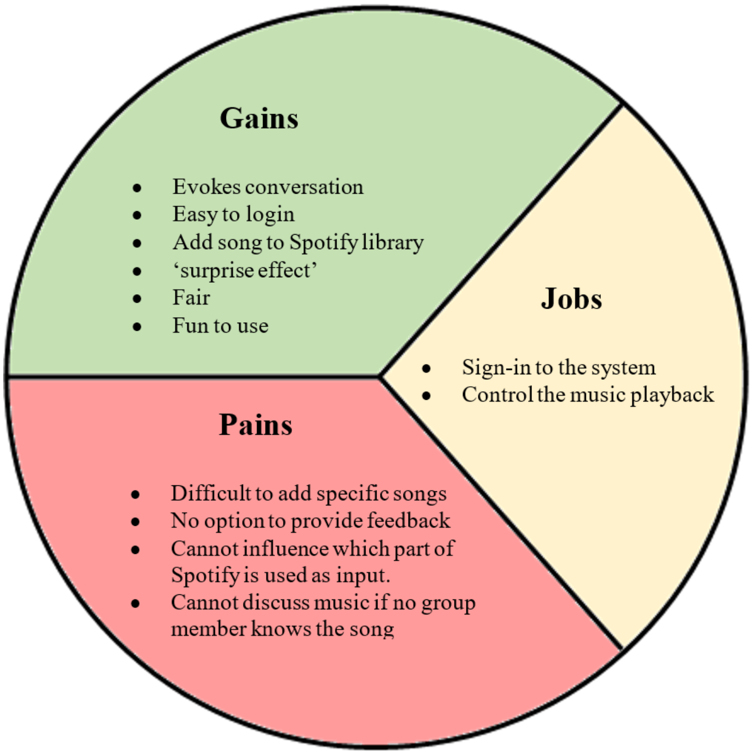

Gains
Evokes conversation
Easy to login
Add song to Spotify library
'surprise effect'
Fair
Fun to use
Jobs
Sign-in to the system
Control the music playback
Pains
Difficult to add specific songs
No option to provide feedback
Cannot influence which part of Spotify is used as input.
Cannot discuss music if no group member knows the song

Figure 17: Participant profile that summarizes what the participants liked and disliked about the system.

Participants also mentioned several improvements other than the features they missed. Some participants thought it would be a good idea to use a voting system to decide the next item. Some participants wanted to be able to scroll back to see the history of played tracks. It was also mentioned that the user profile cards are too salient and that the space could be used for showing the album cover instead.

There were many elements of the user interface that were not very intuitive. People tried to scroll in the playlist and the control seek bar could not be dragged. Additionally, people tended to click on their user cards while the programmed action (user logout) did not match their intention. Some participants actually mentioned that the user cards are too big and not intuitive and necessary. We can conclude that the implementation of the user cards and some other elements are not very intuitive.

All groups noticed in the 'delta' condition, which used a satisfaction function that gave more importance to a particular group member, that the person who received a higher user weight received an unfair amount of music. Even though the unfairness was noticed, some groups would still attribute the recommendations to some form of profile aggregation. Some participants thought to know how the system was working and tried to explain why certain recommendations were decided. The explanations were often related to recent listening behavior, but were less sophisticated than how the group recommender actually worked.

7.2.2 Quantitative analysis

Prior to the quantitative analysis negatively phrased items are inverted and concept scores are aggregated by summing the individual items and dividing by the number of items per concept. All concept scores are rescaled such that the lowest score corresponds to 0 and the maximum score corresponds to 1.

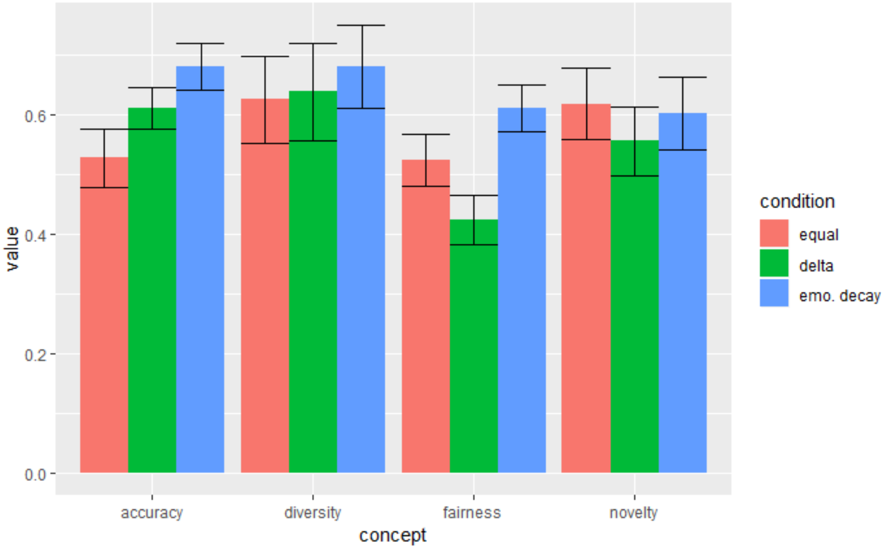


Figure 18: Mean values (bar height) and standard errors (error bars) of accuracy, diversity, novelty, and fairness measured with the recommendation system evaluation scale individually after each condition. The scale ranges from 0 to 1.

