Dissertation

Multi-Context-Aware Recommender Systems: A Study on Music Recommendation

Martin Pichl

submitted to the Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements
for the degree of "Doctor of Philosophy (PhD)"

Advisor: Univ.-Prof. Dr. Günther Specht

Innsbruck, 2018

Abstract

In the last decade, recommender systems became an integral part of today's digital world. They help us to deal with today's information flood and to find information, services as well as products as books, movies or music we like. In particular, recommender systems are important for digital goods as the number of digital products increased dramatically in the last decade. This is, as those products have low production costs per unit accompanied by virtually no inventory- and transportation costs. Besides the challenge of finding items a user likes in this sheer number of available items, there is the challenge to consider the current context of the consumption or rather the user, i.e., the current time, the current activity or the current emotional state of a user. Today, the recommender systems and music information retrieval communities agree that context is inevitable to provide good personalized recommendations. Hence, recommender systems have to jointly incorporate two important aspects in order to be successful: recommendations must be favored by the user and fit the current situation. Relying on matrix factorization, regression and clustering techniques, in this thesis, we present a multi-context-aware music recommender system capable of exploiting different contextual facets of today's music consumption. In particular, we leverage the current listening context, i.e., the current situation during consuming music, the acoustical context and the cultural embedding of a user. We observe that especially the interaction between several contexts improves the performance of our recommender system. Our proposed approach models which type of music is preferred by which type of user. In several offline experiments, we show that modeling these interactions result in a superior performance of our user model in terms of precision and recall compared to all baseline approaches.

Zusammenfassung

In den letzten Jahren sind Empfehlungssysteme zu einem integralen Bestandteil unseres digitalen Lebens geworden. Sie helfen uns, mit der täglichen Informationsflut umzugehen und liefern personalisierte Vorschläge für Informationen, Dienstleistungen und zu Produkten wie Büchern, Filme und Musik. Speziell im Bereich der digitalen Güter sind diese Systeme nicht mehr wegzudenken, da die verfügbare Menge an digitalen Produkten in den letzten Jahren dramatisch angestiegen ist. Gründe dafür sind die niedrigen Produktionskosten pro Stück sowie die geringen Lager- und Distributionskosten. Neben der Herausforderung, aus einer unüberschaubaren Menge an Produkten jene Produkte zu finden, die einem Kunden gefallen könnten, gilt es auch, den Kontext des Benutzers bzw. des Konsums zu berücksichtigen. Die Empfehlungssysteme- sowie die Music Information Retrieval Communities sind sich einig, dass Empfehlungssysteme zwei Aspekte berücksichtigen müssen, um erfolgreich zu sein: Das Produkt muss dem Kunden gefallen und es muss im richtigen Moment vorgeschlagen werden. Basierend auf Matrix Faktorisierung, Regressionsanalyse und Clustering Methoden präsentieren wir in dieser Arbeit ein Multi-Kontext-Musikempfehlungssystem, welches es ermöglicht, den Einfluss von verschiedenen Kontexten auf den Kunden bzw. auf den Konsum des Benutzers zu modellieren. Im Speziellen berücksichtigen wir die Situation, in der Musik gehört wird, den akustischen Kontext der Musik basierend auf Playlisten, sowie den Musik-kulturellen Hintergrund eines Benutzers. Mit dem in dieser Arbeit entwickelten Benutzermodell können wir modellieren, dass bestimmte Benutzer in bestimmten Situationen bestimmte Musik hören und zeigen damit, dass im Speziellen die Interaktionseffekte zwischen diesen Kontexten einen großen Einfluss auf die Qualität der Empfehlungen haben. In Offline-Experimenten zeigen wir, dass das Benutzermodell mit mindestens dem Interaktionseffekt aus dem situativen und dem akustischen Kontext bessere Precison- und Recall-Werte liefert als alle Baseline-Modelle.

Acknowledgements

Numerous people had a large contribution to this work. Above all, my wife Bettina, patently bearing my student lifestyle since my Bachelor studies and marrying me during my PhD. Furthermore, I want to thank my parents for supporting me without any doubt although I sometimes have the feeling that they do not fully understand what I am working on. I also want to thank all my friends supporting my interesting journey throughout the last four years.

Besides family and friends, my working colleagues had a really strong contribution to this work. First and foremost Eva, thank you for bringing me to DBIS and writing *all* papers with me. Doubtless, without your support, we would not have published that well. Besides my main co-author Eva, I want to thank Wolfi for the warm welcome (although I was not a computer scientist and wearing shirts) along with the conversations about recommender systems and personalization while drinking a beer. Robert, we had numerous of nice Saturdays and Sundays in our office with conversations beyond computer science. I will miss them. Benjamin, thank you for all our vivid Klingon conversations, discussions about classifiers and most importantly reminding me to organize the "DBIS Stammtisch". Now it is your job ;). Furthermore, I want to thank all DBIS members and DBIS alumni Michael, Niko, Doris, Dominic, Hansi, Sylvia, Michael, Matthias, Rainer making the last 4 years joyful, we are a great team.

Finally, I want to thank Prof. Dr. Günther Specht for giving me the opportunity working for DBIS, supporting me with strategic input, supporting our publications along with the necessary travels and for supervising my thesis.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht. Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister-/Master-/Diplomarbeit/Dissertation eingereicht.

Datum	Martin Pichl

Contents

CHAPTER 1

Introduction

1.1 Motivation

Recently, a major change in the way people consume music has taken place: people increasingly switch from listening to their private and mostly limited music collections to consume music provided by music streaming platforms providing several millions of tracks [65]. Along with that, driven by the heavy usage of social media platforms and micro-blogging services, music streaming platforms offer facilities allowing their users to share what they are currently listening to. Using these facilities, users share (either manually or automatically) the tracks they are currently listening to via social media platforms as Facebook[1] or micro-blogging platforms as Twitter[2].

Generally, we observe that music collections are that large and diverse such that they cannot be browsed manually anymore in order to find favored music. Hence, users have to rely on music discovery facilities as recommender systems. Furthermore, we witness a worldwide community of users publicly sharing their

[1] `https://www.facebook.com`

[2] `https://twitter.com`

music listening habits. Based on these trends, we derive a set of four facets that make music information retrieval (MIR) and connected to this, music recommendation (MR) particularly interesting in the late 2010s. Next, after giving an overview of all four facets, we elaborate on them in more detail in the remainder of this section.

Facet 1: Data A rise in publicly available data about music consumption via the social web that can be leveraged for music information retrieval research and music recommender systems research.

Facet 2: Millions of Tracks The availability of millions of tracks via music streaming platforms makes music recommendation particularly interesting, as users need support in finding music they like.

Facet 3: Every-time Access Every-time access to music collections on streaming platforms using mobile devices stresses the importance of music recommender systems that consider the current context of music consumption (i.e., whether music is consumed at home, at work or in gym).

Facet 4: Worldwide Music consumption is a culture independent phenomenon as regardless of culture or society, most people enjoy listening to music [109]. For personalized music recommendations, the music-cultural embedding of a user needs to be considered.

The first facet, an increased availability of publicly available data, is driven by the distribution of music streaming and the simultaneous rise of social media platforms and micro-blogging services. An increasing amount of people publicly share to what they are listening to at the moment. This allows MIR researchers to observe the music listening behavior of thousands of users worldwide, for instance by crawling for #nowplaying tweets via the micro-blogging platform Twitter [128, 110, 129]. Besides a precise timestamp, often these tweets contain further valuable meta-information, i.e., the device or software the tweet was sent with and GPS coordinates that reveal the location from which the tweet was sent. Besides that, the tweet itself often contains a link to the track on a music streaming platform. In this work, we discuss different recently introduced MIR datasets (including our own) that are based on #nowplaying tweets and data crawled from the streaming platforms itself (cf. Chapter 4). Those datasets allow unprecedented user studies of music consumption in terms of scale. This is why those datasets are heavily used for music recommender systems and music information retrieval research.

The second facet is based on the fact that music streaming platforms do not have inventory costs, as traditional brick and mortar stores have. Moreover,

for digital goods as music, there are virtually no distribution costs and the same track can be sold to a worldwide market without any modifications. Hence, the production costs per unit are low. This is why nowadays an increased amount and an increased variety of music is available [6, 26]. According to Anderson [6], this yields a distribution where only a limited number of popular products are known by potential customers and that there are many more unknown (-niche) products forming a long tail. In Section 4.5, we show empirically that the distribution of Spotify tracks follows this long-tailed distribution of digital goods. Thus, the task of a music recommender system is to find tracks in the long tail a user likes and simultaneously not aware of it is (yet). This task is also known as the "find good items task" [46].

The third facet is strongly related to the second facet and based on the everywhere and every time access to music, which makes the "find good items task" [46] even more challenging. As already highlighted, due to the sheer number of tracks, streaming platforms heavily rely on recommender systems. Those systems are intended to help users navigate through the provided collections containing millions of tracks and hence, to assist users in discovering music they like. Along with that, we observe that users access music streaming platforms using a diverse set of mobile and stationary devices [65]. Moreover, we know that whether or not a user likes a recommended track heavily depends on the user's current situation we refer to as the current context. Previous research has shown that information about the context of a user (i.e., time, location, occasion, emotional state) is essential for providing personalized music recommendations [59, 64]. Naturally, people listen to different music during different activities. Accordingly, they tend to organize tracks in their music collections by the intended use (e.g. working or exercising) [54]. Cunningham et al. [30] found that people create playlists that are intended for certain activities. This is why we argue that besides the *personalization aspect*, namely finding a track a certain user likes, there is the *suitability aspect*, namely presenting a track a user likes and that simultaneously fits to the current context. In the related work in Chapter 3, where we discuss recent trends and state of the art research, we also state major issues and researcher gaps related to context-aware music recommendation. We overcome the main issues by our proposed ensemble latent feature computation (ELFC) for music recommendation (MR) approach, a multi-context aware music recommender system allowing to incorporate the interaction effects of different contexts even under data sparsity. Besides the recommendation approach and the novel user model, we propose a set of data mining methods for gathering information about the current listening context and the acoustical context of a user in order to overcome a lack of data.

Finally, there is a societal facet. Regardless of culture or society, most people enjoy listening to music [109]. Hence, prior works focused on discovering

regional music listening patterns [113, 114] based on countries and continents. However, political boarders of countries do not necessarily reflect musical- or cultural boarders and continents are a too coarse entity [114]. In this work, we pick up a call for a cultural similarity and propose a method to incorporate the music-cultural embedding of a user in our multi-context aware music recommender system.

1.2 Guiding Research Questions

Based on the four facets that make music recommendation particularly interesting (presented in the previous section), we set up five main research questions (RQs) guiding this work. In particular, our RQs aim to cover open issues and research gaps in the field of music information retrieval.

RQ1 How do people organize and consume music in the music streaming era?

RQ2 How can we model mathematically the music listening behavior of music streaming users?

RQ3 How can we implement the prior model in a multi-context-aware music recommender system?

RQ4 What is the impact of different contextual dimensions on the recommendation accuracy?

RQ5 Are there cultural music listening patterns or do music streaming platforms homogenize the worldwide music consumption?

Firstly, as depicted in RQ1, we are interested in the music consumption behavior of music streaming users. Particularly, we are interested in whether users listen to different types of music and how users organize their music to find (again) music they enjoy listening to inside the huge music collections provided by streaming platforms. Secondly, we are interested in how the findings of RQ1 can be leveraged in a music recommender system, as stated in RQ2 and RQ3. In particular, we are interested in modeling which type of music is listened by which user in which context. Connected to this, and reflected in RQ4, we are interested in which contexts influence the music consumption of streaming users most. Finally, we are interested in the general applicability of our user models. In particular, we are interested in whether there exist cultural music listening patterns or whether music consumption via music streaming platforms is homogeneous and geographically or rather culturally independent. This is reflected in RQ5.

1.3 Methodologies

Besides a profound literature review in the beginning of this work, different data mining and machine learning methodologies have been facilitated to answer the research questions set up during the literature review. We list the most important techniques below and give a short description of their application in the following.

M1 Matrix Factorization

M2 Dimensionality Reduction / Latent Feature Computation

M3 Clustering

M4 Classification

M5 Model-based Collaborative Filtering

First of all, we heavily leverage matrix factorization techniques as singular value decomposition (SVD) and Cholesky decomposition (CD) as they are known to work well [63, 95] to represent user-item and user-item-context interactions (i.e., listening to a track at home) in a latent feature space. Latent features allow to incorporate unobserved features, deal with data sparsity and furthermore allow us to compute recommendations more efficiently. Related to matrix factorization, we moreover use dimension reduction methods as principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) for (i) interpreting and visualizing our data along with the results and (ii) to perform clustering on the reduced matrix to achieve better results. The latter is a common approach for datasets with a high number of features and sparsity similar to matrix factorization.

Secondly, we rely on different clustering techniques as k-means, density-based spatial clustering of applications with noise (DBSCAN) and spectral clustering that allow us to group contexts, users, single tracks as well as whole playlists (unordered sets of tracks). Depending on the entity we want to group, we perform the cluster computation on a different feature set. For playlists and tracks, we mainly rely on acoustical features. In contrast, to group users, we rely on acoustical features as well as cultural features and socio-economic variables. For the sake of simplicity, we refer to these clusters as types or groups. Hence, we refer to track clusters as “music types”, context clusters as “context types” and to user clusters as “user groups”.

Finally, we use the derived groups of users along with the derived types contexts, playlists and tracks as input to several classifier-based recommendation

models. Grouping (latent) features to a single feature using clustering allows us to efficiently compute the interaction effects between different contexts. Hence, we enrich pure collaborative filtering-based (CF) approaches with our contextual clusters and their interactions. This allows us to model which type of user listens to which type(s) of music in which type of context.

1.4 Contribution and Published Work

Throughout the course of this thesis, we contributed to several peer-reviewed workshops and published sub-parts of this work in peer-reviewed conferences and journals (cf. listing in the remainder of this section). We give a brief overview of our contribution to the field of music information retrieval with relation to our research questions in the following: With respect to RQ1, we find, that users enjoy listening to various different types of music and along with that could compute typical types of user-curated playlists [88]. In this and prior works [88, 89, 91] we consider music as different if their audio characteristics differ. Along with that, we find that playlist names can be leveraged for extracting contextual information, for instance the current activity or the occasion [87]. In a logical next step, we have been interested in how both findings can be leveraged in a music recommender system (cf. RQ2 and RQ3) [87, 89]. For this, we focus on how the interaction effect between the type of music and a situational context can be modeled. In particular, we are interested in modeling which type of music is listened by which user in a certain situation. We find that factorization machines as presented in Section 2.3.5 allow to model the influence of the different contexts itself along with the interaction effects between different contexts. To answer RQ4 and hence, to estimate the impact of different contextual dimensions as well as to compare different models, we rely on our implementation of a music recommender system prototype along with it's evaluation framework. This framework is capable of benchmarking different user models, as it easily allows to change the computational kernel and hence, the recommendation strategy. Using different measures for evaluating recommender systems, we benchmark the set of different contextual recommendation models proposed in this work. To contextualize the performance of our proposed models, we furthermore benchmark a set of baseline approaches. Finally, two answer RQ5, we analyze cultural music listening patterns on music streaming platforms. Based on this analysis, we invent a novel user model additionally incorporating the music-cultural-context. We find that when the music-cultural-context is considered in the user model, it further improves the recommendation accuracy.

Conference and Journal Papers (peer-reviewed)

- M. Pichl, E. Zangerle, G. Specht and M. Schedl: Mining Culture-Specific Music Listening Behavior from Social Media Data. In Proceedings of the 19th IEEE Symposium on Multimedia (ISM 2017). IEEE, 2017.

- M. Pichl, E. Zangerle and G. Specht: Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In Proceedings of the 7th ACM International Conference on Multimedia Retrieval (ICMR 2017), pages 201-208. ACM, 2017.

- M. Pichl, E. Zangerle and G. Specht: Understanding User-Curated Playlists on Spotify: A Machine Learning Approach. International Journal of Multimedia Data Engineering and Management (IJMDEM). 8(4), 2017.

- M. Pichl, E. Zangerle and G. Specht: Understanding Playlist Creation on Music Streaming Platforms. In Proceedings of the 18th IEEE Symposium on Multimedia (ISM 2016). IEEE, 2016.

- E. Zangerle, M. Pichl, B. Hupfauf and G. Specht: Can Microblogs Predict Music Charts? An Analysis of the Relationship Between #Nowplaying Tweets and Music Charts. In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016), pages 365-371. ISMIR, 2016.

Workshop Contributions (peer-reviewed)

- M. Pichl, E. Zangerle and G. Specht: Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?. In Proceedings of the 15th IEEE International Conference on Data Mining Workshops (ICDM 2015), pages 1360-1365. IEEE, 2015.

- M. Pichl, E. Zangerle and G. Specht: #nowplaying on #Spotify: Leveraging Spotify Information on Twitter for Artist Recommendations. In Proceedings of the 2nd International Workshop on Mining the Social Web in conjunction with the 15th International Conference on Web Engineering (ICWE 2015), pages 163-174. Springer, 2015.

- M. Pichl, E. Zangerle and G. Specht: Combining Spotify and Twitter Data for Generating a Recent and Public Dataset for Music Recommendation. In Proceedings of the 26nd Workshop Grundlagen von Datenbanken (GvDB 2014), Ritten, Italy, vol. 1313, pages 35-40. CEUR-WS.org, 2014.

- E. Zangerle, M. Pichl, W. Gassler and G. Specht: #nowplaying Music Dataset: Extracting Listening Behavior from Twitter. In Proceedings of the 1st ACM International Workshop on Internet-Scale Multimedia Management (WISMM 2014), pages 21-26. ACM, 2014.

Other Contributions during PhD Studies, not covered in this thesis (peer-reviewed)

- B. Murauer, M. Mayerl, M. Tschuggnall, E. Zangerle, M. Pichl and G. Specht: Hierarchical Multilabel Classification and Voting for Genre Classification. In Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017), CEUR-WS.org, 2017.

- E. Zangerle, W. Gassler, M. Pichl, S. Steinhauser and G. Specht: An Empirical Evaluation of Property Recommender Systems for Wikidata and Collaborative Knowledge Bases. In Proceedings of the 12th International Symposium on Open Collaboration (OpenSym 2016), pages 18:1-18:8. ACM, 2016.

1.5 Thesis Outline

As recommender systems are an integral part of our research, we present different recommendation approaches that are related to our approach in Chapter 2. Next, in Chapter 3, we introduce the reader to the field of music information retrieval and music recommendation, the domain we focus our research on. In this chapter we also elaborate on existing research gaps we cover. We cover those research gaps using methods developed for our ensemble latent feature computation for music recommendation (ELFC-MR) approach we present in Chapter 5, after presenting famous datasets for music information retrieval research including ours in Chapter 4. In the subsequent Chapters 6, 7 and 8 we present the details of the three major components of our approach: the *situational context* component, the *music characteristics* component as well as the *music-cultural context* component. Chapter 9 concludes this work and discusses future work.