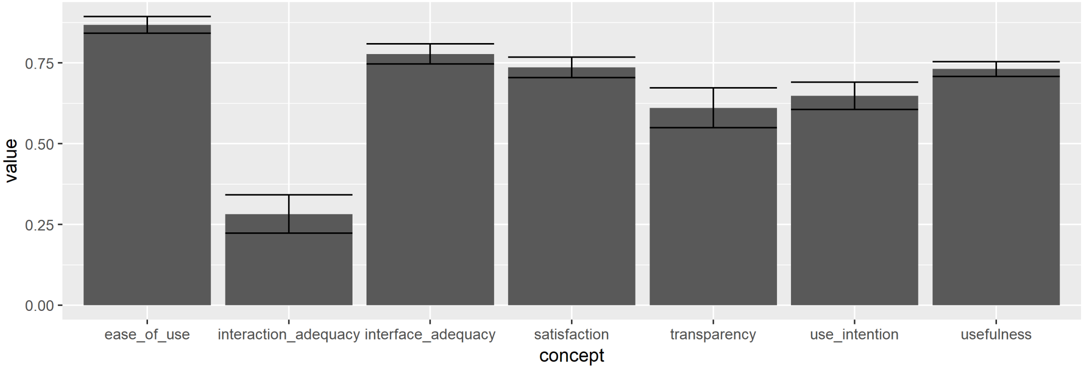


Figure 19: Mean values (bar height) and standard deviations (error bars) of the concepts measured with the general system evaluation scale at the end of each focus group session. The scale ranges from 0 to 1.

Responses to the recommendation system evaluation scale which measured accuracy, diversity, novelty, and fairness, are illustrated in Figure 18. The figure indicates that the satisfaction function ‘emotional decay’ is the fairest, followed by ‘equal’ and ‘delta’. However, a linear regression of condition on fairness is not significant for ‘delta’, $t(51) = -1.720, p = .09$, and ‘emotional decay’, $t(51) = .087, p = .14$, as compared to the ‘equal’ condition. This is most likely related to the limited number of participants.

Responses to the general system evaluation scale are presented in Figure 19. Ease of use was evaluated most positively of all usability measures ($M = .87, SD = .11$) and only interaction adequacy ($M = .28, SD = .25$) scored below the neutral option of the Likert scale on average. While many disadvantages and improvements about the interface were given by the participants the interface adequacy still scored highly ($M = .78, SD = .13$). The participants were satisfied ($M = .74, SD = .13$) and thought the recommender system was useful ($M = .73, SD = .10$).

On the group level fairness and satisfaction of the recommendations were measured. The results are illustrated in Figure 20. Four out of six groups thought the recommendation system’s

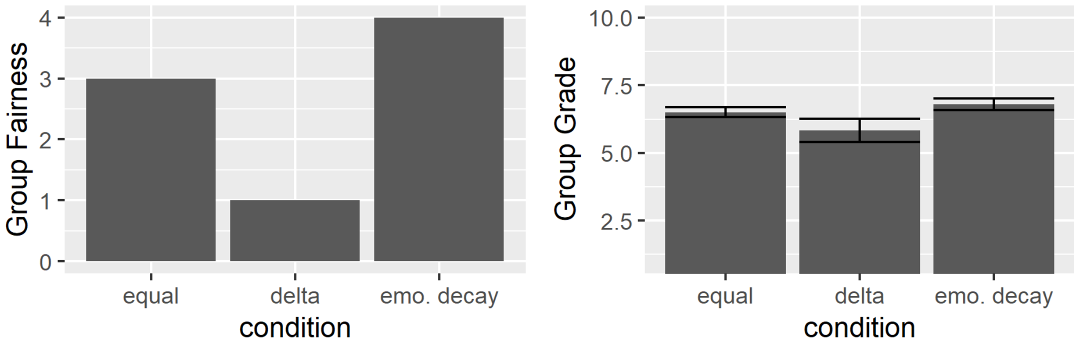


Figure 20: Summary of the responses to the questions asked to the group after the group had interacted with each condition. The left graph is a bar chart of the number of times the group thought the distribution of musical taste within the recommendations was fair. The right graph summarizes the consensus grades given by the group on how much they like the recommendations as a group.

recommendations were fairly distributed amongst the musical tastes of all group members while the 'emotional decay' satisfaction function was used. Three groups thought so for the 'equal' satisfaction function and one group for the 'delta' satisfaction function. The consensus grades given to the recommendations in general in terms of how much the participants liked the music as a group did not vary a lot between the conditions ($M = 6.4, SD = .79$).

7.3 Discussion

The main interest of this study was whether people perceive differences in fairness (i.e. how recommended items spread of the personal interests of all group members) due to the satisfaction function module. Within the framework proposed in this thesis, objective fairness is operationalized as a satisfaction function. The 'equal' satisfaction function was used as a control condition, since this is similar to having no satisfaction function at all. The 'delta' condition was purposely created to be unfair, but was still aggregating interests of all group members. The 'emotional decay' condition used a satisfaction function that dynamically balanced the user weights based on affective state modelling aimed to increase the fairness compared to using equal weights.

Interestingly, all groups in the 'delta' condition identified that the system was not fair. Additionally, they all correctly identified the person whose musical interests were overly represented. Even though the sample size is not large, the results are saturated in this regard. Participants are able to perceive when the recommendation system was unjust. The fairness metric in the recommendation system evaluation scale scored lowest for the 'delta' condition, which further validates this result.

The 'emotional decay' satisfaction function was perceived to be more fair on average than the control condition in both the group-level and individual-level evaluations, although the results are not statistically significant. The sample size in this study was not adequate to show a significant difference between these satisfaction functions.

The secondary purpose of the study was to evaluate the usability of our implementation. All participants managed to sign-in to the system and no improvements to the sign-in method were given by the participants. The interface design did receive a lot of feedback and several points for improvements were learned. There were also a lot of problems with the way people had to interact with the system. Overall, however, most usability measures were evaluated positively.

An interesting finding is that multiple purposes for using the group music recommendation system were mentioned. Prior to the study, we believed the system would be most useful for use as background music in contexts where people required concentration. However, most groups stated they would like to use the system to explore the musical interests of their friends. They would dislike using the system as background music while working, because it elicits discussion and requires attention. People who had the purpose of exploration in their minds disliked it when the system played music that nobody knew. While people mostly mentioned the purpose of exploration, some mentioned the usefulness of the system for playing background music. We believe GMRS can be used for both purposes, but the components of the system should be adapted to the purpose of the users.

When a group of users have the purpose of exploration, it might be best to use recommendation strategies that stay close to the user profiles (e.g., the track and artist distance methodologies). Additionally, aggregation strategies that use some form of averaging should be avoided because this results in group predictions that lie in the average of the interests of the individuals. These recommendations are typically not known by any of the users which would limit the extent to which individual users can identify with the music. A fairness aggregation strategy seems more suitable for this purpose.

For the purpose of background music, it might be better to use methodologies with a wider spread because it is not a problem when items are recommended that nobody knows. It should actually be avoided that recommendations stay very close to the musical taste of the users because this may elicit discussion. For this purpose, an averaging aggregation method is more suited.

There are several other findings derived from this study. All focus groups indicated that they missed the option to provide explicit preferences to constrain the music to a certain style or pick a genre. In terms of the framework, the track weighting function could be used to constrain the music based on such constraints. For example, users could be allowed to set a genre, optionally using the genre distance metric to assign lower nonzero weights to related genres. Other examples could be setting the mood or particular themes (e.g., Christmas music, German music, or danceable music). This can all be achieved by the track weighting function, provided that the appropriate tags and data are available in the metadata. While the above mentioned examples constrain the system to particular themes, the track weighting function could also be used to disable certain moods or themes instead.

All in all we can state that the implementation of group music recommendation system using the framework is suitable to be used for further research. However, there are many important improvements that should be made to increase the usefulness of the system, especially regarding user interaction, the user interface, and feedback elicitation. Additionally, variations in fairness are perceived by people and care should be taken to design satisfaction functions that fit the purpose of the system and that are perceived to be fair.

7.4 Conclusion

In sequential group recommendation systems it is not possible to always choose recommendations that everybody likes. The implementation determines how the recommendations are distributed within the personal tastes of the group members. People seem to perceive when a system unjustly overly represents the personal tastes of certain group members over those of others. Care should be taken to design a system such that groups or group members do not perceive it as being unfair.

People tend to describe two purposes for group music recommendation systems: exploring the musical tastes of their friends or colleagues and for use as background music. The requirements of the two purposes are not identical. When used for exploration purposes recommendations should closely match the personal tastes of individuals while for the background music purpose recommendations should be averaged out over the user profiles.

Chapter 8

General Discussion

This thesis aimed to combine theory and insights in group and sequential recommender systems in a generalizable and modular framework for sequential group music recommendation systems. This framework mainly introduces the satisfaction function module, which can be used for maintaining fairness in the recommendations, and the track weighting function module, which can be used for applying explicit or implicit preferences. An implementation of this framework was developed and evaluated in two user studies. The first study evaluated four methodologies aimed at generating item-user relevance scores based on ordered lists of top tracks and artists while solving the disjoint set problem. The second study aimed to show the relevance of the satisfaction function module in the proposed framework and to explore the usefulness and usability of the implementation in a focus group study.