CHAPTER 2

Foundations: Recommender Systems

2.1 Introduction

As outlined in the introduction, the importance of recommender systems grew substantially in recent years. One reason for this is that the web emerged as a new distribution channel. This distribution channel enables companies to sell their products to a worldwide market, which drove the development of that the number of products, as well as the variety of products, increased dramatically. This development is especially valid for digital products, i.e., music or apps, where the same products can be sold worldwide without major modifications. In 2006, Anderson [6] coined the term *long tail* for that new product availability: Especially for digital products it holds, that substantially more niche products are available. This is, as no inventories are necessary for those goods. Along with that, without or with low transportation and distribution costs, products can easily be sold worldwide. Hence, customer groups demanding niche products that have not been served in the past, as they have been too small due to geographic restrictions are served nowadays, as if they aggregated worldwide, they are not small anymore. This drove the development of

a product distribution with a small number of popular products forming the head of the distribution and much more niche products forming a long tail. An example of a long-tailed distribution is given in Figure 2.1. This is an empirical analysis of the Spotify playlist dataset, which we introduce in Section 4.5, containing more than 1,000 users. The popularities of the tracks are visualized by plotting how often a track is found in the music libraries of the users in the dataset. The number of tracks is plotted on the logarithmically scaled x-axis and the corresponding number of users on the y-axis. We observe, that 10 tracks can be found in not more than 54 music libraries, 12,711 tracks can be found in not more than 5 libraries and 510,878 tracks can be found in only one music library. Hence, we argue that the distribution of the tracks on the music streaming platform Spotify follows a long-tailed distribution: There are rarely popular tracks with high play counts forming the head of the distribution and much more unpopular tracks with low play counts forming a long tail.

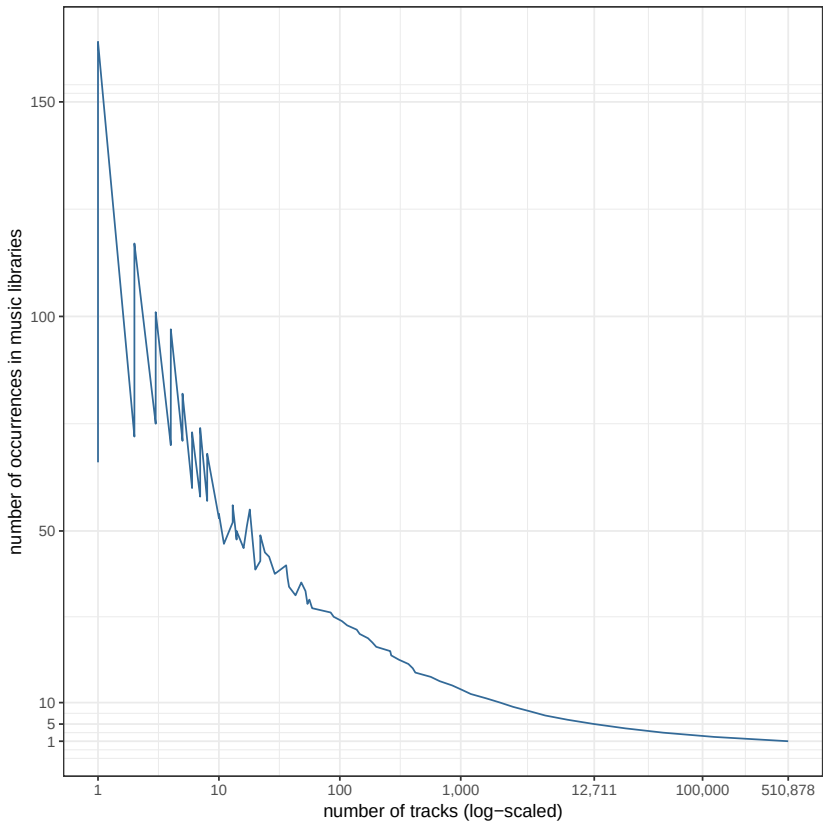

Figure 2.1: Long-tailed Distribution of Musical Tracks

This new distribution of the products yields a challenge: customers who are willing to purchase these products in the long tail are often not aware of their existence [6]. This is one of the reasons why nearly every large e-commerce platform as Amazon[1] or Zalando[2] and streaming platforms as Spotify[3] or Netflix[4] run recommender systems, which amongst others, aim at suggesting new products from the long tail to the customers. These recommender systems are based on profound customer analytics and leverage techniques known from the field of information retrieval and machine learning to analyze customers mainly by their previous purchases of goods or their previous interactions with digital products as apps, music or videos.

Summing up, the change from a market of popular "one fits all" products to a market offering a vast amount of highly diverse products drove the development that recommender systems have been well studied in the past decade. This chapter is intended to give an overview about published work in the field of recommender systems. Firstly, we classify and introduce different recommendation models, before the focus is shifted to collaborative filtering-based recommender systems. We focus on this type of recommender system as the recommender system proposed in this work is based on this technique. To be precise, our proposed recommender system is a context-aware model-based collaborative filtering approach using factorization machines. Hence, we devote the whole Section 2.3.5 to this technique. We are aware of the fact that over the last decades several other recommendation models have been proposed, for instance, knowledge-based recommender systems or recommender system based on neural networks. However, these types of systems are out of the scope of this thesis, as they are hardly related to our proposed music recommender system.

2.2 Recommender Systems: A Definition

Firstly, we start with a simple definition of a recommender system: A recommender system is a system able to suggest items to users [98]. On a more abstract level, a recommender system is a system that suggests content a user is interested in out of an enormous set of choices [98] and hence, is a system to overcome the information overflow. For this task, a recommender system aims at predicting which is the most useful item to a user and states a short list of n recommendations. Based on this definition, we derive two main tasks:

[1] `https://www.amazon.com`, last visited November 26, 2017

[2] `https://www.zalando.com`, last visited November 26, 2017

[3] `https://www.spotify.com`, last visited November 26, 2017

[4] `https://www.netflix.com`, last visited November 26, 2017

(i) the rating prediction task and the (ii) top-n recommendations task also referred to as the "find good items task" [46] or retrieval task. The second task is based on the first task, as a recommender system orders the list of recommendation candidates by the predicted rating and respectively, by the perceived usefulness of a user towards an item. Finally, this ordered list is cut off at n. This rating prediction task is depicted in Equation 2.1, where f_R is a utility function assigning predicted ratings $\hat{r}_{u,i}$ to $<user, item>$-combinations. The challenge is to correctly sort an enormous set of items according to a user's preference.

$$f_R = User \times Item \rightarrow Rating \tag{2.1}$$

However, what makes computing recommendations even more challenging, is that besides the recommended items should have a high utility to the user (cf. Equation 2.1), they simultaneously should be recommended at the right moment. Hence, the recommendations must be personalized and should fit to the current situation, also referred to as the current context of a user. For the latter, we extend the initial problem formulation for context agnostic recommender systems in Equation 2.1 to the context-aware problem formulation depicted in Equation 2.2.

$$f_R = User \times Item \times Context \rightarrow Rating \tag{2.2}$$

In the subsequent sections, we elaborate on different approaches for estimating utility function f_R. We start with giving a brief overview of different filtering approaches.

2.3 Recommender Systems: An Overview

As shown in the overview in Figure 2.2 in the last decade, several recommendation algorithms have been developed in academia and industry. Adomavicius and Tuzhilin [2] classify recommendation approaches to be (i) content-based (CB), also referred to as content-based filtering, or (ii) collaborative filtering (CF)-based. Content-based recommendation models focus on the item characteristics to find similar items, whereas CF-based models exploit user-item interactions to find similar users and derive recommendations from these user similarities. Such user-item interactions are dependent on the domain and might be listening to a track, playing a movie or buying an item. Although we classify recommender systems in those categories, also hybrid recommender systems have been proposed [115, 2]. These approaches combine both methods

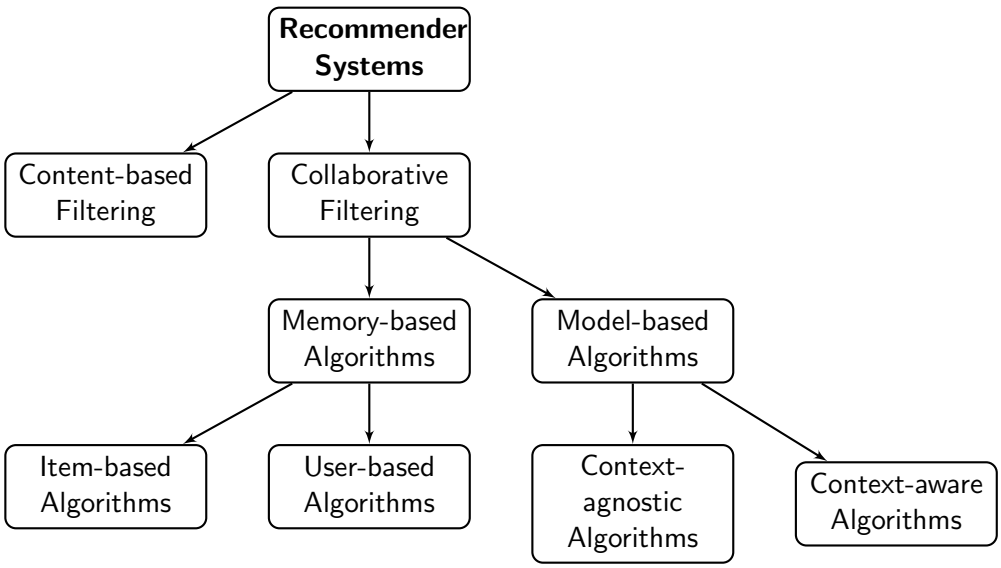

Figure 2.2: Overview of the Discussed Recommendation Approaches

in order to overcome the disadvantages of a single approach [23]. One of the most popular problems in the field of recommender systems is the cold start problem: In case of CF, this is, that for a new user, no or only a very short user-profile is available. Hence, no meaningful similarity to other users can be estimated to compute recommendations. This is known as the "new user" problem. Analogously, there is the "new item" problem: If none of the users has rated a (new) item yet, CF also fails. For the first problem, a hybrid approach leveraging demographic data (if available) to find similar users can be beneficial. For the second problem, a hybrid with content-based information to find similar items to the new item could be beneficial [115].

Moreover, CF-based recommender systems are classified into memory- and model-based types [21]. For the latter, we differentiate between context-agnostic and context-aware recommendation algorithms. Context-aware recommendation algorithms are relatively new approaches, as recently, data became available for investigating a user's current context [3]. We refer to any additional information during the user-item interaction as context. For instance, this can be a certain point in time, a location or a specific mood of a user during the consumption [3]. The music recommender system presented in this work is a context-aware model-based CF approach. This is why we set a focus on CF-based approaches in this chapter and go into the details of context-aware music recommendation in Section 2.3.4.

In the remainder of this section, we introduce the reader to the presented recommendation models in more detail. We start with content-based recommender systems in the following section.

2.3.1 Content-based Algorithms

As outlined in the introductory section of this chapter, content-based (CB) recommender systems focus on item characteristics to find similar items. I.e., these systems recommend items that are similar the items a user already interacted with. Hence, they are also called content-based filtering approaches, as they filter items based on previous user-item interactions. They have their roots in the field of information retrieval [10, 103, 26] and initially focused on recommending items containing text, for instance news articles [121], websites or UseNet messages [2]. For this recommendation task, namely to find the top-n most similar items to a given item, the term frequency-inverse document frequency (tf-idf) measure is applied to the text corpora to compute document vectors. This representation of documents is called the vector space model (VSM). In a next step, similar items are found and ranked by applying a similarity measure as the cosine similarity to the tf-idf vectors in the VSM [103]. In this work, we use a similar approach to find similar playlists, as part of our music recommender system described in Chapter 6. Hence, we present the details of this approach in the following.

The $tf-idf$ measure is depicted Equation 2.3, where we denote tf as the term frequency which measures how often a term t occurs in a certain document d. This document d is part of the whole document corpus D of length $|D|$. The tf-measure is multiplied with the inverse document frequency idf and hence, with the number of documents in the corpus D divided by the number of documents containing the term t ($|\{d \in D : t \in d\}|$). This function yields a result, where the more important a term t is, the higher the $tf - idf_{t,d}$ is. A term t is important if it (i) appears often in the document d and simultaneously (ii) is rare in the corpus D (low df).

$$tf - idf_{t,d} = tf \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{2.3}$$

Besides weighted term counts for text retrieval, a content-based recommender system is usually capable of utilizing any feature vector for computing similarities. In the field of music recommender systems, this feature vector could consist of the audio characteristics of tracks, for example *beats per minute* or *loudness*. To these feature vectors, depending on the domain and content, different similarity measures are applied. For real-valued vectors, the cosine

similarity [103] or correlation-based measures are the most popular similarity measures [2]. For set-based comparisons, i.e., a bag of words describing a document, Jaccard- or the Dice similarity are suitable measures [78, 79]. If the attributes in the feature vectors are orthogonal, the Euclidean or Manhattan distance can be leveraged [26].

As the characteristics of an item are always known in advance to the system, in contrast to the CF-based recommender systems we present next, CB recommender systems do not suffer from the new item cold start problem. Nevertheless, they suffer from the new user problem: without any prior user-item interactions, the system cannot find new items similar to the items a user interacted with. Furthermore, CB models do not take any *personalization* into account, as CF does (described in the next section).

2.3.2 Collaborative Filtering-based Algorithms

The first collaborative filtering-based (CF) recommender system is called Tapestry [40] and was developed by Xerox in their Research Center in Palo Alto. Tapestry was intended to filter any stream of incoming documents, however, was implemented to assist users in managing their mailbox due to an increased mail traffic caused by the dissemination of new groups. Good et al. [40] invented the term and the concept of collaborative filtering, but in fact, their system was a hybrid as it additionally supported content-based filtering (CB). Tapestry paved the way for two collaborative filtering approaches: memory-based algorithms as well as model-based algorithms. Independent of the type, CF-based algorithms have in common that they exploit a user-item matrix R that holds the prior ratings of users u towards items i for the recommendation computation. An example of such a rating matrix R is given in Equation 2.4. It consists of M rows (equal to the number of users) and N columns (corresponding to the number of items). The elements $r_{u,i}$ of the matrix correspond to the rating a user u has assigned to item i. Based on this user-item matrix R, the utility- or rating prediction function f_R, which assigns predicted ratings $\hat{r}_{u,i}$ to $<user, item>$-combinations, is learned.

$$\begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} i_1 & i_2 & \cdots & i_n \end{array} \\ \begin{pmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ r_{2,1} & r_{2,2} & \dots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \dots & r_{m,n} \end{pmatrix} \end{array} \tag{2.4}$$

Besides explicit ratings, as in the famous Movie Lens Dataset[5] where users rated movies with 1 to 5 stars and hence $1 \leq r \leq 5$, often only implicit feedback is available. In this case, the ratings are Boolean, as a recommender system only knows if a user interacted with an item or not. Depending on the domain, such a user-item interaction could be a watching a movie, playing a song or buying an item. In such a setting, an interaction can be encoded as $r_{u,i} = 1$ and a non-interaction as $r_{u,i} = 0$. Beginning with the class of memory based algorithms, we introduce the reader to memory-based and model-based algorithms exploiting the user-item matrix for item recommendations.

Memory-based Algorithms

Memory-based algorithms leverage the whole user-item matrix for estimating rating predictions [21]. This approach can be subdivided into (i) item-based approaches and (ii) user-based approaches. In contrast to the latter, where the similarity between users is calculated, main idea of the first approach is to calculate the similarity between items in the recommendation matrix and recommend similar items to the items a user interacted with. In its trivial form, as introduced by Sarwar 2001 [104], the similarity between each of the items is calculated. For this task, a cosine- [104], correlation- [97] or a set-based similarity measure is applied along the columns of the rating matrix R [78, 79]. Whereas the first two measures are suitable for continuous values, set-based similarity measures are especially suitable for Boolean ratings. Considering our matrix in Equation 2.4, the similarity measure is computed along the columns of the matrix. I.e., we can estimate the similarity s between two items i and j by computing the cosine similarity across the vectors $\vec{i}$ and $\vec{j}$ as depicted in Equation 2.5. Hence, the similarity is computed across all ratings of any user towards the items, as the vectors $\vec{i}$ and $\vec{j}$ correspond to the columns i and j in the rating matrix R.

$$s_{i,j} = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \|\vec{j}\|} \tag{2.5}$$

Alternatively, recommender systems compute the Pearson similarity as depicted in Equation 2.6 among all co-rated items. We denote $U_{i,j}$ as the set of users who rated both, the item i and j, cov as the covariance between two vectors, σ as the standard deviation as well as $\bar{r}$ as the mean rating computed among all ratings towards a certain item. In contrast to the cosine similarity, the Pearson similarity incorporates the average user rating and hence, reflects the different rating styles of the users.

[5] `https://grouplens.org/datasets/movielens/`

$$s_{i,j} = \frac{cov(i,j)}{\sigma_i, \sigma_j} = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_j)}} \tag{2.6}$$

As this computation scales quadratically with the number of items, it can be computationally intensive. However, in contrast to user-based CF, the similarities can be pre-computed and persisted. This is, as the number of similarities to compute only grows with the number of items. User-based CF, in contrast, grows with the number items and with the number of users [23].

After this first item similarity commutation step, in a second step, the predicted rating $\hat{r}$ for a given user u and an item i is computed. Common methods utilize the weighted sum and linear regression. The idea behind the first approach is to predicted a rating for a certain item $\hat{r}_i$ by using the weighted average over all rated items of the user that are similar. The weights for this average are the corresponding item similarities $s_{i,j}$ as depicted in Equation 2.7, where we denote $S_{i,u}$ as the set of all items i a user u rated. The recommender system computes the predicted rating $\hat{r}_{u,i}$ by weighting each rating $r_{u,i}$ by the item similarity $s_{i,j}$.

$$\hat{r}_{u,i} = \frac{\sum_{j \in S_{i,u}} s_{i,j} * r_{u,i}}{\sum_{j \in S_{i,u}} s_{i,j}} \tag{2.7}$$

In contrast to the weighted average, to compute $\hat{r}$, another approach is to use a linear regression to compute an approximation of the ratings instead of directly incorporating the ratings $r_{u,i}$ in a weighted sum. Such a regression is depicted in Equation 2.8. The regression parameters α and β are determined by applying ordinary least squares (OLS), where we denote the error term as ϵ [104].

$$\hat{r}_{u,i} = \sum_{j \in S_{i,u}} \alpha_{i,j} r_{i,j} + \epsilon \tag{2.8}$$

Although more complex, the regression approach could lead to improved results as the cosine similarity based on the raw ratings could be misleading: items can be similar according to the cosine similarity, although the Euclidean distance between two rating vectors i and j is high. This is, as the cosine similarity measure only the parallelness, not the magnitudes of the rating vectors. In this case, approximating the ratings using a linear re-

gression yields better results than incorporating the similarity directly as in Equation 2.7 [104].