The main contribution of this thesis is the description of the framework for sequential group music recommendation systems. The findings of the first user study were particularly important to confirm the effectiveness of the implemented recommendation strategies for solving the disjoint set problem. The second study showed that people are able to perceive changes in fairness based on user weighting. Dynamic balancing of user importance weights by means of affective state modeling, i.e. using the satisfaction function module, can improve the perceived fairness of group recommendation systems.

8.1 Relevance

The goal of this thesis is to describe a general modular framework for sequential group music recommendation systems. Such a framework should have merits beyond the scope of an individual study. This framework brings together the fields of sequential and group music recommendation systems in a structured fashion. This is useful since it can help to place work related to these fields in a broader perspective. For instance, Masthoff and Gatt (2006); Piliponyte et al. (2013); Quijano-Sanchez et al. (2013) have worked on user satisfaction modeling and group dynamics which fit the satisfaction function module; Masthoff (2004) evaluated several aggregation functions; and Zhao, Willemsen, Adomavicius, Harper, and Konstan (2018); Liang and Willemsen (n.d.); Kaminskas and Ricci (2012) worked on user inaction, genre exploration, and contextual recommendations which fit within the track weighting module of the framework.

Despite their limited scopes the studies described in this thesis can show the usefulness and applicability of the framework. Furthermore, the modularity of the framework should positively affect its external validity. The modularity allows results from various topics (i.e. aggregation techniques, contextual recommendations, recommendation strategies) to be combined together in a structured fashion and to be investigated separately. Additionally, modular recommendation systems increase the extensibility of existing modules and decreases the effort required to implement new algorithms.

A specific framework intended to combine sequential recommendation with group recommendation for music has not been described in earlier research. However, the idea of modular recommendation systems is not new and frameworks for more general recommendation systems have been proposed. For instance, Yang, Bagdasaryan, Gruenstein, Hsieh, and Estrin (2018) have described a modular framework for recommendation systems that can also support group recommendations. Their framework, OpenRec, is composed of three types of modules: interaction, extraction, and fusion. Interaction modules relate interactions between users and actions (i.e. behavior data) and can be related to the track weighting function described in the current framework. Extraction modules can be related to the recommendation strategy module, and the fusion modules relate to the aggregation function module. While OpenRec is similar in the sense that it describes a modular framework for recommendation systems, their framework is a technical implementation while the current framework is theoretical in nature. The current framework is therefore able to put theory in perspective while OpenRec is not. Furthermore, in contrast to OpenRec the scope of the current framework specifically addresses sequential group recommendation and puts emphasis on the various topics and challenges related to those fields. For instance, the satisfaction function is designed to be able to include effects from group dynamics whereas OpenRec only supports group recommendations as an extension to their framework.

8.2 Limitations

Although a modular framework based on user and item weights is beneficial, there also are disadvantages to using such a structure. First of all, the modular structure of the frameworks puts a constraint on interaction effects between the modules. For instance, the recommendation strategy cannot dynamically adapt to changes in explicit preferences (e.g. contextual constraints). This might be useful when session-based purposes for a group recommender are predicted during operation. The second study showed that people intent to use the system for exploration of friends' and colleagues' musical interests and background music. Depending on the purpose, it may be useful to adapt the recommendation strategy (e.g. change the weights between low-spread and high-spread methodologies, see Section 6.3). The modular composition of the framework does not allow the track weighting function to dynamically adapt the recommendation strategy. In that sense, the framework can be seen as being too simplistic for that particular purpose. During the development of the framework trade-offs were made between generalizability and simplicity. Although the framework could be generalized to include these interaction effects, this would increase the complexity to a too large extent.

Secondly, the dynamic nature of the framework constrains satisfaction functions and track weighting functions to real-time computation times. When a new recommendation is required, the item and user weights need to be available and adjusted to the current context depending on what kind of functions are used. For instance, a track weighting function that prevents repetition of tracks and artists needs to be updated using the current time and latest playback history. Solutions to this problem include caching and approximation techniques. Caching can be used such that portions of the functions are pre-computed prior to the request of a new recommendation.

Thirdly, there are certain design choices that may be constraining the scalability of the framework. For some recommendation systems is would make more sense to use weighting for concepts other than tracks. For instance, a track weighting function that assigns more weight to items of a particular genre would be unnecessarily complex. Such a track weighting function would need to find all tracks related to the genre and assign a weight value to each one of them ($complexity \approx \mathcal{O}(n_{\text{items}})$). In such a situation it would be more simplistic to use genre weighting, because this would only require a single weight value saving both computation time and memory. A solution for this problem is to replace the track weighting module with a more general music concept weighting module. This module would generate multiple output vectors (e.g., genre weights, track weights, artist weights) and indicate for each output vector whether items not in the vector should be included or excluded from the complete list of recommendations. Such a module would allow weighting to occur on multiple concepts at once (e.g., prevent track and artist repetition and constrain to particular genres at once) without drastically increasing computation time.