Simple non-trivial item-based algorithms are the Slope One predictors as introduced by Lemire and Maclachlan [66]. Key idea of this approach is to infer the predicted rating from the rating differences of the set of co-rated items. For a user, co-rated items are items that have been also rated by other users. Formally, the average difference d between two items i and j can be computed as depicted in Equation 2.9. All users who rated items i and j can be derived from the intersection of the set of all user who rated item i ($S_{i,u}$) and the set of all users who rated the item j ($S_{j,u}$). The number of users is determined by the cardinality of the intersection ($|S_{i,u} \cap S_{j,u}|$).

$$d_{i,j} = \sum_{u \in (S_{i,u} \cap S_{j,u})} \frac{r_{u,i} - r_{u,j}}{|S_{i,u} \cap S_{j,u}|} \tag{2.9}$$

Subsequently, $\hat{r}_{u,i}$ can be computed as depicted in Equation 2.10.

$$\hat{r}_{u,i} = \frac{1}{|S_{i,u} \cap S_{j,u}|} \sum_{u \in (S_{i,u} \cap S_{j,u})} (r_{u,j} + d_{i,j}) \tag{2.10}$$

For a better understanding, we give an example using the rating matrix depicted in Equitation 2.11. The predicted rating $\hat{r}_{3,2}$ for user u_3 towards item i_2 can be computed by using the differences of all co-rated items as described in the following. In this example, for user u_1, the difference between item i_1 and item i_2 is $(3-5) = -2$ and between item i_3 and i_2 it is $(4-5) = -1$. For user u_2, the difference between item i_1 and item i_2 is $(2-5) = -3$ and between item i_3 and i_2 is $(3-5) = -2$. From this, we infer that the average difference between item i_1 and item i_2 is $\frac{-2-1}{2} = -1.5$ and between item i_2 and item i_3 is $\frac{-3-2}{2} = -2.5$ respectively. We know that user u_3 rated items i_1 and i_2. Hence, the Slope One approach uses the average differences to compute $\hat{r}$: $\hat{r}_{3,2} = \frac{(2-1.5)+(3-2.5)}{2} = 0.5$

$$\begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \end{array} \begin{array}{c} \begin{array}{ccc} i_1 & i_2 & i_3 \end{array} \\ \begin{pmatrix} 3 & 5 & 4 \\ 2 & 5 & 3 \\ 2 & - & 3 \end{pmatrix} \end{array} \tag{2.11}$$

Lemire and Maclachlan found that for dense datasets, where a rating $r_{i,j}$ exists for the majority of the item combinations, the computation as shown in Equation 2.10 can be simplified to Equation 2.12. For this simplification, the average among all ratings ($\bar{r}$) is used, as in a dense dataset it holds that $\bar{r} = \frac{1}{|S_u|} \sum_{i \in S_u} r_i \sim \frac{1}{|S_{i,u} \cap S_{j,u}|} \sum_{i \in (S_{i,u} \cap S_{j,u})} r_i$. In the former equation, we denote S_u to the set of all ratings of a certain user u towards any item in the dataset.

$$\hat{r}_{u,i} = \bar{r} + \frac{1}{|S_{i,u} \cap S_{j,u}|} \sum_{u \in (S_{i,u} \cap S_{j,u})} d_{i,j} \tag{2.12}$$

Summing up, by comparing the linear regression-based utility function f_{lr} (cf. Equation 2.13) to the Slope One utility function f_{sl} (cf. Equation 2.14), we observe a significant difference in complexity. For the regression model, both, the weight α and the constant β have to be estimated using OLS or similar methods. In contrast, for the Slope One model, only the constant α has be estimated.

$$f_{lr}(r) = \alpha + \beta r \tag{2.13}$$

$$f_{sl}(r) = \alpha + r \tag{2.14}$$

Besides the presented item-based approaches, a well-known approach which has been shown to work well in the field of music recommendation [128, 114, 87] is user-based collaborative filtering [39]. User-based CF uses the same user-item matrix as input as depicted in Equation 2.4, but leverages user similarity rather than item similarity. The main idea behind this approach is to find groups of similar users based on their rating behavior. Based on the most similar users, considered as the nearest neighbors, items for a given user are recommended by choosing the items the nearest neighbors rated positive and that are new to the user.

As for item-based CF, it is common to use the Pearson correlation [97, 21, 45] or the cosine similarity [21, 45] to estimate the similarity $s_{u,v}$ between two users u and v. Hence, in contrast to item-based CF, these measures are applied to the rows of the rating matrix and hence to all users rather than on the columns and thus, to all the items. This is shown in Equations 2.15 and 2.16. In the first, the vectors $\vec{u}$ and $\vec{v}$ represent the row vectors i and v of the rating matrix R. Therefore, these vectors contain all ratings of a user u or v towards any

item i in the rating matrix R. We see the application of the cosine similarity in Equation 2.15, where we compute the similarity between the users u and v.

$$s_{u,v} = cos(\vec{u}, \vec{v}) = \frac{\vec{u} * \vec{v}}{\|\vec{u}\|\|\vec{v}\|} \tag{2.15}$$

Analogous to the cosine similarity, a correlation based similarity can be computed between two users u and v as depicted in Equation 2.16. We denote $\bar{u}$ to the average rating of a user computed among all rated items, *cov* as the covariance between two vectors and σ as the standard deviation.

$$s_{u,v} = \frac{cov(u,v)}{\sigma_u, \sigma_v} = \frac{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_v)}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_v)}} \tag{2.16}$$

To compute the predicted rating $\hat{r}_{u,i}$ in the final step and hence to predict the utility of a user u towards an item i, a user-based CF recommender system relies on the user neighborhood. As depicted in Equation 2.17, we estimate $\hat{r}_{u,i}$ by taking the ratings of the k-nearest-neighbors for an item i into account. This is done by computing a weighted average using the user similarity $s_{u,v}$ of all users in the set of k-nearest-neighbors S_u. This is why the algorithm needs to know k in advance. The parameter k must be tuned in offline experiments as presented in Section 2.4.

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{j \in S_u} s_{u,v} * r_{u,i}}{\sum_{j \in S_u} s_{u,v}} \tag{2.17}$$

Summing up, in this section, we firstly presented where the term collaborative filtering came from. Along with that, we discussed the two basic CF algorithms: item- and user-based CF. In the next two sections, we focus on more advanced models and differentiate between context-agnostic and the relatively new context-aware approaches.

2.3.3 Context-agnostic Model-based Algorithms

Besides the presented pure CF-based approaches, approaches utilizing matrix factorization (MF) techniques, for instance, singular value decomposition (SVD), have been shown to deliver even better recommendation accuracies [105, 58, 15]. Those approaches are also known as latent factor models, as factorizing the user-item matrix yields a latent representation of user-item rat-

ings and hence to a representation on a more abstract level. Besides SVD, the principal component analysis (PCA) [58] which often uses SVD as the computational kernel and non-negative matrix factorization (NMF) are applied. The core idea of this approach is to measure how similar a user u and an item i, both with certain characteristics expressed in the latent features vectors, are. These latent feature vectors correspond to the principal components if a PCA is applied or to the singular vectors if SVD [62] is applied. A high correspondence of the latent user- and item vectors leads to a recommendation. The features discovered by a MF approach might be obvious (i.e., instrumental music versus rap music) or hidden (i.e., music with an orientation to European male users). The latter is not obvious from the musical characteristics itself. Based on a rating matrix as presented in Equation 2.4, a MF-based recommender system learns the utility function f_R, using stochastic gradient descent (SGD) or alternating least squares (ALS) [63]. In Equation 2.18, we depict a SVD model. In this model, $U \in \mathbb{R}^{M \times k}$ and $V \in \mathbb{R}^{k \times N}$ are orthogonal factor matrices, that embed users and tracks onto a lower dimensional space of k latent features. $\Sigma \in \mathbb{R}^{k \times k}$ is the matrix of singular values, estimating the impacts of the latent features to a rating r. Please note that R' is the closest approximation to $R = U\Sigma V$, where only the k largest eigenvalues are used. Using this representation, a single rating $\hat{r}$ can be estimated using the dot product between the feature vector of the user $\vec{u_u}$ and the feature vector of the item $\vec{v_i}$ as depicted in Equation 2.19.

$$R' \approx U\Sigma V^T \tag{2.18}$$

$$\hat{r}_{u,i} = \vec{u_u} \cdot \vec{v_i} = \sum_{f=0}^{k} u_{u,f} v_{f,i} \tag{2.19}$$

A common method is to extend this model with a global bias μ, a user bias b_u also known as *baseline predictor* as well as an item bias $\bar{r}_i$. The global bias is computed by computing the average rating of R and the user- and item bias is estimated by computing the average rating of a user u and analogously, computing the average rating of an item i. This leads to a model as stated in Equation 2.20.

$$\hat{r}_{u,i} = \mu + \bar{r}_i + b_u + \vec{u_u} \cdot \vec{v_i} \tag{2.20}$$

Due to the general success of latent feature approaches, several extensions have been proposed. One extension, SVD++ [61], improves the recommendation

accuracy by incorporating implicit user feedback. Also extensions incorporating contextual information into a recommender system, i.e., timeSVD++ [62], have been proposed. We discuss such approaches in the next section, where we present context-aware model-based CF algorithms.

2.3.4 Context-aware Model-based Algorithms

Several extensions to the classical MF model as presented in the previous section have been introduced. Amongst others, extensions for implicit feedback data [49, 94] and extensions for context-aware recommendations [62, 14] are proposed. As recently, contextual data became easily available through various devices equipped with different sensors (i.e., smart phones, tablets, fitness trackers, ...), we observe a rise in research on context-aware recommender systems [3, 108]. Context can be considered as circumstances influencing the perceived usefulness of an item. Hence, incorporating contextual information into a recommender system improves the recommendation accuracy. Thus, it is widely agreed upon the fact that the incorporation of a user's context improves personalized recommendations [3] and we observe a shift from purely CF-based approaches towards more user-centric approaches incorporating the user's context [108]. This development is especially valid for the field of music recommender systems, as studies showed that users often seek for music that matches their current context for instance the occasion, event or emotional state [59, 64]. To incorporate these findings in music recommender systems, different data sources are exploited. Examples for contextual data or information that is leveraged for music recommendations are emotion and mood [41, 99, 11, 20], the user's location [55, 57, 7, 28] or recommending music fitting to documents on the web a user reads at the moment [24]. As for these different types of contexts, Kaminskas and Ricci [56] distinguish (i) environment-related context (location, time or weather), (ii) user-related context (activity, demographic information, emotional state of the user) and (iii) multimedia context (text or pictures the user is currently reading or looking at).

Adomavicius and Tuzhilin [3] classify approaches incorporating the user context into (i) contextual pre-filtering, (ii) contextual post-filtering and (iii) contextual modeling approaches. The first two approaches apply non-contextual models to the recommendation problem by applying a filter for a context prior or after the recommendation computation. In contrast, the latter recommendation approaches leverage contextual information directly in the model. The filtering approaches are easy to implement and deliver an increased prediction accuracy for music recommender systems. We show this in our prototype implementation in Chapter 6. Nevertheless, contextual modeling approaches are interesting as they promise to deliver even better recommendation accuracies.

Based on classic MF approaches and based on timeSVD++ [62], Baltrunas et al. [14] invent a context-aware matrix factorization (CAMF) approach capable of handling an arbitrary number contextual factors k as depicted in Equation 2.21.

$$\hat{r}_{u,i,c_1...c_k} = \bar{r}_i + b_u + \vec{u_u} \cdot \vec{v_i} + \sum_{c=1}^{k} B_{u,i,c} \tag{2.21}$$

As presented in the SVD model in Equation 2.20, we denote the user and item vectors as $\vec{u_i}$ and $\vec{v_j}$. Further, we denote $\bar{r_i}$ to the average rating of an item i. Finally, b_u is the baseline predictor for a user u to which an individual contextual bias $B_{u,i,c}$ is added for all k contextual dimensions. This contextual bias replaces the global bias μ as depicted in Model 2.20. In such a model, a bias for each context c, for each item i and for each user u has to be estimated. The model can be reduced in complexity by only estimating a bias for each user and context combination $c_{u,c}$, without considering the item. Analogously, only a bias for each item and context combination $B_{i,c}$ can be considered. The simplest model is a model where a bias is only estimated for a certain context B_c [14]. Besides that, to better fit the data and hence to increase the recommendation accuracy, the quadratic interaction effects of the different contexts as depicted in Equation 2.22 can be added [14].

$$\hat{r}_{u,i,c_1...c_k} = \bar{r}_i + b_u + \vec{u_u} \cdot \vec{v_i} + \sum_{c=1}^{k} B_{u,i,c} + \sum_{c=1}^{k} \sum_{l=c+1}^{k} B_{u,i,c,l} \tag{2.22}$$

However, this model suffers from an increased complexity and there is the problem of data sparsity that makes it hard to estimate the additional parameters. The data sparsity is caused by the problem that most user have not rated the items they interacted with in every context. Simultaneously with CAMF, a different context-aware approach for recommendation and classification tasks has been invented: the factorization machines by Rendle et al. [92]. This approach is able leverage the discussed interaction effects and moreover is able to do this under high sparsity, as we present in the following section.

2.3.5 Factorization Machines

A recent and highly successful enhancement of CF, as they have been proven to be one of the most successful context-aware recommendation approaches up to now, are factorization machines (FM) invented by Rendle et al. [92]. Generally, FMs combine the advantages of support vector machines (SVM)

or broadly speaking regression approaches for classification with factorization models. Factorization enables the FM to model all interactions between variables even under high sparsity of data in linear time [92]. Hence, they solve the problem of computational complexity caused by the integration of interaction effects between contexts in CAMF models [14]. Besides this, in contrast to the presented MF approaches, the model variables can be metric, nominal or ordinal. This allows to integrate different types of contexts, for instance nominal variables as colors or weekdays. We depict the classical FM model in Equation 2.23.

$$\hat{r} = \mu + \sum_{i=1}^{n} w_i b_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \vec{v_i}, \vec{v_j} \rangle b_i b_j \tag{2.23}$$

In Equation 2.3.5, we see that a FM computes the rating predictions by modeling a global bias μ as well as the influence w_i of an arbitrary number of n additional features b_i along with the quadratic interaction effects of those features ($\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \vec{v_i}, \vec{v_j} \rangle$). Besides a user bias and hence, the influence of the user, an item bias and hence the influence of the item, which are features we know from traditional matrix factorization approaches, additional contextual features as weekdays or user groups can be added into the model. However, instead of learning all weights $w_{i,j}$ for the interaction effects of the features directly, a FM relies on factorization to model the impact of the interaction as the inner product of two lower dimensional vectors ($\langle \vec{v_i}, \vec{v_j} \rangle$) [92]. In contrast to MF approaches that leverage SVD, FMs leverage Cholesky decomposition as depicted in Equation 2.24. Given the number of factors k, Cholesky decomposition decomposes the matrix $W \in \mathbb{R}^{n \times n}$ containing the weights $w_{i,j}$ that have to be additionally learned if quadratic interactions are added, to an upper triangular matrix $V \in \mathbb{R}^{n \times k}$. Using V, the parameters $\hat{w}_{i,j}$ can be learned efficiently in $\mathcal{O}(kn)$ time, instead of learning the weights $w_{i,j}$ directly from W which demands $\mathcal{O}(n^2)$ time. Furthermore, as the independence of the interaction weights is broken, FMs work under high data sparsity [92].

$$W = V \cdot V^T \tag{2.24}$$

$$\hat{w}_{i,j} = \langle \vec{v_i}, \vec{v_j} \rangle = \sum_{f=1}^{k} v_{f,i} v_{f,i} \tag{2.25}$$

Using $\hat{w}$ we can rewrite the FM model depicted in Equation 2.23 to the FM model depicted in Equation 2.26.

$$\hat{r} = \mu + \sum_{i=1}^{n} w_i b_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j \tag{2.26}$$

A further recent enhancement of FMs are the training algorithms for higher order Factorization Machines (HOFM) [77, 18]. Although HOFMs were mentioned by Rendle [92], no training algorithm for HOFMs was proposed. As an application, Blondel et al. [18] show that HOFMs are well suited for link predictions. The model of a higher order FM is depicted in Equation 2.27, where we see that the main difference is the addition of higher order interaction effects, i.e., cubic effects as in the given model.

$$\hat{r} = \mu + \sum_{i=1}^{n} w_i b_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{l=j+1}^{n} \hat{w}_{i,j,l} b_i b_j b_l \tag{2.27}$$

Besides that, inspired by the work of Rendle and Schmidt-Thieme [96], Field-aware Factorization Machines (FFMs) have been invented. FFMs factorize the features of the feature matrix pairwise and thus, in separate latent spaces (fields). They have been leveraged for click-through rate (CTR) predictions [53]. In contrast to FMs, FFMs exhibit a quadratic computational complexity with the number of fields.

In this work, we rely on FMs in order to leverage different contextual features mined from social media and streaming platforms. To validate our proposed approach and to benchmark the approach against other recommendation approaches, evaluations are necessary. For these evaluations, often also referred to as experiments, several methods and measures have been proposed in the information retrieval and recommender systems community. We present these evaluation methodologies along with the measures in the next section.

2.4 Recommender Systems Evaluation

As outlined in the introductory part of this chapter, there are two main tasks of recommender systems: Firstly, there is the rating prediction task. Secondly, there is the top-n recommendations task, which is also referred to as the retrieval task or find good items task [46]. For both, the same evaluation

setups can be used. However, different recommendation metrics are necessary. We firstly introduce different evaluation setups that are commonly used in the field of recommender systems, before focusing in the actual evaluation metrics.

2.4.1 Evaluation Setups

Herlocker et al. [46] give an extensive overview of the evaluation of recommender systems and related issues. A first differentiation of evaluation setups is whether an evaluation is done offline, solely using a dataset containing previous user-item interactions, or is done with real users in a live user experiment. Whereas the latter delivers better feedback, they are very costly compared to offline analyses. Along with that, there is the problem, that users should objectively evaluate the recommendations and hence, the recommendation algorithm and not any other influences as the user interface [46]. In offline evaluations, a certain portion of the data is withheld to compare the recommendations computed using the remainder of the data to the withheld data. Using this method, experiments with thousands of users can be repeated continuously to train and evaluate recommendation algorithms. However, they are limited in feedback. Firstly, there is the problem of data sparsity. I.e., only items in the dataset a user interacted with can be evaluated and this set of items is probably limited. Secondly, there is the difficulty, that often only implicit feedback as interactions, play counts or dwell times are available. In contrast to explicit feedback in terms of ratings, there is a certain uncertainty, if a user really liked an item.

Besides those limitations, offline evaluations are popular in the field of recommender systems. As already mentioned, for offline evaluations, a certain portion of the data is withheld to compare the recommendations computed using the remainder of the data to the withheld data. Hence, for conducting such offline analyses, a data splitting method is necessary. Besides time based splitting of the data, which might be the most realistic setup [118], there are two prominent splitting strategies: (i) sample a fixed number of items for each user known as the *given-n* or *all-but-n* method and (ii) non-overlapping sampling of a percentage of the dataset, where each sample is evaluated once. This is known as *cross-validation* [101]. We elaborate on both methods in more detail next.

All-but-n

If the all-but-n evaluation methodology is applied, n items are held out. This set of holdout items is referred to as the test set and the remainder of the dataset is referred to as the training set. The hold out of the user-item inter-

actions can be done either per-user or globally for the whole dataset. The first method guarantees that all users are represented in the training and the test set. Using the training set, n recommendations are computed and compared to the test set using evaluation metrics as described in Section 2.4.2 [101]. In case the data was split on a per-user basis, the computed results are averaged among all users. Often different measures are given for a different number of n. We refer to this as $measure@n$.

k-fold Cross-validation

As outlined, cross-validation is one of the most prominent evaluation methodologies [101]. For k-fold cross validations, we observe that $k = 5$ and hence, an 80% to 20% training to test ratio or $k = 10$ and hence, a 90% to 10% ratio are widely used parameters. To perform this evaluation, the dataset is randomly split into k folds of equal size. Analogously to the all-but-n method, this is done either for each user (per-user cross-validation) or done for the whole dataset (global cross-validation). For training a recommendation model, the first $k - 1$ folds, in case of $k = 5$ the first 4 folds which equals to 80% of the dataset, is used. The recommendations computed on this training set are then evaluated against the remaining 20% of the user-item interactions. In a k-fold cross-validation, this process is repeated k times, holding out a different fold each time, such that each interaction is in the test set once. The evaluation metrics, as described in the following section, are computed for each fold separately and finally averaged over all folds.