Finally, the framework is based on an aggregated predictions strategy for group recommendations. Therefore, group recommendation systems based on an aggregated models strategy are not supported by the framework. The aggregated models strategy is particularly useful when group members should have the option to explicitly adapt or negotiate group preferences or when there are privacy concerns regarding the use of individual user profiles (Felfernig, Boratto, et al., 2018).

8.3 Recommendations for future research

This thesis described a modular approach towards sequential group music recommendation. Our research showed the relevance and applicability of the framework, but did not fully validate it. The framework is also limited in scope since it is only described for the music domain. To determine whether the framework can be extended beyond the scope of music, further validation and evaluation of the framework should include other content domains.

A particularly interesting finding is that two purposes are identified for group music recommendation systems that have distinct requirements. An interesting question would be how a GMRS can adapt after its purpose is determined (either explicitly or implicitly). It might be a good idea to adapt to low-spread methodologies and fairness aggregation when the purpose of the users is exploratory in nature and use high-spread methodologies and averaging aggregation functions for background music purposes.

Only a limited number of modules were implemented during this thesis. Future research is needed to evaluate how more sophisticated modules (or different parameter values) can extend the functionality of recommender systems based on the framework. For instance, all participants indicated that they would like to set explicit preferences when using the recommendation system. Such functionality can be implemented in a track weighting module.

In summary, our framework provides a basis for a modular approach towards sequential group recommendation. Only the first steps towards validation and evaluation of the framework have been made. More research is needed to grow the framework into a more solid, applicable, and scalable state.

Chapter 9

Conclusions

This thesis described a theoretical framework that combines group recommendations with sequential recommendations in the music domain. The primary contribution of this framework is the introduction of the satisfaction function and track weighting function modules to the classical approach to group recommendation. The thesis showed how the theoretical framework can be applied in user studies to accumulate theory on various group recommender topics such as user satisfaction modeling and recommendation aggregation (e.g. Masthoff, 2015).

We have described an implementation of the framework with four methodologies for the recommendation system that solve the disjoint set problem. These methodologies were evaluated in an online study and were shown to be sufficiently accurate. A focus group study subsequently showed that the implementation had a good overall usability and also identified two purposes people had in mind for group music recommendation systems.

Finally, the usefulness of the satisfaction function module was evaluated. We showed that the satisfaction function module can influence the perceived fairness of group recommendation systems and that affective state modelling within a satisfaction function can be used to increase its perceived fairness.

Although this thesis has provided a theoretical platform for sequential group recommendation, there is still much work to be done to develop the framework into a platform that is thoroughly evaluated and that has a wider applicability and scalability. Moreover, the framework provides many endpoints for exciting future research in sequential group recommendation.

Bibliography

Baccigalupo, C. (2009). *Poolcasting: an intelligent technique to customise music programmes for their audience* (dissertation, Autonomous University of Barcelona). Retrieved from `http://www.iiia.csic.es/~enric/tesis/Baccigalupo-2009-PhdThesis.pdf` 23

Baccigalupo, C., & Plaza, E. (2006). Case-Based Sequential Ordering of Songs for Playlist Recommendation [Doctoral]. In T. R. Roth-Berghofer, M. H. Göker, & H. A. Güvenir (Eds.), *Advances in case-based reasoning: 8th european conference, eccbr 2006* (pp. 286–300). Fethiye, Turkey: Springer-Verlag. doi: 10.1007/11805816_22 5, 15, 16

Bonnin, G., & Jannach, D. (2014). Automated Generation of Music Playlists: Survey and Experiments. *ACM Computing Surveys*, *47*(2), 1–35. doi: 10.1145/2652481 3, 4, 6

Boratto, L., & Carta, S. (2010). State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups. In A. Soro, E. Vargiu, G. Armano, & P. Gavino (Eds.), *Information retrieval and mining in distributed environments* (pp. 1–20). Springer. 1, 9, 10

Breusch, T. S., & Pagan, A. R. (1979). A Simple Test for Heteroscedasticity and Random Coefficient Variation. *Econometrica*, *47*(5), 1287–1294. 38, 39

Chao, D. L., Balthrop, J., & Forrest, S. (2005). Adaptive radio: achieving consensus using negative preferences. *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work - GROUP '05*, 120. doi: 10.1145/1099203.1099224 10, 11, 16, 20, 21

Crossen, A., Budzik, J., & Hammond, K. J. (2002). Flytrap: intelligent group music recommendation. *Proceedings of the 7th international conference on Intelligent user interfaces - IUI '02*, 184. doi: 10.1145/502716.502748 10, 11, 16, 18, 20, 21

Felfernig, A., Atas, M., Helic, D., Tran, T. N. T., Stettinger, M., & Samer, R. (2018). Algorithms for Group Recommendation. In A. Felfernig, L. Boratto, M. Stettinger, & M. Tkalčič (Eds.), *Group recommender systems* (pp. 27–58). Springer. 10, 20

Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). *Group Recommender Systems* (A. Felfernig, L. Boratto, M. Stettinger, & M. Tkalčič, Eds.). Springer. 1, 5, 10, 15, 59

Fields, B., Rhodes, C., & Casey, M. (2008). Social playlists and bottleneck measurements: Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values. In *Ismir 2008, 9th international conference on music information retrieval* (pp. 559–564). Philadelphia, PA, USA. Retrieved from `https://www.semanticscholar.org/paper/Social-Playlists-and-Bottleneck-Measurements%3A-Using-Fields-Rhodes/14485e6e745d8e46f4d4e0523d16f180fd974f3f` 6

Forsyth, D. R. (2014). Group Dynamics. *Annu. Rev. Psychol.*, *15*(1 1), 421–446. doi: 10.1146/annurev.ps.15.020164.002225 9

Gärtner, D., Kraft, F., & Schaaf, T. (2007). An adaptive distance measure for similarity based playlist generation. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *1*, 229–232. 6

Hadash, S., Willemsen, M. C., Tintarev, N., Knees, P., & Tkalčič, M. (n.d.). *How to Order a Generated Playlist: The Effects of Tempo and Mood Variations on the Flow ofMusic Track Transitions.* Eindhoven. 5, 6, 19

Hansen, D. L., & Golbeck, J. (2009). Mixing It Up : Recommending Collections of Items. In D. R. Olsen & R. B. Arthur (Eds.), *Chi '09: Proceedings of the sigchi conference on human* *factors in computing systems* (pp. 1217–1226). Boston, MA, USA: ACM. doi: 10.1145/1518701.1518883 5

Jannach, D., & Lerche, L. (2015). Beyond " Hitting the Hits " – Generating Coherent Music Playlist Continuations with the Right Tracks. *Proceedings of the 9th ACM Conference on Recommender Systems - RecSys '15*, 187–194. 5

Jehan, T., & Desroches, D. (2004). *Analyzer Documentation [version 3.2]* (Tech. Rep.). Somerville, MA: The Echo Nest Corporation. Retrieved from `http://docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf` 4, 29

Kahneman, D., Fredrickson, B. L., Schreiber, C. A., Donald, A., & Redelmeier, D. A. (1993). When more pain is preferred to less: Adding a better end. *Psychological Science*, *4*(6), 401–405. 6

Kaminskas, M., & Ricci, F. (2012). Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, *6*(2-3), 89–119. doi: 10.1016/j.cosrev.2012.04.002 3, 4, 57

Knees, P., Pohle, T., Schedl, M., & Widmer, G. (2006). Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, 147. doi: 10.1145/1178677.1178699 6

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, *22*(4-5), 441–504. 6

Konstas, I., Stathopoulos, V., & Jose, J. M. (2009). On social networks and collaborative recommendation. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, 195. doi: 10.1145/1571941.1571977 4

Lamere, P. (2012). *Map of Music Styles.* Retrieved from [19-2-2019]`https://musicmachinery.com/2012/04/22/map-of-music-styles/` 30, 31

Liang, Y., & Willemsen, M. C. (n.d.). *Personalized Recommendations for Music Genre Exploration.* 's-Hertogenbosch. 57

Liebman, E., Saar-Tsechansky, M., & Stone, P. (2015). DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation. In Bordini, Elkind, Weiss, & Yolum (Eds.), *Proceedings of the 14th international conference on autonomous agents and multiagent systems (aamas 2015)* (pp. 591–599). Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems. Retrieved from `http://arxiv.org/abs/1401.1880` 5

Logan, B. (2004). Music recommendation from song sets. In *Ismir'04 - proceedings of the international conference on music information retrieval* (pp. 10–14). Barcelona, Spain. Retrieved from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.2322&rep=rep1&type=pdf` 6

Maillet, F., Eck, D., Desjardins, G., & Lamere, P. (2009). Steerable Playlist Generation by Learning Song Similarity from Radio Station Playlists. In K. Hirata, G. Tzanetakis, & Y. Kazuyoshi (Eds.), *10th international society for music information retrieval conference (ismir 2009)* (pp. 345–350). Kobe, Japan: International Society for Music Information Retrieval. doi: 10.1.1.205.8853 5

Masthoff, J. (2004). Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modelling and User-Adapted Interaction*, *14*(1), 37–85. 12, 15, 57

Masthoff, J. (2015). Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Springer-verlag* (2nd ed., Vol. 54, pp. 217–253). Springer. doi: 10.1007/978-1-4899-7637-6 1, 6, 9, 10, 11, 12, 61

Masthoff, J., & Gatt, A. (2006). In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems. *User Modeling and User-Adapted Interaction*, *16*(3-4), 281–319. 12, 16, 17, 19, 23, 32, 57