Besides an evaluation methodology, to assess the performance of recommender systems, also evaluation metrics are necessary. We describe those metrics in the following section.

2.4.2 Evaluation Metrics

To assess the performance of recommender systems, different measures have been proposed since the mid 1990s [46]. Herlocker et al. classify these metrics into predictive (rank) accuracy metrics suitable for assessing the rating prediction task as well as classification accuracy metrics useful for assessing the retrieval task. We present measures for both tasks in the following.

Metrics for the Retrieval Task

For assessing the retrieval task, the recommendation confusion matrix [47] as depicted in Table 2.1 is a helpful concept to compute two popular measures borrowed from the field of information retrieval [10]: *precision* and *recall* [29].

	relevant	non relevant
recommended	TP	FP
not recommended	FN	TN

Table 2.1: Recommendation Confusion Matrix [47]

In the confusion matrix, we denote recommended and relevant items as *true positives* (TP) and relevant items that were not recommended as *false negatives* (FP). Analogously, we denote recommended but non relevant items as *false positives* and not recommended and non relevant items as *true negatives*. Using these definitions, we can define the *precision* and *recall* measures [29] as depicted in Equations 2.28 and 2.29.

$$precision = \frac{TP}{TP + FP} \tag{2.28}$$

Precision is the ratio between the number of true positives and the number of recommended items ($TP + FP$). In contrast, recall is the ratio between the number of true positives and the number of relevant items $|I_R|$. In Equation 2.29, we denote I_R as the set of relevant items. We see, that due to their definition, both measures range between 0 and 1.

$$recall = \frac{TP}{|I_R|} \tag{2.29}$$

Precision and *recall* are measures that can't be considered in isolation, as a system can always achieve a *recall* of 1 which is 100% by simply returning all items in the dataset. In return, such a system achieves low *precision* values. Based on the harmonic mean of both measures, the F_1-score which combines both of them with a bias towards the lower value can be computed as depicted in Equation 2.30.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{2.30}$$

The generalization of the F_1-score is the F_β-score depicted in Equation 2.31. Two popular values for β are 2 and 0.5. Whereas the first weights *recall* two times higher than precision, the latter weights *precision* two times higher [100].

$$F_1 = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall} \tag{2.31}$$

After presenting measures suitable for evaluating the retrieval task in this section, we elaborate on measures for evaluating the rating prediction task next.

Metrics for the Rating Prediction Task

For assessing the rating prediction task, predictive accuracy metrics are popular in the field of recommender systems. Those measures compute the difference between the predicted ratings $\hat{r}$ and the actual ratings r of a user [46]. As already outlined in Section 2.2, the predicted rating is often used to compute the top-n recommendations for the find good items task. This is, as the predicted ratings allows an ordering of the recommendations. Hence, the recommendations candidates of recommender systems are ordered by $\hat{r}$ and afterwards cut off at position n [45]. Popular measures for computing the difference between $\hat{r}$ and r are the mean absolute error (MAE), mean absolute percentage error (MAPE) and the root mean squared error (RMSE), as known from statistics and time series analysis [8]. Whereas the MAPE is unit free but needs the rating data to have meaningful zero points, in academia the unit dependent RMSE is mostly used [8]. The MAE for assessing n ratings is depicted in Equation 2.32.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |r_i - \hat{r}_i| \tag{2.32}$$

We see, that the MAE measures the average absolute deviation between $\hat{r}$ and r. A similar measure is the MAPE, as depicted in Equation 2.33, where the average deviation is expressed as percentage value.

$$MAE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{r_i - \hat{r}_i}{r_i} \right| \tag{2.33}$$

In contrast, the RMSE as depicted in Equation 2.34, measures the quadratic error and is hence more sensitive to deviations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{r}_1 - r)^2}{n}} \tag{2.34}$$

For assessing the ranking and hence the position of the items in the top-n recommendations list, the mean reciprocal rank (MRR) has been invented in the field of information retrieval [125]. The MRR, as depicted in Equation 2.35, computes the mean reciprocal rank for n recommendations. We refer to $rank_i$ as the position of the i^{th} recommendation in the top-n list. This measure has been invented, as for maximizing the user experience, the most relevant items should be stated on top of the top-n recommendations list. Hence, the MRR can be used to complement the precision as presented in Equation 2.28, as the precision only measures if an item is contained in the top-n list, independent of the position.

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \tag{2.35}$$

We observe, that various measures for assessing the rating prediction task have been proposed and used in industry as well as academia. Most of them are borrowed from the field of time series analysis and statistics or from the field of information retrieval and there exists no "one fits all" error measure. For instance, in contrast to the MRR, precision neglects the position of the item in the top-n recommendations list. Moreover, both measures do not consider the total number of relevant items a user might be interest in as recall does. Hence, we opt for computing a set of different measures, each of them covering different aspects. This is what we did in our experiments to assess the performance of our recommendation approach: we computed precision and recall to assess the top-n retrieval task as well as the RMSE and MAPE to assess the rating prediction task.

2.5 Summary

In this chapter, we introduced the reader to the main tasks of recommender systems: Firstly, there is the item retrieval task, also referred to as the top-n recommendations or find good item task [98, 46]. Secondly, there is the rating prediction task [98]. Whereas the first task aims at finding n items the user is most interested in, the second task tries to predict the rating a user would give towards a certain item. On a more abstract level, the rating can be interpreted as the user's perceived usefulness towards an item. As the rating prediction gives an ordering to items, the predicted ratings can be used to generate the top-n recommendations list by sorting all items by their predicted rating and cut off at position n. Along with that, we presented a set of popular content- and collaborative filtering approaches, which were developed to fulfill these two tasks. Although we discussed content-based filtering

approaches, we set a strong focus on model-based collaborative filtering using matrix factorization techniques. This focus was set, as these approaches have been shown to deliver the best performance in several recommendation scenarios, i.e., in the Netflix Challenge[6] [63]. Along with that, we presented context-aware models as context-aware matrix factorization and factorization machines. Generally, we refer to context as any circumstance that influences the perceived usefulness of an item. Hence, incorporating contextual information into a recommender system improves the recommendation accuracy [3]. Along with that, we pointed out that also for the proposed music recommender system in this work different matrix factorization techniques are leveraged.

Besides presenting different recommendation models, we elaborated on different evaluation methodologies also known as experimental setups. For this, we presented a set of metrics and measures to assess the performance of recommendation algorithms with respect to the top-n recommendations task as well as the rating prediction task in offline experiments.

[6] http://www.netflixprize.com

CHAPTER 3

Related Work: Music Information Retrieval

3.1 Introduction

Recommender systems as introduced in the previous chapter are heavily leveraged in the field of music information retrieval (MIR). With respect to them, there are certain domain specific aspects of recommender systems that need to be covered. Hence, in this chapter, we introduce the reader to MIR and elaborate on MIR related aspects with a special focus on music recommender systems. Today, MIR is one of the most important IR research fields. This is, as independent of the society or culture, people enjoy listening to different forms of music in various situations in daily life [109]. Along with that, we observe two trends that make music retrieval facilities inevitable: Firstly, there is the trend that more and more users use streaming platforms and hence access large music collections in the cloud making millions of tracks easily accessible [65]. Secondly, initiated by this new distribution channel, there is the trend that the music market offers a lot more music and particularly a greater variety of music, i.e., a lot more "niche-music" became available recently [26].

We examine both trends along with their implications in more detail in the remainder of this section.

Recently, music consumption changed due to the rise of the web along with the dissemination of smart mobile devices as smartphones or tablets. Today, music is consumed in various different situational listening contexts (i.e., during a certain activity, fitting a certain mood or during a certain occasion) throughout the whole day. Along with that, music streaming platforms as Spotify[1], Deezer[2] or Apple Music[3] emerged and allow users to access millions of tracks easily. This drove the development, that more and more users switch from private, mostly limited music collections, which were locally stored on hard drives and CDs, to public, nearly unlimited music collections in the cloud [65]. Users cannot browse this enormous amount of music tracks manually to find music they like. Hence, streaming platforms heavily rely on MIR facilities as music exploration services and recommender systems. These systems guide the user through the large music libraries and hence help them to discover (new) music they like.

Before music was digitally distributed, according to Celma [26], the "hit vs. miss rule" was valid and coined the availability of music. This rule states, that in brick and mortar stores, the sales space is limited and along with that the covered customer catchment area is often too small to offer a broad assortment of niche music. Hence, mostly very popular music was offered. In contrast, nowadays, this rule does not hold for music and most other digital goods as movies or apps anymore: today, a track can be produced once and the digital version can be offered via streaming platforms or online stores to a worldwide market without any further production and inventory costs [6, 26]. Consequently, the variety of music heavily increased and today, we face a long-tailed distribution of consumed musical tracks. In case of music, a long-tailed distribution can be described as a distribution where there are a few popular tracks with high play counts forming the head and where there are many more tracks with low play counts forming a long tail. Those tracks with low play counts forming the long tail are considered as niche music [6, 26]. As there are many tracks in the long tail, there exists a lot of music to the favor of the users, but they are not aware of it. Before we give an overview of music retrieval facilities that help users discovering new music from the long tail they like, we define the field of MIR in the next section.

[1] `https://www.spotify.com`, last visited November 26, 2017

[2] `https://www.deezer.com` last visited November 26, 2017

[3] `https://www.apple.com/music/`, last visited November 26, 2017

3.2 Definition and Applications

Music information retrieval is a relatively young field compared to classical information retrieval. MIR started in the late 1990s with the invention of audio compression techniques [109]. According to Downie [34], MIR has many facets and sub-fields. In this work, we classify MIR into two main fields: The *music analysis* as well as the *music retrieval* task. Whereas the first task is concerned with extracting meaningful features, the latter is concerned with indexing music by leveraging the features developed in the former task. Indexing is the first step to build models for music search and retrieval.

With respect to feature engineering, we observe that there are two main directions: (i) content-based features mined directly from the audio signal of tracks as well as (ii) features based on the meta-data of the tracks. Content-based features aim at indexing tracks via their audio characteristics, which are acquired by signal processing. Feature engineering based on the meta-data of the tracks is a broad field: Features leverage conventional classifications as the genre, exploit the lyrics of the tracks using text-mining algorithms known from the field of information retrieval and also focus on user-generated content in the web, i.e., user-generated tags from MusicBrainz[4] or last.fm[5] [108, 109]. A rather new development is the research on user-centric features. We observe that contextual music consumption information, for instance the mood or the current activity, is mined from social media platforms as Twitter[6] or from music streaming platforms as Spotify[7] [108, 129, 87, 88, 89, 90].

Based on the extracted features, various music retrieval models have been invented and numerous music retrieval applications using these models were built. These retrieval systems reach from the "query by humming" approach as proposed by Dannenberg et al. [32] over three-dimensional musical maps guiding a user through music libraries [71] to different types of music recommender systems leveraging different data sources. Based on the leveraged data and the applied model, for the latter we differentiate between *content-based* [25, 81, 127], *collaborative filtering-based* [128, 85, 86] as well as *context-aware* [99, 41, 20, 12, 56, 89] recommendation approaches. In the next three sections, we will give the reader an overview of those music recommendation approaches.

[4] `https://musicbrainz.org`, last visited November 26, 2017

[5] `https://www.last.fm`, last visited November 26, 2017

[6] `https://twitter.com`, last visited November 26, 2017

[7] `https://www.spotify.com`, last visited November 26, 2017

3.2.1 Content-based Music Recommender Systems

Content-based recommender systems exploit mostly the audio characteristics of a track for track recommendations [26], but also incorporate meta-data as the genre [9, 5]. In principle, those systems compute a similarity metric based on audio-features aiming to retrieve tracks similar to a given track. For instance, MusicSurfer finds similar tracks to track a user stated to like [25] but also systems creating whole playlists were invented. The latter facilitates a seed song along with the traditional k-nearest neighbors approach to find similar songs to the given start song [70]. Later, more advanced approaches in which the user selects a start and an end song with a smooth transition in between [36] and approaches based on user-defined constraints [9] have been proposed. The used constraints may be content-based, i.e., based on the tempo or the loudness of a song, or based on meta-information like the genre [9, 5]. Besides the presented filtering-based approaches, also neural networks and more recently, deep learning approaches and hence convolutional neural networks are leveraged [81]. Oord et al. [81] use deep learning to avoid the cold-start problem of unrated items as it occurs if collaborative filtering-based approaches are applied. Furthermore, deep learning approaches try to overcome the limitations of signal processing techniques. Although they are used for content-based recommendations, they were initially not intended for music recommendations and hence do not represent all relevant information [127].

3.2.2 Collaborative Filtering-based Music Recommender Systems

Collaborative filtering-based recommender systems, where we also subsume association rule-based systems, often leverage publicly available data [128, 85] crawled from Twitter[8] or last.fm[9]. One of the first of this type of systems was the system introduced by Zangerle et al. [128], a recommender system that computes association rules based on the listening behavior of Twitter users. The listening behavior is extracted from #nowplaying tweets. These are tweets in which users tweet to which musical track they are currently listening to. Later, the leveraged dataset named the #nowplaying dataset[10] was made publicly available [129] and is continually updated with new tweets until today. Simultaneously, the Million Musical Tweets Dataset[11] was made publicly available by Hauger et al. [43]. The presented recommender system [128] along with the presented ideas and statistical analyses of tweets [43, 110] in conjunction with the data which was made publicly available [43, 129] laid the

[8] `https://twitter.com`, last visited November 26, 2017

[9] `https://www.last.fm`, last visited November 26, 2017

[10] `dbis-nowplaying.uibk.ac.at`, last visited November 26, 2017

[11] `http://www.cp.jku.at/datasets/MMTD/`, last visited November 26, 2017

foundation for more advanced systems. Those advanced systems, including the proposed music recommendation approach presented in this work, utilize model-based collaborative filtering techniques [89]. They often base on the same data (or an enriched version) as the initial systems and close several research gaps pointed out in these early works.

3.2.3 Context-aware Music Recommender Systems

Triggered by the rise of social media platforms and hence, rooted in the rise of publicly available data, research on context-aware recommender systems for music recommendations got in the focus of the IR and in particular of the MIR community. Context-aware approaches are often hybrid approaches, i.e., using content-based and contextual information to enhance a CF-based base model [112, 114, 89]. Approaches recommending music fitting to web pages a user reads at the moment [24] and approaches incorporating the user's current emotion and mood [41, 99, 11, 20] have been among the first context-aware approaches. Later, as activity- and location information became available by facilitating rich sensory devices as smartphones, this type of information was exploited to provide personalized music recommendations. I.e., the location of a user during the day [126], during driving a car [12] or the user's location with respect to points of interest [20] was leveraged. Along with that, as today, GPS-tagged #nowplaying tweets are available, approaches incorporating the geodesic distance between two users in CF-based recommendation approaches were developed [113]. Also, models approximating the cultural distance of users by the countries or continents they are located in were proposed [114]. In conjunction with this research, Hauger and Schedl [42] visualize artist and genre distributions on interactive maps to let researchers explore regional listening patterns. Schnitzer et al. [112] conclude that if users listen to various different artists, the integration of geospatial information is beneficial. In a later study they moreover observe that countries or continents do not necessarily reflect cultural borders [113]. The latter is an open issue, which we cover in this work. Besides that, we propose a different approach to mine for activity and occasion related information about the music consumption context: We extract contextual information from the names of user-created playlists of the music streaming platform Spotify [87, 89].

3.3 Current Trends in Music Information Retrieval

A recent development in MIR is the focus on the user, rather than solely focusing on the audio characteristics or meta-data of music. Although a user focus is naturally important, as music is perceived differently by different users and is perceived differently by the same user in different situations, the user-focus

was neglected in the beginning of MIR research [108]. We lead this back to a lack of data: For research on user-centric music discovery and recommendation facilities, data about the user and the current music consumption context is necessary. Today, in contrast to the CD era, this information is available. This is, as firstly people use social media platforms to share to which track they are currently listing to and secondly more and more people use music streaming platforms and music discovery services as last.fm. Those platforms and services generate interesting data, which can be crawled via public APIs. This data availability drove two trends, namely (i) engineering of user-centric features comprising the user context and (ii) developing context-aware recommendation algorithms leveraging these features. We present both trends in more detail in the next two sections.

3.3.1 User-centric Features

In the following, we present four factors that influence the music perception of a user as defined by Schedl et al. [108]: Firstly, there is the *musical content*. As elaborated on earlier in this chapter, the music content can be represented by features derived via signal processing and has been studied well. Secondly, there is the *music context*. Music context subsumes any additional information about the track, i.e., the lyrics, the album cover artwork, the music video clip of the track or any other background information about the artist or track. This type contextual information has been successfully leveraged for music retrieval systems [60, 111]. Thirdly, there is the *user context*. Analogously to the music context, this is any additional information about the music consumption of the user. Examples of this type of information are the current emotion or mood of a user, spatial-temporal aspects or information about the current activity while listing to music. We refer to the latter as *situational music consumption context*. Fourthly, there are *user properties* as demographic features or musical experience and musical preferences. The latter two, *user context* and *user properties* are user-based factors influencing the music perception. These factors are currently especially of interest, as with the rise of social media and music streaming, data for large-scale analyses became available.

In this work, we present feature extraction methods for both, *user context* and *user properties* which we invented and published in prior works. In Chapter 6, we present our proposed approach for extracting activity and occasion related contextual information from Spotify playlist names [87]. For this task, we rely on traditional text mining methods known from information retrieval in order to group playlists to "situational playlist clusters". Our findings are congruent with the findings of Cunningham et al., who found in a qualitative user study that users are organizing music according to the intended use [30]. In a follow-up study [88], we analyzed the acoustical characteristics of those playlists. This

analysis aimed at finding listening patterns, in particular, which type of music is listened in which situational context. Based on the musical features that characterize a playlist, we found five typical types of user-created playlists. We refer to these five types as "playlist archetypes" and found that users listen to these playlist archetypes in various situational contexts. We present the details of these studies in Chapter 7. Although these archetypes are mined from user-generated Spotify playlists, we argue that these archetypes are also suitable for grouping or classifying musical tracks in general, as we found the same archetypes in the LFM-1b dataset[12] [106, 107] presented in Chapter 4.

Besides our approach for extracting the situational music consumption context and the method to map this context to playlist archetypes, we also present an approach for mining for cultural music listening patterns in Chapter 8. Recently, approaches for leveraging the geodesic distance between two users for approximating a geographic or cultural user similarity [113] have been proposed. This similarity has been subsequently incorporated into collaborative filtering recommender systems. Simultaneously, approaches for approximating the cultural distance by the country or continents a user is located in [114] have been presented. However, the underlying assumption that culture matches with political borders neglects the existence of ethnic groups within and beyond country borders. Therefore, a measure that integrates musical similarity and cultural similarity beyond countries' geographical borders is called for. We close this research gap by our approach presented in Chapter 8. We model the user similarity by integrating two dimensions: The first dimension incorporates the users personal listening habits crawled from Twitter and described by the acoustical features of the listened tracks. The second dimension incorporates the cultural characteristics based on socio-economic and cultural factors derived from the World Happiness Report[13] [91].

In Chapter 5, we present our context-aware music recommender system leveraging the user-centric features we introduced. However, before that, we give a brief overview of context-aware recommender systems in the MIR domain in the next section.