McCarthy, J. F., & Anagnost, T. D. (1998). MusicFX: An Arbiter of Group Preferences for Computer Supported Collaborative Workouts. *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW '98*, 363–372. doi: 10.1145/289444.289511 10, 11, 16, 21

Müllensiefen, D., Gingras, B., Stewart, L., & Ji, J. (2013). *Goldsmiths Musical Sophistication Index (Gold-MSI) v1.0: Technical Report and Documentation Revision 0.3* (Tech. Rep.). London: Goldsmiths University of London. Retrieved from `https://www.gold.ac.uk/music-mind-brain/gold-msi/` 34

O'Hara, K., Lipson, M., Jansen, M., Unger, A., Jeffries, H., & Macer, P. (2004). Jukola: Democratic Music Choice in a Public Space. In *Proceedings of the 2004 conference on designing interactive systems processes, practices, methods, and techniques - dis '04* (pp. 145–154). Cambridge, MA, USA: ACM New York, NY, USA. doi: 10.1145/1013115.1013136 10, 11, 21

Pachet, F., Westermann, G., & Laigre, D. (2001). Musical data mining for electronic music distribution. *Proceedings - 1st International Conference on WEB Delivering of Music, WEDELMUSIC 2001*(May 2014), 101–106. 30

Pampalk, E., Pohle, T., & Widmer, G. (2005). Dynamic Playlist Generation Based On Skipping Behavior. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, 634—-637. 6

Piliponyte, A., Ricci, F., & Koschwitz, J. (2013). Sequential music recommendations for groups by balancing user satisfaction. *CEUR Workshop Proceedings*, *997*. 10, 11, 12, 13, 16, 19, 21, 57

Popescu, G., & Pu, P. (2012). What's the best music you have? Designing Music Recommendation for Group Enjoyment in GroupFun. In *Proceedings of the 2012 acm annual conference extended abstracts on human factors in computing systems extended abstracts - chi ea '12* (pp. 1673–1678). Austin, Texas, USA: ACM New York, NY, USA. doi: 10.1145/2212776.2223691 10, 11, 16, 21

Quijano-Sanchez, L., Recio-Garcia, J. A., Diaz-Agudo, B., & Jimenez-Diaz, G. (2013). Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology*, *4*(1), 1–30. doi: 10.1145/2414425.2414433 23, 57

Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook* (2nd ed., Vol. 54; F. Ricci, L. Rokach, & B. Shapira, Eds.). Springer. doi: 10.1007/978-1-4899-7637-6 15

Schedl, M., Knees, P., & Gouyon, F. (2017). New Paths in Music Recommender Systems Research. *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17*, 392–393. doi: 10.1145/3109859.3109934 4

Schedl, M., Zamani, H., Deldjoo, Y., Elahi, M., & Chen, C.-w. (2018). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, *7*(2), 95–116. doi: 10.1007/s13735-018-0154-2 3, 4, 6

Statista. (2019). *Monthly reach of on demand platforms Spotify and Netflix in the Netherlands from 2014 to 2018.* Retrieved from [18-2-2019]`https://www.statista.com/statistics/666260/reach-of-spotify-and-netflix-in-the-netherlands/` 25

Tintarev, N., & Masthoff, J. (2015). Designing and Evaluating Explanations for Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 479–510). Springer. 12

Van Der Zwaag, M. D., Janssen, J. H., & Westerink, J. H. (2013). Directing physiology and mood through music: Validation of an affective music player. *IEEE Transactions on Affective Computing*, *4*(1), 57–68. 4

Wiechert, E. C. E. J. (2018). *The peak-end effect in musical playlist experiences* (Master). Eindhoven University of Technology. 6

Willemsen, M. C., Graus, M. P., & Knijnenburg, B. P. (2016). Understanding the role of latent feature diversification on choice difficulty and satisfaction. *User Modelling and User-Adapted Interaction*, *26*(4), 347–389. 34

Williams, B., Onsman, A., & Brown, T. (1996). Exploratory factor analysis: A five-step guide for novices. *Journal of Emergency Primary Health Care*, *19*(May), 42–50. 34

Yang, L., Bagdasaryan, E., Gruenstein, J., Hsieh, C.-K., & Estrin, D. (2018). OpenRec: A modular framework for extensible and adaptable recommendation algorithms. In *Wsdm '18 proceedings of the eleventh acm international conference on web search and data mining* (pp. 664–672). Marina Del Rey, CA, USA: ACM New York, NY, USA. doi: 10.1145/3159652.3159681 58

Zhao, Q., Willemsen, M. C., Adomavicius, G., Harper, F. M., & Konstan, J. A. (2018). Interpreting User Inaction in Recommender Systems. In *Recsys '18 proceedings of the 12th acm conference on recommender systems* (pp. 40–48). Vancouver, BC: ACM. 57

Acknowledgements

I would like to thank my supervisors Martijn Willemsen and Nava Tintarev for the constant involvement and encouragement I received throughout this thesis process. Additionally, I'd like to thank Martijn for making the recommender lab meetings we had a fun and inspiring moment to look forward to each week.