3.3.2 Context-aware Music Recommendation

For a recommender system, it is important to precisely estimate a user's perceived usefulness towards an item. The MIR and recommender systems communities agree, that this perceived usefulness is context-dependent. I.e., in a different context, the same item is perceived differently by a user. It has been

[12] `http://www.cp.jku.at/datasets/LFM-1b/`, last visited November 26, 2017

[13] `http://worldhappiness.report`, last visited November 26, 2017

shown that this is especially valid for music [108]. In particular, context-aware systems perform superior compared to context-agnostic systems. This is, as the track recommendations retrieved by context-aware recommender systems are better fitting the users' needs than those computed by classical collaborative filtering or content-based algorithms not considering any context. This is why the MIR community calls for research on context-aware music recommender systems [1, 108].

Following up this research we show that besides modeling the influence of different types of user-, content- and contextual information in isolation, music recommender systems modeling the interaction effects provide the most precise fitting recommendations. To give an example, we combine the situational information mined from playlist names, the playlist archetypes and the user listening history in a context-aware recommender system. Such a recommender system learns which type of user prefers which type of music in which situation. Although Baltrunas et al. [14] already stated in 2011 that the recommendation accuracy can be improved by adding such interaction effects, they neglected these models due to their computational complexity and data demands. The latter is a problem as due to the data sparsity caused by the long-tailed distribution of user-item-context interactions, not enough data is available to fit a model with quadratic interaction effects in a setting with several contexts. In this work, we overcome both problems by facilitating factorization machines [92], a technique that allows to model and estimate the impact of quadratic and higher-order interaction effects in linear time and even under high data sparsity.

3.4 Summary

In this chapter, we introduced the reader to MIR along with current trends, open issues and the domain specific aspects of recommender systems. Currently, MIR is concerned with focusing on the user and the user's context during consuming music, rather than on the musical content itself as in the beginning of MIR research [108]. With respect to this user-focus, we located several research gaps: Firstly, data regarding the information about the current listening context for training and evaluating context-aware recommendation models is rarely available. Although the current activity of a user has already been exploited to provide more personalized music recommendations, i.e., based on the location of a user during the day [126] or even during driving a car [12], no large datasets are publicly available. Hence, in a prior work, we invented a novel approach for mining the situational music consumption context, for instance, the current activity or certain occasions, from the names of playlists created by Spotify users [87]. Along with that, we conducted a large-scale analysis of Spotify playlists and could extract several playlist archetypes,

which are typical user-created playlists. We characterize the playlists and the playlist archetypes by the acoustical features of the contained tracks [88, 90]. These archetypes enable us to classify which type music is listened by which user in which situational context. Complementary to information about the music consumption context, also demographic information about the user has been exploited as additional contextual information. For instance, the countries or continents users are located in, have been leveraged to approximate the cultural similarity between those users [113, 114]. However, as political borders do not necessarily reflect cultural borders, a measure that integrates musical similarity and cultural similarity beyond countries' geographical borders is called for. We close this research gap by our proposed cultural similarity model presented in Chapter 8. This similarity is based on musical-, socio-economic- as well as cultural factors. Finally, there is the challenge that different types of contexts, i.e. the current situation and the cultural embedding of a user along with the interaction effects need to be modeled in a context-aware music recommender system. Although user-centric contextual features already improve personalized recommendations if they are modeled isolation, we show that we can provide even better recommendations by modeling an explicit combination of them. Despite quadratic interaction effects have already discussed in the field of recommender systems, they were later neglected due to the computational complexity and the data demands [14]. Often, the available data is too sparse in order to fit a model with several contexts. However, we solved this problem by applying factorization machines [92]. This technique allows us to estimate contextual interactions effects even under high data sparsity in linear time.

CHAPTER 4

Data Sources for Music Information Retrieval

4.1 Overview

As most of the applications presented in the previous chapter rely on publicly available data about music consumption and hence, leverage different public data sources, we briefly present the most prominent data sources for MIR research next. However, we not only introduce famous MIR data sources and datasets which have been heavily exploited to develop and compare different music recommender systems but also present the dataset we created throughout the course of this thesis. We refer to this dataset as the "playlist dataset" and give a detailed description in Section 4.5.

4.2 Echo Nest and the Million Song Dataset

One of the most popular datasets in the field of MIR is the Million Song Dataset (MSD)[1], leveraged for a large-scale personalized music recommenda-

[1] `https://labrosa.ee.columbia.edu/millionsong/`, last visited November 26, 2017

tion challenge in 2012 [76]. It has been released in 2011 and was a joint project between The Echo Nest, a company providing audio analyses and LabROSA, the Laboratory for the Recognition and Organization of Speech and Audio located at the Columbia University in New York [16]. The dataset was never updated and is hence outdated today. Meanwhile, the Echo Nest was also acquired by Spotify. The acoustical features formerly provided by the Echo Nest API are now available via the Spotify API, as presented in Section 4.5.

4.3 last.fm

Last.fm originally started as an internet radio station and is nowadays a social music discovery service. It suggests music to users using their music recommender system called "Audioscrobbler". Similar to the recommender system presented in this work, Audioscrobbler uses the track listening histories of their users as input data. The prior user-track interactions are gathered on the platform itself but also via plugins for music players and streaming platforms as Spotify. As last.fm provides a public API, last.fm data is also heavily leveraged for MIR research. I.e., the team that created the MSD also published a last.fm dataset[2], which is linked to the original MSD. Today, the LFM-1b dataset[3] by Schedl [106, 107] is the largest last.fm dataset containing 120.000 users who listened to 32 million tracks. As this dataset also contains demographic information about the users, amongst others, it has been leveraged for an analysis of country-specific listening patterns [107].

4.4 Microblogging Service Twitter

Since the early 2010s, #nowplaying tweets became a popular data source. We refer to #nowplaying tweets as tweets in which users tweet to which musical track they are currently listening to. The research group Databases and Information Systems (DBIS) at the University of Innsbruck is crawling #nowplaying tweets by crawling tweets containing the hashtags *#nowplaying, #listento* and *#listeningto* along with their metadata since 2011. By leveraging the Twitter Streaming API, which allows for crawling tweets containing specified keywords, 70 million tweets have been crawled until end of 2017. To extract the artist and track out of #nowplaying tweets, one method is to compare the text of the tweets to entries in a reference database as MusicBrainz[4] using some similarity measure or regular expressions [128, 43, 112]. I.e., this

[2] `https://labrosa.ee.columbia.edu/millionsong/lastfm`, last visited November 26, 2017

[3] `http://www.cp.jku.at/datasets/LFM-1b/`, last visited November 26, 2017

[4] `https://musicbrainz.org`, last visited November 26, 2017

method was applied for creating the #nowplaying dataset as well as the Million Musical Tweet dataset. Besides this, also exploiting the subset of crawled tweets containing a Spotify URL (a URL leading to the website of the music streaming service Spotify) has been proposed by us in a previous work [85]. A typical tweet, published via Spotify, is depicted in Figure 4.1. We depict a tweet of the User Arlei Xavier, stating that he is listening to the country track Lay, Lady, Lay by Bob Dylan. Furthermore, we see a preview of the Spotify web player[5], which is an HTML site embedded in the tweet preview. In a previous analysis [85], we found, that although the URLs inside tweets are shortened to URLs like `https://t.co/kSFfZgYuUY`, Twitter also provides the resolved URL via their API. This allows identifying all Spotify-URLs by searching for all URLs containing the string "spotify.com" or "spoti.fi". By following the identified URLs, the artist and the track can be extracted from the title tag of the according website. For instance, the title of the website behind the tweet in Figure 4.1 is `<title>Spotify Web Player - Lay, Lady, Lay - Bob Dylan</title>`. In contrast to other contributions aiming at extracting music information from Twitter, where the tweet's content is used to extract artist and track, our approach enables an unambiguous resolution of the tweets [85]. Furthermore, our approach allowed us to link Spotify and Twitter users, which was valuable for the research we present in Chapter 5.

Figure 4.1: #nowplaying Spotify Tweets

[5] `https://play.spotify.com/track/4uYwlMp841PLJmj1gJJwIq?play=true&utm_source=open.spotify.com&utm_medium=open`, last visited November 26, 2017

4.5 Music Streaming Platform Spotify

Spotify is a European music streaming platform which started their service in 2006. Today, it offers 30 million tracks to 100 million registered users. We leverage a dataset which is amongst other sources based on Spotify users for (i) a prototype of a context-aware recommender system we implemented and published in 2015 [87], (ii) a large-scale playlist analysis we presented at the International Symposium on Multimedia in 2016 [88], (iii) a second prototype of a context-aware recommender system we published in 2017 [89], our research on music-cultural patterns presented at the International Symposium on Multimedia in 2017 [91] as well as for (iv) the music recommendation approach in this work. The used dataset combines #nowplaying tweets, playlists of Spotify users (along with the contained tracks) and the Echo Nest data providing the audio characteristics of the tracks. We give an overview of the creation and the dataset itself in the remainder of this section.

To get an initial list of users to crawl, we extracted the usernames of the users tweeting via Spotify from the #nowplaying dataset we introduced in the previous section. By querying the official Spotify API with this set of users, we could find 1,137 Spotify users organizing 796,024 distinct tracks in 18,296 playlists. A playlist is a set of tracks a user subsumed under a certain title. We refer to this title as playlist name. Tracks among the crawled playlists are an unordered list, as we do not have timestamps indicating at which time a track was added to the playlist nor the exact order in which a user added or listened to the tracks. As we see in Table 4.1, the first quantile, the median as well as the mean are much higher for the playlist dataset if we compare it to the #nowplaying dataset. Hence, we were able to smooth the sparsity in terms of listening events of the #nowplaying dataset as we successfully could increase the number of user-track interactions. We refer to these user-track interactions as listening events.

Dataset	Min.	1^{st} Qu.	Med.	Mean	3^{rd} Qu.	Max.
Playlist	1.0	88.0	334.0	702.8	790.0	186,196.0
#nowplaying	1.0	1.0	1.0	5.94	3.0	84,527.0

Table 4.1: Dataset Comparison: Number of Tracks per User

In a next step, we enlarged the dataset with the audio characteristics of the tracks. We crawled this information from the Echo Nest platform via their API[6]. The API allows querying for the audio characteristics of a track using a Spotify track identifier. Meanwhile, this step is not required, as the Echo Nest has been acquired by Spotify and the audio characteristics can be crawled directly via the Spotify API. Using this approach, we were able to retrieve audio

[6] `http://developer.echonest.com/docs/v4`, last visited January 25, 2016

characteristics in terms of acoustic features for more than 90% of the tracks. In particular, we retrieved the audio summary of all tracks encoded in seven acoustical features as provided by the Echo Nest. These features are danceability, energy, loudness, speechiness, acousticness, liveness and tempo. A detailed description of the acoustic features can be found online[7]. As common in the field of recommender systems, we refer to these audio features also as content-based features. Tracks for which we could not retrieve content-based features were removed from the final dataset, as our analyses require those features. The resulting dataset contains 1,133 Spotify users, 18,146 playlists, 706,989 tracks and for each of the tracks the seven acoustical features. On average, the dataset features 18,25 (SD=19.07) playlists and 1,084.07 (SD=2,659.45) tracks per user.

We refer to the final dataset as the "playlist dataset" in the remainder of this work. Compared to the #nowplaying dataset the playlist dataset is smaller with respect to the total numbers users, however, as we see in Table 4.1, the user profiles in terms of user-track interactions are in average 260 times longer. Along with that, the dataset includes content-based features allowing us to develop more advanced music recommender systems.

[7] `https://developer.spotify.com/web-api/get-audio-features/`, last visited November 26, 2017

CHAPTER 5

ELFC-MR: Ensemble Latent Feature Computation for Music Recommendation

5.1 Introduction

In this chapter, we introduce the reader to our ensemble latent feature computation for music recommendation (ELFC-MR) approach. This approach is an ensemble recommender system for track recommendations facilitating different machine learning techniques, to model a user's listening behavior on three user-centric dimensions: the *situational context*, the *musical characteristics* as well as the *cultural embedding* of a user. All three dimensions are used as input features for a factorization machine-based recommender system. We present our proposed recommendation model along with the evaluation of the model in this chapter and give the details of the recommender system's components, where each component process one user-centric dimension, in the respective Chapters 6, 7 and 8.

We highlighted in the previous chapter, that the recommendation of music and in particular, the recommendation of certain artists and tracks is nowadays more important than ever. This is, as due to the rise of the web, new distribution channels emerged. Besides online stores as iTunes[1], streaming platforms as Spotify[2], Deezer[3] or Apple Music[4] are continually attracting more and more users [65]. The MIR community observes that people switch from private, mostly limited music collections, to public music streaming collections containing several millions of tracks [65]. Hence, people increasingly do not store music locally on CDs and hard drives anymore. Instead, they access music they like via cloud-based streaming services using various mobile and stationary devices as smartphones, tablets or personal computers. Along with that, in contrast to traditional channels like the radio or TV, on those platforms, users can freely decide when they want to listen to which tracks. However, this wide variety of different music makes it difficult for the users to find music they like. In particular, it is impossible for the users to browse millions of tracks manually in order to find tracks they like and that simultaneously fit the current situation. Problems like this are known as information overflow and described by Resnick and Varian 1997 in their article about recommender systems [98]. For such a problem, recommender systems have been proven to be useful [39].

As discussed in the related work in Chapter 3, previous research [128, 86] utilized #nowplaying tweets as a publicly available data source. Along with that, additional (meta-) information about the artists and tracks provided by community driven databases as MusicBrainz[5] was leveraged. Besides music data, different types contextual information, for instance the geo-location [112, 114], the emotion and mood [41, 99, 11] or a combination of various factors depending on the domain [13], have been proven to be valuable to provide personalized recommendations. This context-awareness of music recommender systems leads to better recommendations, as a track should be to the favor of the user and simultaneously fit the current situation, also referred to as the current (listening) context of a user. This is, as the perceived usefulness of a track is *user-context-dependent*. The term user-context-dependent reflects that the perceived usefulness is influenced by the user, the context and finally by the interaction effect between a certain user and a certain context. As outlined in Chapter 3, with respect to those context-aware recommender systems, there

[1] `https://www.apple.com/at/itunes/`, last visited November 26, 2017

[2] `https://www.spotify.com`, last visited November 26, 2017

[3] `https://www.deezer.com`, last visited November 26, 2017

[4] `https://www.apple.com/music/`, last visited November 26, 2017

[5] `https://www.musicbrainz.org`, last visited November 26, 2017

are research gaps we address with our ensemble latent feature computation for music recommendation (ELFC-MR) approach. These research gaps are described in the remainder of this section.

Firstly, there is a lack of knowledge as only little is known about the music listening behavior of users in the cloud. In particular, little is known about the influence of different types of context on the music consumption. One of the reasons for this is that publicly available data regarding the current listening context is rarely available, however, is necessary for conducting research on context-aware recommendations models. In our analysis of users of the music streaming platform Spotify, we find several music listening patterns [86, 88]. We leverage these patterns for music recommendation using the situational context component described in Chapter 6 and our component extracting listening patterns based on audio features described in Chapter 7 [89]. Besides the current listening context and musical content, geographic information about the user in terms of GPS coordinates or countries and respectively continents has been used to approximate the cultural similarity between users [113, 114]. However, as political borders do not necessarily reflect cultural borders, a measure that integrates musical similarity and cultural similarity beyond countries' geographical borders is called for. We close this research gap by our proposed music-cultural similarity in Chapter 8, which is based on musical- as well as socio-economic features.

Finally, different types of contexts, i.e., the current situation along with the cultural embedding of a user including their interaction effects need to be jointly modeled in a context-aware music recommender system. Besides the direct effects and the interaction effects of different types of contexts, also the user-context interactions need to be modeled. This is, as the influence of a context towards the perceived usefulness of a track is *user-dependent*. Although quadratic interaction effects have been already discussed in the field of recommender systems to model context-context and user-context interaction effects, they were later neglected due to the complexity and data demands [14]. Often the available data is too sparse to fit the model. However, by leveraging factorization machines [92] we are able to estimate all the interactions effects even under high sparsity in linear time.

5.2 Component Overview

As already stated in the introductory part of this section, the ensemble latent feature computation for music recommendation (ELFC-MR) approach is an ensemble recommender system based on four major components. Three components are concerned with modeling a user's context of music consumption and rely on machine learning techniques to analyze the music consumption

behavior of the users. The fourth component is the component that predicts the perceived usefulness of a user towards a certain track in a certain context. This component is based on a factorization machine (FM) [92]. We provide a brief overview of all components and their interaction in this section and present the details of each component in a separate chapter.

In the workflow diagram in Figure 5.1, we depict the interaction of all components. The *acoustic clusters component* analyses user-generated playlists using matrix factorization and hence, in a latent feature space. Based on this latent feature analysis, the component extracts so-called "playlist archetypes", which are groups of tracks often found together in user created playlists as they share common audio characteristics. As along with matrix factorization, clustering techniques are used, we refer to these "playlist archetypes" also as acoustical clusters. The *situational clusters component* relies on classical text-mining algorithms known from the field of information retrieval to extract situational information from playlist names. The goal of this component is to group tracks that users are listening to in the same situational context using clustering techniques. Hence, we tag tracks with the intended use. The last contextual component is the *music-cultural clusters component*. This component mines for cultural listening patterns based on the geographic location of users, which is used to estimate music-cultural embedding of the user. Please note that we rely on clustering techniques to encode the k latent features computed in each of the three preprocessing steps into a single feature. This allows us to compute interaction effects between those features efficiently.

Before we present more details on each of the components in the Chapters 6, 7 and 8, we aggregate and present the main findings of our ELFC-MR approach in the next sections.

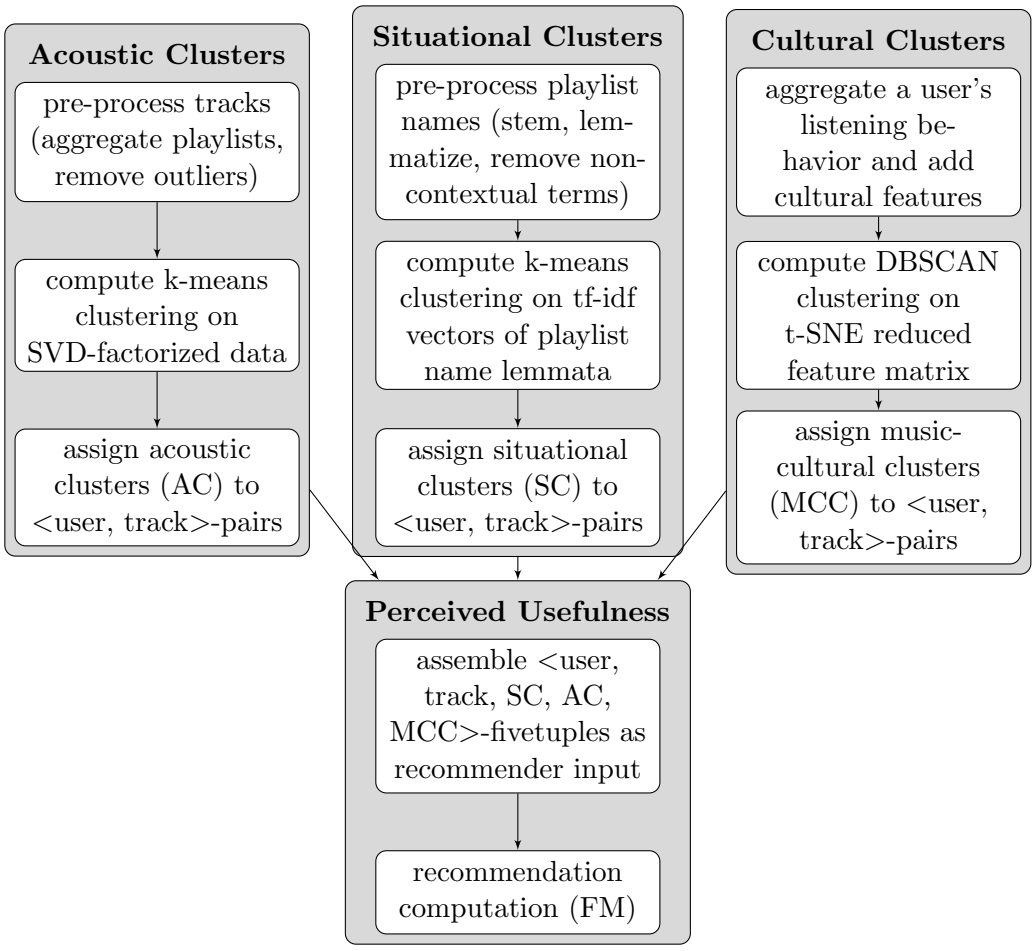

Figure 5.1: Workflow for Computing Recommendations

5.3 Proposed Recommendation Model

In several offline evaluations (c.f. Sections 6.5, 7.5 and 8.5) we can show that the most accurate recommendations are computed by a model incorporating situational clusters (SC), acoustic clusters (AC) and music-cultural clusters (MCC). Each cluster is computed by one of the three components presented in Figure 5.1. Please note that we condense the computed latent features of each component to a single feature by using clustering techniques, in order to compute the corresponding interaction effects. In Equation 5.1 we state the final model for estimating the predicted rating $\hat{r}$ of a user u towards a track i in a certain situation s.

$$\hat{r}_{u,i,s} = \mu + b_u + b_i + b_{sc} + b_{ac} + b_{mcc} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j \tag{5.1}$$

For estimating $\hat{r}$, our model considers a global bias μ, a user bias b_u, a track bias b_i, a situational cluster bias b_{sc}, an acoustical cluster bias b_{ac} as well as a music-cultural cluster bias b_{mcc}. Besides these individual biases, we moreover add all quadratic interaction effects $\sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j}$, where $n \in \{1, 2, 3, 4, 5\}$. As already elaborated on in Section 2.3.5, instead of learning all weights $w_{i,j} \in W \in \mathbb{R}^{n \times n}$ for the interaction effects, a FM relies on Cholesky decomposition to factorize the matrix W containing all weights $w_{i,j}$ (given a number of latent features k) into a diagonal matrix $V \in \mathbb{R}^{n \times k}$. This breaks the independence of the weights $w_{i,j}$ and allows to model the weights $\hat{w}_{i,j}$ as the inner product of the low dimensional vectors contained in V: $\hat{w}_{i,j} = \langle \vec{v_i}, \vec{v_j} \rangle$ [92]. To solve the factorization task, we rely on a Markov Chain Monte Carlo (MCMC) solver [102] as proposed by Freudenthaler et al. [37]. We apply a MCMC solver as this solver has been shown to perform efficiently and accurately for FMs [37, 93].

5.4 Component Performance

In Figure 5.2, we depict the F_1 recommendation performance[6] of our proposed recommendation model (SC+AC+MCC). To measure the effects of incorporating different contextual information encoded as clusters into our recommender system, we additionally state the performance of a model solely leveraging situational clusters (SC), a model leveraging situational clusters and acoustic clusters (SC+AC) and a model leveraging situational clusters along with music-cultural clusters (SC+MCC) in Figure 5.2.

[6] Details on the evaluations and the experimental setups are given in Section 6.5.3 and 6.5.4

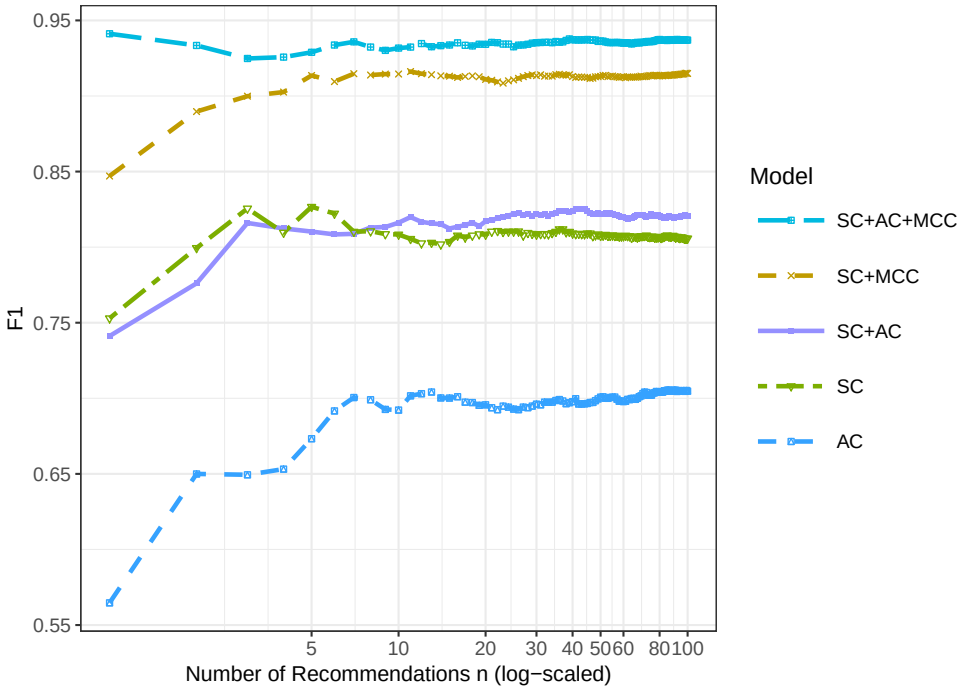

Figure 5.2: F_1 Curves

In the results in Figure 5.2 we observe that the best model is a model where situational-, acoustical- and music-cultural clusters are jointly leveraged (SC+AC+MCC). This model achieves an average F_1-score of 0.89 computed among all recommendations. This improves the F_1-score by 4.5% compared to the SC+MCC model which is the second best model. We detect a lower improvement by adding acoustical clusters, as parts of the content-based features are already captured in the MCCs. Finally, we observe that a model solely leveraging situational clusters (SC) exhibits a 19.53% lower F_1-score compared to the SC+AC+MCC model. Naturally, as we moreover see in Figure 5.2, a model that does not leverage situational context cannot compete with situational context-aware models. To conclude, we observe a substantial improvement by adding MCCs to the SC baseline and the best results are achieved by adding MCCs and ACs. Please note that we evaluate further baselines in Chapters 6,7 and 8. In particular, we evaluate random and non-model based most popular approaches as well as a set of different model-based approaches.

5.5 Summary

Throughout the course of this thesis, we substantially contributed to the field of music information retrieval. In this section, we shortly give an overview whereas more details are given in the corresponding Chapters 6, 7 and 8. To begin with, we introduce a novel method to extract the current listening context of a user from his or her playlist names [87]. Along with that, we propose a method to compute groups of playlists based on their audio characteristics. We refer to these groups as acoustical clusters or playlist archetypes, as these clusters represent distinct types of music listened by different users [88, 90]. We are able to successfully leverage both findings in a multi-context-aware recommender system. For this recommender system, we invent a novel user model that allows for taking into account which user enjoys listening to which type of music (acoustical clusters) in which situation (situational clusters) [89]. Finally, in Chapter 8, we present how to make this user model culture aware. We are able to show that by replacing the types of music or rather the acoustical clusters by our invented music-cultural clusters, recommendations can be improved substantially. For computing music-cultural clusters, we develop a novel model that captures the cultural music listing behaviors based on socio-economic factors and acoustical characteristics. We find that incorporating culture in isolation does not improve the recommendation accuracy, however, if leveraged in a joint user model additionally considering situational clusters, it is highly beneficial.

CHAPTER 6

ELFC-MR I: Situational Context[1]

6.1 Introduction

Music information retrieval and recommender systems researchers agree, that incorporating the user context into recommender systems results in more accurate recommendations [3]. In a qualitative study, Cunningham et al. [30] found that people organize music by the indented use in their music libraries. Along with that, the current activity of a user has been already successfully exploited to provide personalized music recommendations, i.e., based on the current location of a user during the day [126] or during driving a car [11]. However, no large datasets are publicly available. Hence, in a prior work, we proposed a novel approach for mining the situational context. We extract the

[1]This chapter is based on and content is partly reused from the following papers:
M. Pichl, E. Zangerle and G. Specht. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?. In Proceedings of the 15th IEEE International Conference on Data Mining Workshops (ICDM 2015), pages 1360-1365. IEEE, 2015.
M. Pichl, E. Zangerle and G. Specht. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In Proceedings of the 7th ACM International Conference on Multimedia Retrieval (ICMR 2017), pages 201-208. ACM, 2017.

current activity or the current occasion during the music is played from the playlist names of Spotify users. We presented this approach at the Social Media Retrieval and Analysis (SoMeRA) workshop co-located with the International Conference on Data Mining (ICDM) in 2015 [87]. Aiming at improving the recommendation accuracy of music recommender systems, in this prior work we analyzed how (i) the situational listening context can be extracted from playlists names of playlists created by users found on the music streaming platform Spotify and (ii) how to leverage this context to improve track recommendations. For the latter, we implemented a first prototype of a music recommender system based on pre-filtering collaborative filtering. Later, to show that modeling the interaction effect between users and contexts is beneficial, we published a model-based CF approach facilitating factorization machines at the International Conference on Multimedia Retrieval (ICMR) in 2017 [89]. In that work, we showed that our proposed method for extracting the current listening context of a user is also beneficial for state-of-the-art recommendation algorithms.

In this chapter, we aggregate the findings of both works in order to give the reader insights into to the mining- and model-based recommendation approach. For the presented model-based approach [89], we use the preliminary pre-filtering recommender system presented at the Social Media Retrieval and Analysis (SoMeRA) workshop [87] as a further baseline.

6.2 Research Overview

Aiming at improving the recommendation accuracy of music recommender systems and driven by the fact that people prefer different music in different situations [30], we conduct a study facilitating the listening histories and playlists of users of the music streaming platform Spotify. By leveraging the Spotify API, we are able to query the API for users, their playlists and the contained tracks. In the resulting dataset, which we already presented in Section 4.5, we observe that playlist names provide information about the situational context in which the contained tracks are listened to. Examples are “RUNNING”, “Work Playlist”, “Summer Fun”, “Dustin’s Workout Mix”, or “Christmastime”. This finding is congruent with the research by Cunningham et al. [30] and triggered the compilation of the following research questions:

RQ1 How can we extract contextual information hidden in the playlist names?

RQ2 How can we leverage the extracted context in a music recommender system?

To answer RQ1, we compute so-called "situational clusters" [87]. A situational cluster is a cluster that groups playlists that are listened in the same context. To give an example, the "my summer playlist", "summer 2015 tracks", "finally summer", "summer feelings" and "summer outside" playlists are clustered in the *summer cluster*. In short, we extract the common listening context or situation, which is hidden in the playlist name. We observe, that this common context is mainly either temporal, refers to occasions and events or refers to certain activities. By using clustering techniques, we are able to tag listening events, which are $<user, track>$-tuples in the dataset, with a certain contextual cluster and hence, a situational context. This allows us to apply a contextual pre-filtering algorithm as proposed in 2015 [87] and a model-based algorithm as proposed in this work and thus, to answer RQ2: we show that facilitating the situational clusters in a music recommender system helps to increase the recommendation quality. In several experiments based on k-fold cross-validation, we show that our proposed factorization machine-based recommender system utilizing situational clusters [89] outperforms context-agnostic recommender systems, pre-filtering context-aware recommender systems as well as other classifier-based context-aware recommender systems substantially in terms of precision, recall and in terms of the F_1-measure. Moreover, our experiments show that factorization machines are particularly capable of tackling the major limitation of our pre-filtering approach [87], that is to split dataset and hence, to split user-profile for the recommendation computation.

6.3 Analyzing Situational Music Listening Behavior

As mentioned in the research overview, we aim to extract and aggregate contextual information hidden in the playlist names to meaningful information about the user's music consumption behavior. In this section, we describe the process we propose in more detail.

As we want to group similar playlists based on their names using clustering techniques, we homogenize the playlist names in a first step by applying lemmatization. Lemmatization is a technique to find the lemma of a given word, which is the base form. For our study, we lemmatized the playlist corpus using WordNet [80], a well-known toolkit for natural language processing (NLP). Besides this homogenization, in a further step, we exclude playlists which do not contain any additional contextual information. These have been mainly playlists named after artists, albums or genres. For this filtering task, we rely on AlchemyAPI's entity recognition. AlchemyAPI is a commercial semantic text-analyzing tool owned by IBM. We use the AlchemyAPI, as prior experiments showed that its entity recognition works well with respect to tracks, artists and genres. With relation to that, our exper-

iments show that AlchemyAPI mainly relies on the MusicBrainz database[2], a community-driven database for music meta-information. As the remaining (cleaned) playlist names are rather short, for a better matching of similar playlists, we expand the bag of extracted lemmas for each playlist with synonyms and hypernyms using WordNet. This enables us to create a more expressive term frequency-inverse document frequency ($tf - idf$) matrix. The $tf - idf$ measure is depicted in Equation 2.3 and is applied to the bag of words describing each playlist. This bag of words is based on the derived lemmas, synonyms and hypernyms.

Finally, we aim to find groups of contextually similar playlists. For finding these contextual clusters in the $tf - idf$ matrix, we apply k-means clustering. As k-means requires the parameter k and hence, the number of clusters ex-ante as an input variable, we determine this parameter in the training phase of the recommender system. This is done by computing the within-cluster sum of squares (WCSS) as the quality measure assessing the clustering quality for $2 \leq k \leq 2 * \sqrt{\frac{n}{2}}$, where we denote n to be the number of distinct playlists. We apply this quality measure, as the k-means algorithm aims at minimizing the WCSS [72]. The upper limit of $2 * \sqrt{\frac{n}{2}}$ is based on a common approximation rule for estimating the number clusters k [74]. As we (i) do not recognize an elbow point in the WCSS-curve (cf. Figure 6.1), which is often used as indicator for the number of clusters k [17] and (ii) aim to determine k numerically, we invent an approach for determining a good k: Firstly, as we know that WCSS declines with the number of clusters, we compute the first order difference (ΔWCSS) to de-trend the WCSS curve [22]. This de-trended curve is shown in Figure 6.2. The de-trended WCSS curve can be interpreted as the decline/increase of WCSS if k is increased by 1.

In Figure 6.2, besides ΔWCSS, also the mean of ΔWCSS, which is subsequently referred to as $\overline{\Delta WCSS}$, is plotted. $\overline{\Delta WCSS}$ is represented by a solid line, whereas the dashed lines represent $\overline{\Delta WCSS} + / -$ the standard deviations, which we refer to as $\sigma_{\Delta WCSS}$. We determine k by choosing the largest k for which it holds that $\Delta WCSS < \overline{\Delta WCSS} - \sigma_{\Delta WCSS}$. We argue that if the reduction of WCSS is smaller than $\overline{WCSS} - \sigma_{WCSS}$, WCSS is not reduced significantly. Hence, the clustering quality is not increased significantly by increasing k. We underpinned this assumption by a two-sample Mann-Whitney U test [73], where the computed *p-value* is smaller than 0.01 indicating a highly significant difference.

By applying the presented procedure for mining for contextual information in playlist names on the Spotify dataset described in Section 4.5, we aggregate the contained playlists to 23 situational context clusters.

[2] `https://www.musicbrainz.org`, last visited November 26, 2017

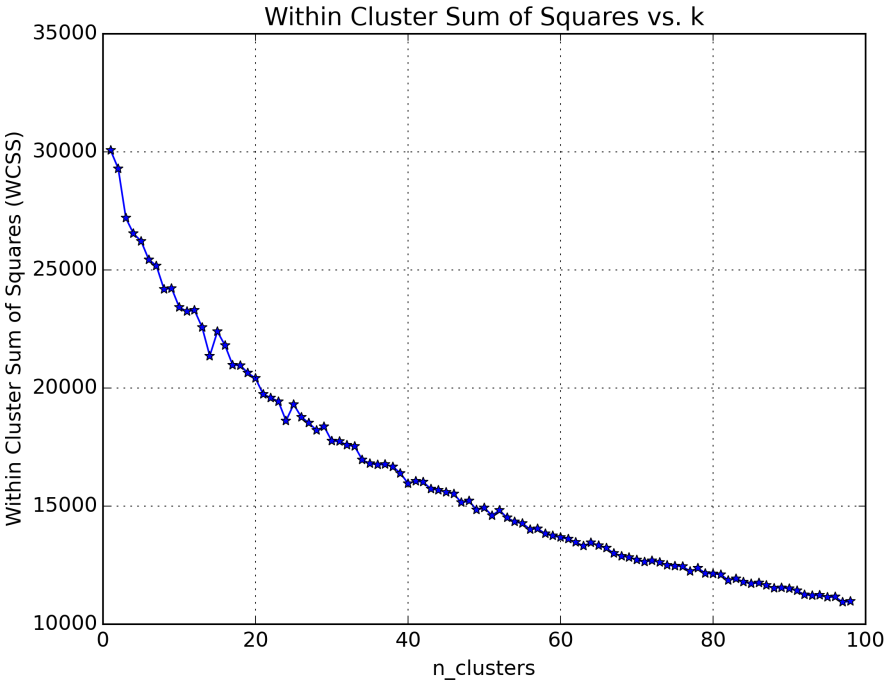

Figure 6.1: Within Clusters Sum of Squares (WCSS)

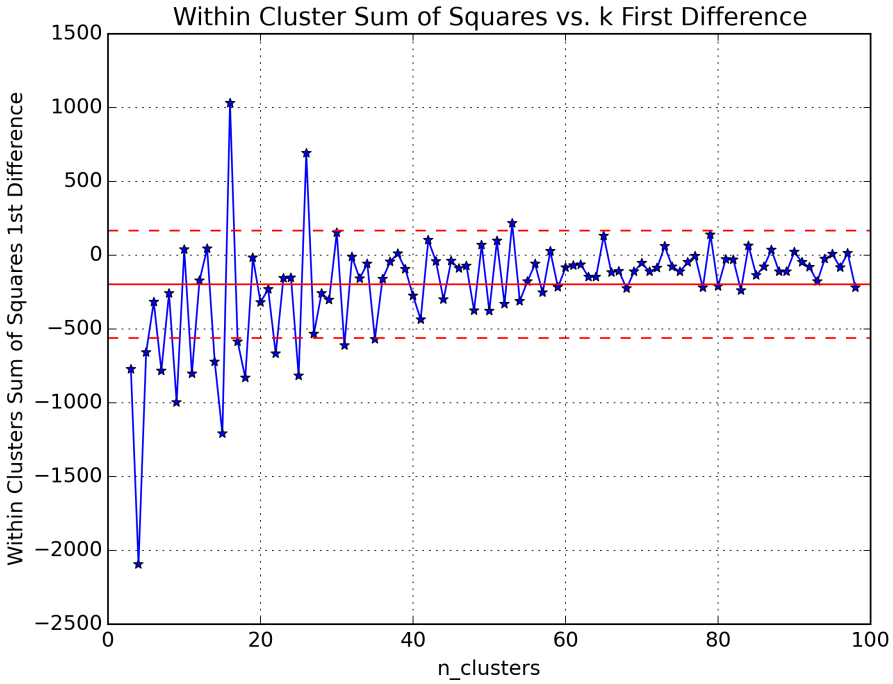

Figure 6.2: De-Trended Within Clusters Sum of Squares (ΔWCSS)

6.4 Recommender System Prototype

As outlined in the introduction, we propose a recommendation approach aiming to provide track recommendations for a given user in a given situational context (SC). Hence, we model the listening behavior of the users by the tracks they listened to and tag this information with the situational contexts in which the particular track was listened to. As the playlist dataset (cf. Section 4.5) does not contain explicit ratings, we assume that by adding a track to a playlist, the user expresses some preference for the track. For means of simplicity, we describe a user-track interaction extracted from a playlist as "a given user listened to a given track".