I would also like to thank Yu and Danielle for their invaluable assistance during my user study. You dedicated your time and effort towards helping me conduct my study and I appreciate this greatly.

Finally, I'd like to thank Masja for all the love, encouragement, strength, and support I received from her during my studies.

Appendix A

Proof of Equation 9

Let us start the proof with a couple of definitions:

$$A_i : \textit{user likes track } i \tag{16}$$

$$B_k : \textit{user likes cluster } k \tag{17}$$

$$C_{ij} : \textit{track } i \textit{ belongs to cluster } k \tag{18}$$

We are interested in finding the probability that a user likes a cluster given that we know which tracks the user likes and that we know the probability of each track belonging to a cluster.

$$p(B_k|\boldsymbol{A}, \boldsymbol{C}) \tag{19}$$

Using marginalization, we find

$$p(B_k|A_i) = \sum_{j=1}^{N_c} p(B_k|A_i, C_{ij})p(C_{ij}) \tag{20}$$

Then, we define our first assumption: the probabilities that a user likes a cluster given that a user likes a track that does or does not belong to that cluster are constant.

$$p(B_k|A_i, C_{ij}) = \begin{cases} \pi & \text{if } j = k \\ \nu & \text{otherwise} \end{cases} \tag{21}$$

Using assumption 1, Equation 20, and the rule of total probability we find

$$p(B_k|A_i) = \nu + (\pi - \nu)p(C_{ik}) \tag{22}$$

Applying marginalization again leads to

$$p(B_k) = \sum_{i=1}^{N_t} (\nu + (\pi - \nu)p(C_{ik}))p(A_i) \tag{23}$$

We then define the second assumption: a user likes all his/her top tracks. Therefore, we limit ourselves to this set of tracks.

$$p(A_i) = \begin{cases} 1 & \text{if } A_i \in \text{top tracks} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Applying the second assumption to Equation 23 gives

$$p(B_k) = N_t\nu + (\pi - \nu)\sum_{i=1}^{N_t} p(C_{ik}) \tag{25}$$

Using the rule of total probability, we find

$$\sum_{i=1}^{N_c} p(B_k) = 1 \tag{26}$$

$$N_tN_c\nu + (\pi - \nu)N_t = 1 \tag{27}$$

$$\pi = \frac{1}{N_t} + \nu(1 - N_c) \tag{28}$$

Substitution leads to the final equation

$$p(B_k) = N_t\nu + (\frac{1}{N_t} - N_c\nu)\sum_{i=1}^{N_t} p(C_{ik}) \tag{29}$$

And for the special case where $\nu = 0$:

$$p(B_k) = \frac{1}{N_t}\sum_{i=1}^{N_t} p(C_{ik}) \tag{30}$$

Appendix B

Interview Plan

session ______, condition order _________

Study Procedure Form

Preparation

- Prepare all documents
  - 3 consent forms
  - 9 condition forms
  - 3 end forms
  - 3 demographic forms
  - 1 study procedure form
  - Money receive form with 3 empty slots
- Sign metadata (generate random condition order)
- Test audio recorder

General

0. For the duration of the study, does one participant notably take the lead?

Lead participant? _____

Welcome Phase ~ 5 min

1. Hand out consent forms.
2. Meanwhile, assign local participant IDs.

1	
2	
3	

3. Introduce the study to the participants. Indicate that thinking out loud is encouraged.
4. Start audio recording

Login Phase ~ 5 min

5. Ask participants to login to the system. Do not give details on how this is done unless specifically asked or participants are uncertain.

Details asked? Yes / no

____

____

____

____

____

Condition __ ~ 10 min (fill in for each condition)

6. Ask participants to interact with the system, listen to the music, and discuss how you like each track with each other.

____

____

____

____

____

7. Ask participants to fill in private form. Make sure to fill in the metadata afterwards.

8. Ask the following: "Did the recommender overly represent the musical interests of certain group members over those of others? If yes, please indicate in what order your musical tastes were represented."

Yes / no

most	
neutral	
least	

Note. In case of equal representation, but multiple names / participant ids in the same box

9. Ask the following: "If you would grade this recommender based on how it is able to play music that all group members enjoy listening to, what would it be? Please use a scale from 1 to 10."

Grade: ____

____

____

____

General Discussion ~ 5 min

1. Would you use this recommender system as a group for example while working on a project?

2. In what context or situation would you find this recommender to be most useful?

3. Would you use this recommender system with your friends to discover each other’s music tastes and preferences?

4. What do you like most about this recommender? What do you miss, what would you improve?

Like most:

Miss:

Improve:

Finalization

1. Hand ending form to each participant. Mark metadata afterwards
2. Hand demographics form. Mark metadata afterwards
3. Let participants sign the money receive form
4. Hand out money to participants
5. Debrief participants

After participants leave

1. End audio recording
2. Bundle all files