In a first step, we transform the $<user, track, playlist>$-triples of the playlist dataset to $<user, track, SC>$-triples, where SC represents the situational cluster in which the user listened to the track, by applying the clustering method presented in Section 6.3. Using this method, we assign each user-track pair with one of the 23 situational contexts in which the given user has listened to the given track. Adding a fourth factor *rating* to the dataset, allows us to model a recommender systems and in particular a rating prediction task: for each unique $<user, track, SC>$-triple, the *rating* $r_{u,i,s}$ is 1 if the user u has listened to the track i in situational context s. As the playlist dataset does not contain any implicit feedback by the users, for instance play counts, skipping behavior, session durations or dwell times during browsing the catalog, we can not estimate more detailed preferences of a user towards an item as proposed by Hu et al. [49]. Thus, for each $<user, track, SC>$-triple without interaction, we follow the approach by Pan et al. [83] and assume the rating to be $r = -1$. The rating $r_{u,i,s}$ for each user u, track i and situational cluster s can now be defined as stated in Equation 6.1.

$$r_{u,i,s} = \begin{cases} 1 & if\ u_u\ listened\ to\ t_i\ in\ SC_s \\ -1 & otherwise \end{cases} \tag{6.1}$$

As the distribution of relevant and irrelevant items is highly unbalanced in this dataset, as other works (i.e., [69]) we rely on oversampling as an efficient method [51] get nearly equally distributed classes and hence, an equal number of relevant and irrelevant items in the dataset. To foster a better understanding of the transformed dataset, we depict a sample of the dataset in Table 6.1. Leveraging this dataset, we train a classifier that decides whether a user has listened to a track in a contextual cluster or not. For the classifier and hence, the rating computation, we opt for a factorization machine (FM) [95, 93], as a FM is considered as a state-of-the-art classification approach [95]. As described in Section 2.3.5, FMs are a generalization of factorization models and

allow to model interactions of input variables in a lower-dimensional space (i.e., interactions are mapped onto a latent features-space of lower dimensionality). As we aim to exploit the interaction effects of users, tracks and clusters, we choose to utilize a FM of the order $d = 2$, modeling all single and pairwise interactions between the input variables. This model is as depicted in Equation 6.2, where $i \in \{1, 2, 3\}$. Please note that a bias b_i represents a user bias in case $i = 1$, a track bias in case $i = 2$ and the situational clusters bias in case $i = 3$.

$$\hat{r} = \mu + \sum_{i=1}^{n} w_i b_i \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j \tag{6.2}$$

Equation 6.2 can be rewritten to Equation 6.3, showing that our proposed FM-based model computes rating predictions by modeling a global bias (μ), the influence of the user ($b_u = w_1 b_1$), the influence of a track ($b_i = w_2 b_2$) as well as the influence of the situational clusters ($b_s = w_3 b_3$) along with the quadratic interaction effects of those ($\sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j$), where $n \in \{1, 2, 3\}$. However, as already elaborated on in Section 2.3.5, instead of learning all weights $w_{i,j} \in W \in \mathbb{R}^{n \times n}$ for the interaction effects, a FM relies on Cholesky decomposition to factorize W (given the number of latent features k) into a diagonal matrix $V \in \mathbb{R}^{n \times k}$. This breaks the independence of the weights $w_{i,j}$ and allows to model the weights $\hat{w}_{i,j}$ as the inner product of the low dimensional vectors contained in V: $\hat{w}_{i,j} = \langle \vec{v_i}, \vec{v_j} \rangle$ [92]. To solve the factorization task, we rely Markov Chain Monte Carlo (MCMC) solver [102] as proposed by Freudenthaler et al. [37]. We apply MCMC, as this solver has been proven to perform efficiently and accurately for FMs [37, 93].

$$\hat{r}_{u,i,s} = \mu + b_u + b_i + b_s \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j \tag{6.3}$$

To estimate the performance of the presented recommender system, we conduct a set of experiments as described in the following section.

6.5 Experiments

In this section, we introduce the experiments conducted to evaluate our FM-based recommender system. We start with a description of the data used for the evaluation before focusing on the experimental setup and the evaluation

measures. Please note that we use a similar evaluation procedure for the evaluations of our extensions of the ELFC approach in Chapters 7 and 8.

6.5.1 Data Modeling

For our experiments, we apply our proposed clustering method described in Section 6.3. Hence, as shown in Table 6.1, we reshape the input dataset into a dataset containing $<user, track, SC, rating>$-quadruples. We assigned each track in a playlist with a rating value to indicate whether a certain user listened to a certain track in a certain situational cluster encoded as a nominal variable ($r = 1$) or not ($r = -1$). A fragment of the dataset is shown in Table 6.1. This excerpt shows that user 872 has listened to track 250,246 in situational cluster 0, whereas user 911 has listened to track 250,246 in situational cluster 2. This dataset forms the foundation for our experiments, which are presented in the next section.

User	Track	SC	Rating
872	250,246	0	1
872	250,246	1	-1
911	250,246	2	1
911	250,246	0	-1
911	309,275	1	-1

Table 6.1: Data Set Fragment

6.5.2 Evaluated Recommender Systems

We compare our proposed FM approach to three baseline recommender systems: a system leveraging user-based collaborative filtering (UBCF), an SVD-based system and an alternative classifier-based system. To incorporate the contextual information into the UBCF- and SVD-based models, we apply pre-filtering [3]. Hence, if the UBCF or the SVD model is used, the recommendations are computed for each contextual cluster individually. I.e., we compute the recommendations on a subsample of the dataset restricted to a certain cluster. The classifier-based system uses the computed contextual clusters as an input feature to the classifier. Summing up, we benchmark classical CF approaches, approaches facilitating latent features (considered as state-of-the-art in recent years [2]) and a classification-based approach against our proposed factorization machine-based recommender system. We give an overview of all evaluated recommender systems in Table 6.2.

The most basic recommender system we benchmark is a user-based collaborative filtering approach [2]. The idea behind UBCF is to recommend items the k-nearest neighbors of a user interacted with. For determining the near-

Model	Context Aware	Classifier-based	Latent Features
UBCF			
PREF CF	✓		
SVD	✓		✓
PREF SVD	✓		✓
RF	✓	✓	
FM	✓	✓	✓

Table 6.2: Overview of Evaluated Models

est neighbors, we compute pairwise user similarities using the Jaccard Coefficient [50] of the set of tracks each of the two users listened to. Thus, we measure the number of commonly listened tracks in relation to the tracks both users listened to. The computation is depicted in Equation 6.4, where we denote S_i as the set of tracks a user i has listened to. Analogously, we denote S_j as the set of tracks a user j has listened to.

$$Jaccard_{i,j} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{6.4}$$

The second baseline recommender system we benchmark is a model-based approach leveraging singular value decomposition (SVD) [63]. SVD predicts ratings by extracting a pre-defined number of latent features from the user-item matrix R. In our setting, this is a sparse matrix containing all the binary ratings $r_{u,i}$ (cf. Equation 6.1) of all m users u and all the n tracks i they listened to. Such a rating matrix $R \in \mathbb{R}^{n \times m}$ can be decomposed into a user matrix $U \in \mathbb{R}^{n \times r}$, an item matrix $V \in \mathbb{R}^{r \times m}$ and a matrix $\Sigma \in \mathbb{R}^{r \times r}$ containing all singular values in descending order. This factorization is depicted in Equation 6.5.

$$R = U\Sigma V \tag{6.5}$$

To compute k latent features characterizing types of tracks, we firstly reduce the value r to k by selecting the top-k largest eigenvalues. Secondly, using stochastic gradient descent optimization (SGD) [63], we look for the closest approximation of R using k. We refer to this dimensionality reduced matrix as R_k. The approximation of R_k is depicted in Equation 6.6, where the matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{k \times m}$ are factor matrices containing the latent user- and item factors.

$$R_k \approx U_k \Sigma_k V_k \tag{6.6}$$

A final rating for a given user u and a given track i can be computed by using the corresponding user $\vec{u_u}$ and item vector $\vec{v_i}$ of U_k and V_k as shown in Equation 6.7.

$$\hat{r}_{u,i} = \vec{u_u} \cdot \vec{v_i} \tag{6.7}$$

Thirdly, we aim to compare our proposed approach with a classifier-based recommendation approach, as the performed recommendation computation can also be considered as a one-class classification problem [83] and random forest classifiers allow to estimate the probability of a user-item interaction [130, 4]. Therefore, we implement a random forest classifier [68] based system as it has two main advantages. Firstly, we only have to tune one parameter: the number of trees [82]. Secondly, all trees can be computed in parallel and the algorithm scales linearly with the number of trees.

Finally, we compare all recommender systems to a random-choice baseline. The assumption behind this baseline is that the fundamental chance of guessing whether a track was listened by a user ($r = 1$) or not ($r = -1$) is 50%. In particular, the chance of correctly guessing the correct rating in the sample space $\Omega = \{0, 1\}$ is $P(0) = P(1) = 0.5$ for each track. This is, as in our dataset both classes are equally distributed due to the oversampling (cf. Section 6.4). Hence, the random baseline for RMSE and MAPE is 0.5. The same holds for the precision measure, however, for the recall measure, we cannot state a single number. This is, as recall is dependent on the number of recommendations n and the number of relevant items $|T_r|$ in the test set. Assuming that the guessed rating of every second track is a correct guess, we correctly classify $\frac{1}{2}n$ tracks and recall would be $\frac{1}{2}\frac{n}{|T_r|}$.

6.5.3 Evaluation Measures

For assessing the rating prediction task, we compute two widely used error measures: the root mean square error (RMSE) as well as the mean absolute percentage error (MAPE) as stated in Equations 2.34 and 2.33. Using these error measures, we compare the predicted rating $\hat{r}$ to the actual rating r which is contained in the test set. For the results stated Table 6.4, we compute the average error among all ratings r_i in the test set. Please note that for computing the error measures, we scale the predicted rating $\hat{r}$ between 0 and 1 using min-max scaling to be able to directly compare all evaluated approaches.

For assessing the top-n recommendations task for different recommendation models, we compute the predicted rating $\hat{r}$ for each track in the current test set and sort the tracks by the predicted rating $\hat{r}$. Using the resulting top-n recommendations, we compute the *precision* and *recall* measures as described in Section 2.4.2. Furthermore, we combine both measures into the F_1-*score* in order to make different recommender system easily comparable based on a single measure. Besides the standard recall measure, we compute an *adapted recall*. This adapted recall reflects that in a setting as ours, computing the recall measure yields to a natural upper bound depending on the number of recommendations. As the number of relevant items in the test set $|T_r|$ is often larger as the number of recommendations n, the recall is bound to $\frac{n}{|T_r|}$. In contrast, to recall, the adapted recall only considers the first n relevant items in the test set ($|T_{nr}|$). This allows us to compute a recall without an upper bound. As the F_1-*score* is computed using precision and recall, we analogously compute an *adapted* F_1-*score*. In our experiments, we observe that the adapted recall is the preferred measure for our setting: As we sort the computed recommendations by the predicted rating $\hat{r}$ to evaluate the top-n tracks, the better an algorithm performs, the more relevant items where $\hat{r} = r = 1$ are contained in the top-n recommendations. This ultimately results in a higher number of relevant items in the test set and the best performing algorithms approach a recall of $\frac{n}{|T|}$ where we denote T to the size of the test set. Finally, especially if n is small, the top-algorithms can hardly be distinguished as we see in Figures 6.4 and 6.6.

$$Recall = \frac{TP}{|T_r|} \tag{6.8}$$

$$Adapted\ Recall = \frac{TP}{|T_{nr}|} \tag{6.9}$$

6.5.4 Experimental Setup

To evaluate the performance of the different recommender systems, we conduct a 5-fold cross-validation as presented in Section 2.4.1. Therefore, we randomly split the dataset into five folds of equal size. Subsequently, we utilize four folds as training data and the remaining fold as test data. This process is repeated 5 times such that every fold serves as test data once. Due to the random selection of data of the folds, each fold contains an arbitrary number of relevant and irrelevant items. However, due to the underlying distribution, the number of relevant and irrelevant items is approximately the same. The relevant items are tracks a user has listened to within a certain situational

cluster and analogously irrelevant items are items a user did not listen to within a certain cluster.

For assessing the rating prediction performance of the different recommender systems, we compute the predicted rating $\hat{r}$ for each track in the current test set. Using the min-max scaled predicted ratings $\hat{r}$ as well as the actual ratings r in the test set, we compute the evaluation measures as described in Section 2.4.2. We compute these evaluation measures for each fold separately. For the final results in Section 6.5.5, we compute the average across all folds. For evaluating the top-n recommendations performance, we sort the resulting set of recommendations by the predicted rating $\hat{r}$ and subsequently use the top-n recommended tracks for the evaluation. We compare $\hat{r}$ to the actual rating r for the current user, track and cluster in the test set. For this comparison, we assume all track recommendations with $\hat{r} \geq 0.5$ as relevant for the user in the given context and hence, $\hat{r} = 1$.

As for the learning method utilized for the FM, we make use of Markov Chain Monte Carlo (MCMC) inference as proposed by Rendle et al. [93]. Generally, we tune each of the recommender systems (except the random baseline), using k-fold cross-validation. For the random forest classifier, we train the random forest classifier with 500 trees. In preliminary experiments, we found that this is a sufficient number of trees to get stable results. Similarly, in our preliminary experiments we found that for UBCF $n = 50$ and for SVD $k = 10$ are suitable parameter options.

6.5.5 Experimental Results

As listed in Table 6.2, in our experiments we assess the performance of the following recommender systems: a pure user-based CF recommender system (UBCF), a context-aware recommender system based on UBCF with pre-filtering similar to the prototype we proposed in 2015 [87] (PREF CF), a context-agnostic SVD-based recommender system (SVD), a context-aware SVD-based recommender system with pre-filtering (PREF SVD), a context-aware random forest classifier-based recommender system (RF) as well as our proposed context-aware FM-based recommender system (FM). As presented in the preceding Sections 6.5.3 and 6.5.4, we evaluate the rating prediction task using the RMSE and MAPE and the top-n recommendations task using precision, recall and the resulting F_1-score.

For presenting the results of the top-n recommendations task, we depict the precision-, recall- and adapted recall curves for $n = \{1 \dots 100\}$ in Figures 6.3, 6.4 and 6.5. Aiming at making the performance of the recommender systems easily comparable, we plot the F_1- and the adapted F_1-score in Fig-

ures 6.6 and 6.7. The precision plot (c.f. Figure 6.3) shows that FM- and RF-based approaches outperform all other approaches substantially. On average, the FM-based approach yields to 22.80% higher precision values compared to the RF approach and to 169.23% higher precision values compared to the pre-filtering CF approach. Prefiltering CF achieves the highest precision values among the non-classifier-based approaches. Notably, the non-classifier-based approaches perform worse than the random baseline across all n. Furthermore, we observe a bad SVD performance. We lead this back to the boolean ratings, where SVD cannot fully exploit its strengths.

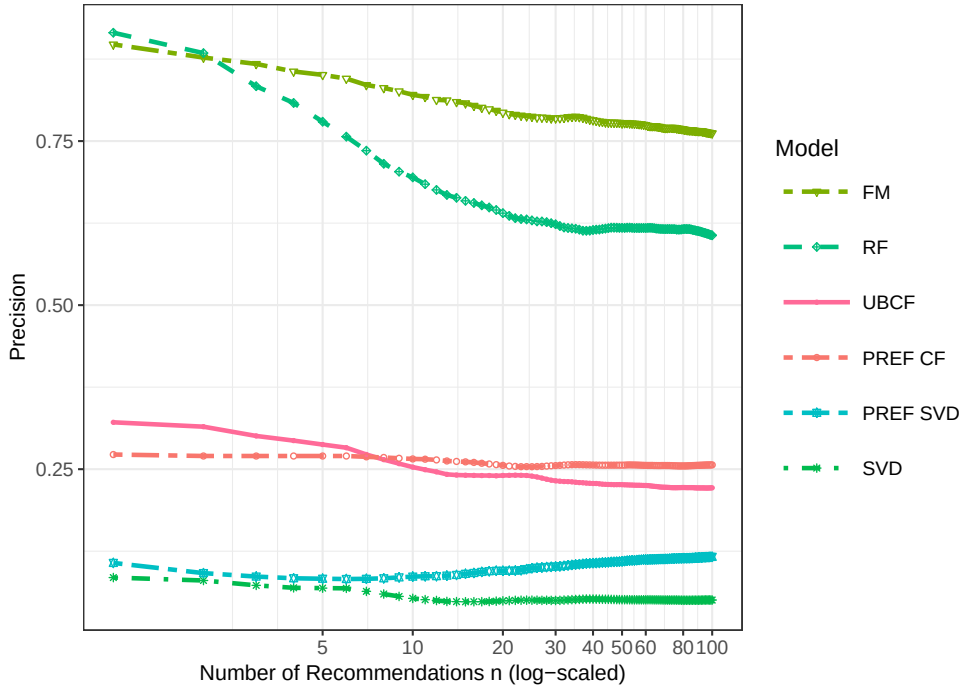

Figure 6.3: Precision Curves

As for both recall computations (Figures 6.4 and 6.5), we detect a similar pattern: Again, the classifier-based approaches exhibit substantially higher recall values than the other approaches. In particular, the adapted recall of the classifier-based approaches is 193% higher compared to UBCF. Besides that, in Figure 6.4, we observe that pre-filtering is solely beneficial for precision, but has a negative impact on the recall values. We suspect two reasons for this: Firstly, pre-filtering computes recommendations based on parts of the dataset. This is beneficial for the precision, as the number of recommendation candidates is limited. However, this configuration naturally limits the recall. Secondly, as we compute user similarities on a restricted amount of data, similarities are computed on less data which also possibly limits the set of possible recommendations. Besides that, we observe that for small n, the

top-performing algorithms (RF and FM) approach a recall of $n/100$. Hence, they can be hardly distinguished. This is, as the recommender systems sort the recommendation candidates by the predicted rating $\hat{r}$ and cut off at n. Thus, the better an algorithm performs, the more relevant items where $\hat{r} = r = 1$ are suggested. This ultimately results in a higher number of relevant items in the test set as well as an enumerator similar to the number of recommendations n. To solve this issue, we compute an adapted recall as stated in Equation 6.9. This adapted recall only considers the relevant items among the first n items in the test set in contrast to the classical recall computation where all relevant items in the test set are considered. With this method, we avoid the situation where the number of relevant items is greater than the number of recommendations n. Similar to the F_1-score, when examining the adapted F_1-score results in Figure 6.7, we observe that the FM- and RF-based approaches consequently outperform the baseline approaches.

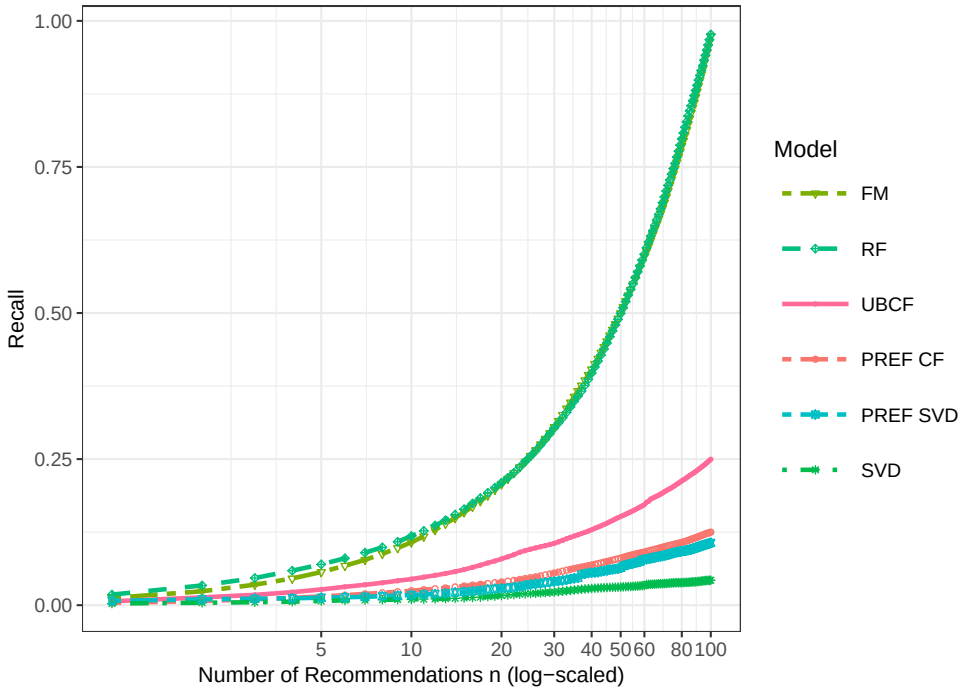

Figure 6.4: Recall Curves

Finally, the precision and F_1-scores in Table 6.3 show that our proposed FM-based approach clearly outperforms the RF-based approach. We observe that an RF-based approach exhibits similar recall values, however, for assessing the user satisfaction, a short list of recommendations is important (naturally assuming that the recommendation list contains a sufficient number of relevant items) [19]. Hence, from the perspective of maximizing user satisfaction, the FM approach outperforms the RF approach, although the RF approach similar

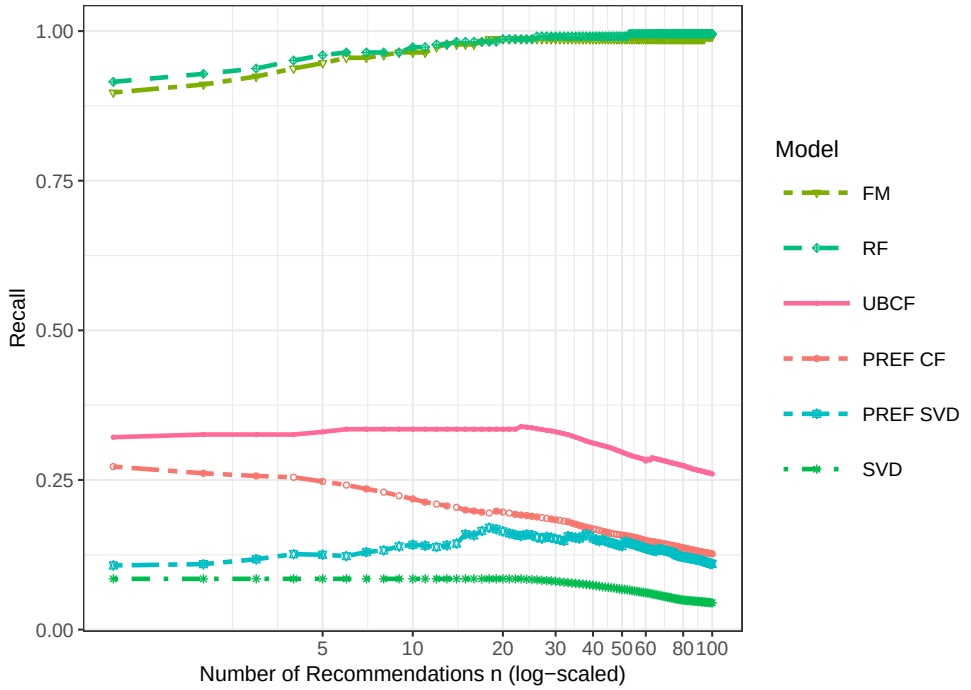

Figure 6.5: Adapted Recall Curves

to the FM approach models the situational context explicitly. However, a RF lacks the pair-wise interactions ($\sum_{i=1}^{n}\sum_{j=i+1}^{n}\hat{w}_{i,j}b_i b_j$) as depicted in the FM model in Equation 6.3. This is why we conclude that a hybrid approach combining regression with two-way interaction effects, where the weights of these effects are estimated via matrix factorization for classification (as provided by a factorization machine) is the best approach for situational context-aware music recommendation in a setting similar to ours.

Model	**Precision**	**Recall**	**Adapted Recall**	**Adapted F_1**
FM	**0.70**	**0.51**	**0.89**	**0.78**
RF	0.57	**0.51**	0.88	0.69
UBCF	0.23	0.15	0.30	0.26
PR-CF	0.26	0.08	0.17	0.21
PR-SVD	0.10	0.06	0.14	0.12
SVD	0.05	0.03	0.07	0.06

Table 6.3: Top-n Recommendations Performance@100 ordered by F_1

Complementary to the evaluation of the find good items task, we compute the rating prediction errors which is very common in the field of recommender systems. The results are stated in Table 6.4. By computing RMSE and MAPE over all recommendations, we can validate the results of the evaluation

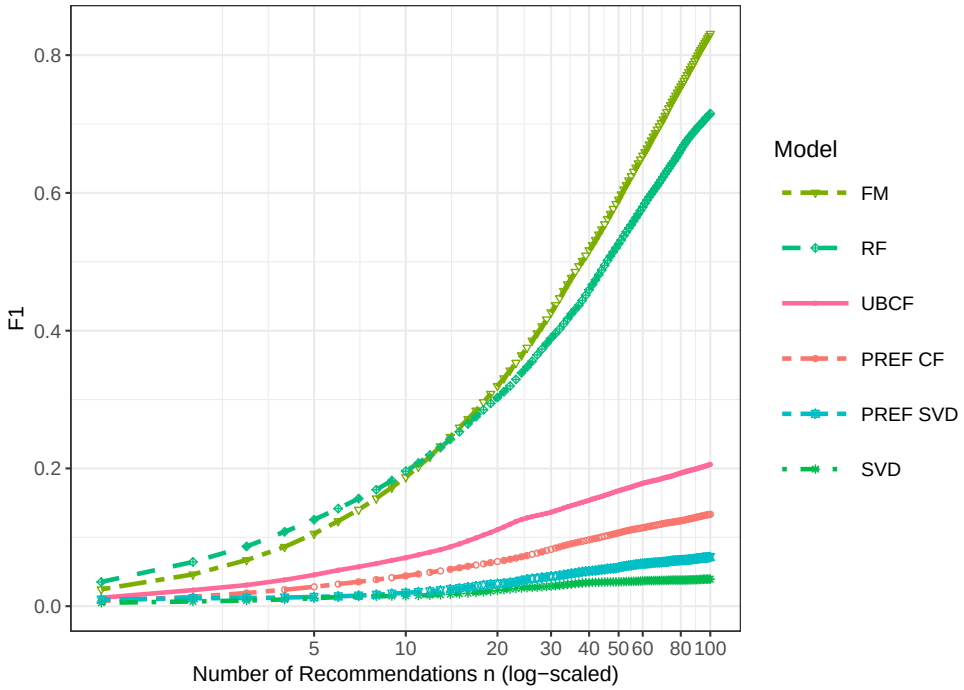

Figure 6.6: F_1 Curves

of the top-n recommendations task: the classifier-based approaches substantially outperform all other approaches. Furthermore, the FM-based recommender outperforms the RF-based baseline by 18.18% in terms of RMSE and by 30.78% in terms of MAPE. Besides that, we observe that the FM-based approach is the only approach that performs better than the random baseline with respect to the RMSE. This is in contrast to MAPE, where only the Pre-filtering CF approach misses the random baseline. This is, as the RMSE is more sensitive to outliers as discussed in Section 2.4.2.

Recommender	**RMSE**	**MAPE**
FM	**0.45**	**0.27**
RF	0.55	0.39
CF	0.76	0.34
SVD	0.77	0.37
Pre-filtering CF	0.81	0.51
Pre-filtering SVD	0.82	0.37

Table 6.4: Rating Prediction Measures ordered by RMSE

Summing up, we show that our proposed FM-based recommender system clearly outperforms all other approaches in the rating prediction task and hence, in terms of the RMSE and MAPE. Furthermore, with respect to the

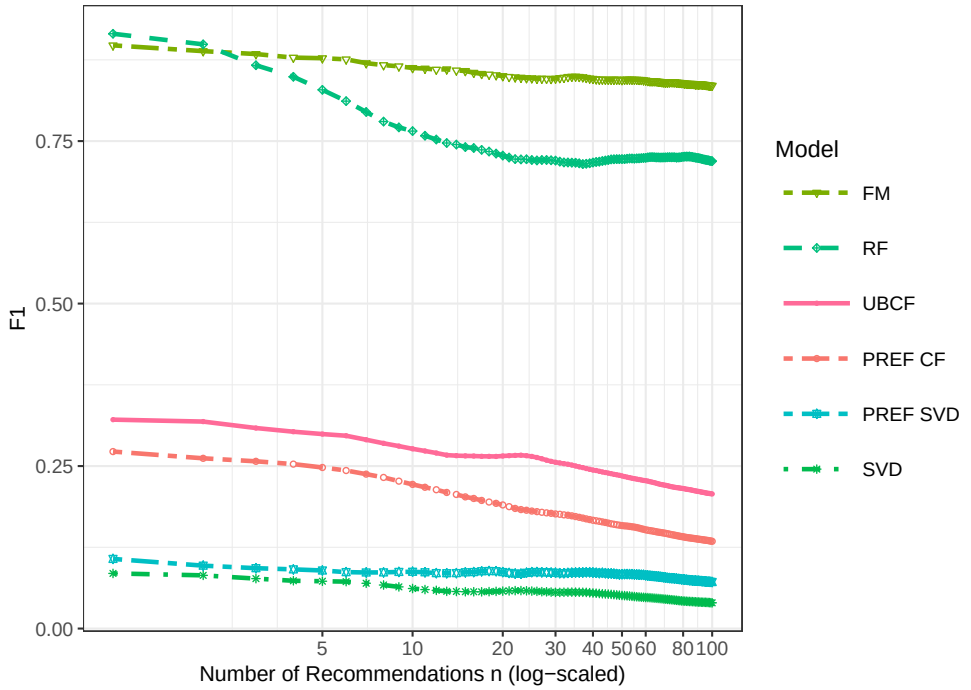

Figure 6.7: Adapted F_1 Curves

top-n recommendations task, the FM-based approach outperforms all other approaches with respect to the precision and the F_1-score, whereas the recall is similar to the RF-based approach, the second-best approach. We lead this behavior back to the recall computation. For each algorithm, the tracks are ordered by the predicted rating $\hat{r}$ and hence, by the likelihood of being relevant to a given user in a given context. Secondly, there is a natural upper bound of the recall dependent on the number of recommendations ($\frac{n}{|T_r|}$). As we sort the recommendations by the predicted rating $\hat{r}$ to evaluate the top-n tracks, the order of tracks is essential. The better an algorithm performs, the more relevant items with $\hat{r} = r = 1$ are sorted into the top-n recommendations. This ultimately results in a higher number of relevant items in the test set ($|T_r|$) and is the reason why the top-algorithms approach a recall of $\frac{n}{|T|}$ which is in our setting $\frac{100}{|T|}$. To avoid this problem, we compute an adapted recall as stated in Equation 6.9. By applying this recall measure, we observe differences between the RF- and FM-based approach to the favor of the FM-based approach. Besides that, according to Bollen et al. [19], for maximizing the user satisfaction a recommender system should state a short list of recommendations. Hence, precision is the more important measure for our use case. Referring to our second research question, namely, how to leverage the extracted context in a music recommender system, we find that situational clusters can be leveraged for music recommendations without the drawbacks of the pre-filtering

approach by using a classifier approach. Our experiments show a substantial improvement of the recommendation performance by leveraging situational clusters using classifiers. Particularly, we find that by using Factorization Machines, the best results with respect to the top-n recommendations and rating prediction tasks is obtained.

6.6 Summary and Contribution

The main contribution to the field music information retrieval presented in this chapter is three threefold. Firstly, we propose a method to extract and aggregate contextual information out of playlist names [87]. Secondly, we invent a numerical method for determining the number of situational clusters based on the playlist names [87]. Thirdly, we implement a recommender system prototype based on a factorization machine [89] and benchmark this prototype against a set of baseline approaches. In these experiments, we can show that (i) situational context aware-recommendation models outperform context-agnostic models and that (ii) our FM-based recommender system outperforms other context-aware recommendation models as it delivers the most precise recommendations. According to Bollen et al. [19], for maximizing the user satisfaction, short lists of recommendations are important. Hence, our FM-based approach maximizing the precision also maximizes the user satisfaction.

CHAPTER 7

ELFC-MR II: Music Characteristics[1]

7.1 Introduction

Due to the success of our situational context-aware collaborative filtering recommender system [87, 89], a logical next step is to study whether we can further improve our recommendations by facilitating content-based musical features. Celma [26] found, that collaborative filtering-based recommender systems do not necessarily exploit the long tail and content-based systems, that exploit the long tail, empirically do not perform good enough. Along with that, prior studies found that people prefer *different* types of music and thus, create different playlists biased to a certain type of music [119]. Triggered by these prior works, we amplify our context-aware recommender sys-

[1]This chapter is based on and content is partly reused from the following papers:
M. Pichl, E. Zangerle and G. Specht. Understanding Playlist Creation on Music Streaming Platforms. In Proceedings of the 18th IEEE Symposium on Multimedia (ISM 2016), pages 475-480. IEEE, 2016.
M. Pichl, E. Zangerle and G. Specht: Understanding User-Curated Playlists on Spotify: A Machine Learning Approach. International Journal of Multimedia Data Engineering and Management (IJMDEM). 8(4), 2017.

tem. This amplified recommender system is capable of modeling that users categorize tracks in their music libraries after the intended use [31, 54] and that users prefer different types of music [119] in different situations. Hence, our multi-context-aware recommender system is able to leverage the situational clusters modeling the intended use and additionally exploits audio features to differentiate between different types of music.

In a first step, we conduct a study aiming at getting a deeper understanding of the musical characteristics of user-generated playlists. We argue that understanding how users create and maintain their playlists can naturally contribute to more personalized and thus, more accurate recommendations. The results of the study have been published in the full paper "Understanding Playlist Creation on Music Streaming Platforms" presented at the IEEE International Symposium on Multimedia 2016 [88] and an extended version has been published in the International Journal of Multimedia Data Engineering and Management [90]. As we observe music listening patterns that can be potentially leveraged in music recommender systems, in this work we are particularly interested to model that users tend to listen to a certain type of music in a certain situation.

7.2 Research Overview

In contrast to the well-studied field of automatic playlist generation, studies about the characteristics of playlists created by human users on music streaming platforms hardly existed in 2016. Hence, we conduct a study to deepen the understanding for user-curated playlist on the music streaming platform Spotify. In contrast to the field of automatic playlist generation, we shift the focus from automatic playlist generation to the analysis of playlists. To conduct this study, we require a dataset containing information about users and their playlists. We already presented this dataset in Section 4.5. In total, we base our analysis on 1,133 users and their 18,146 playlists. The analysis is particularly driven by research questions RQ1 and RQ2. We focus on the third RQ in Section 7.4, where we present a recommender system that is capable of exploiting the interaction effects between a certain user, a certain situational cluster and his or her musical preferences in this cluster. Thus, we model that users tend to listen to a certain type of music in a certain situation.

RQ1 How do users organize their music in the music streaming era?

RQ2 How can we observe and explain acoustical differences between playlists?

RQ3 How can we leverage the gained knowledge in a multi-context-aware music recommender system?

With respect to RQ1 and RQ2, by analyzing the playlist generation behavior of Spotify users [88] using a principal component analysis (PCA) with singular value decomposition (SVD) as the computational kernel (cf. Section 7.3.2), we are able to explain differences using content-based music features. When clustering playlists into five clusters according to their musical features, we observe that on average each user creates playlists within three different clusters and 17% of all users create playlists in all five clusters. This finding suggests that users arrange different styles of music in different playlists. Complementary to that, we find that although nearly half of the users create playlists with classical and rap-style music, these playlists account only for 8 and 7% of all playlists, respectively. Moreover, we detect a cluster where 91% of all users create playlists in that contains a form of "feel-good" popular music, serving as a common musical ground across all users. Furthermore, our analysis shows that people do not necessarily group their music by genre and hence, our clusters cannot be replaced by using the musical genre. We consider the insights gained in this work to be useful for improved automatic playlist generation and music organization using recommender system. For the latter, we implemented a recommender system leveraging the newly gained insights and show the system's superior performance in several offline experiments.

7.3 Analyzing Music Listening behavior on Spotify

We describe our proposed procedure for analyzing the music listening behavior of Spotify users in the following sections. To begin with, we give a brief overview: In a first step, we aggregate the acoustical features of each individual track in a playlist to a single vector characterizing a whole playlist. These acoustic features and are provided directly via the Spotify API and are features that are extracted and aggregated from the audio signal of a track. The features comprise: *danceability* (how suitable a track is for dancing), *energy* (perceived intensity and activity), *speechiness* (presence of spoken words in a track), *acousticness* (confidence whether track is acoustic), *instrumentalness* (prediction whether track contains no vocals), *tempo* (in beats per minute) and *valence* (musical positiveness conveyed). A detailed description of these features and the API can be found on the Spotify Website[2]. These audio features provided via the Spotify API have already been exploited by a number of other analyses (e.g., [27, 117, 38]).

[2] `https://developer.spotify.com/web-api/get-several-audio-features`, last visited November 29, 2017

In a second step, aiming at finding the latent features that explain most of the variance in the dataset, we conduct a PCA using SVD as the computational kernel on the matrix containing all playlist vectors as described in Section 7.3.2. Hence, we mine for differences in the user-generated playlists with respect to their acoustic characteristics, in order to answer RQ1. In a third step, we make use of k-means clustering to aggregate playlists into groups (or types). The clustering is performed in order to answer RQ2 and hence, to find certain types of playlists. To find user types creating such playlists, we rely on several correlation and similarity measures which allow us to observe which users create certain playlists in certain clusters. We elaborate on all steps in more details in the forthcoming sections.

7.3.1 Data Cleaning and Aggregation

As we aim to get a deeper understanding of music playlists, we have to filter for musical tracks within our dataset. Thus, we restrict the playlist dataset to tracks with a *speechiness* of 0.66 or below. According to the Spotify documentation, tracks with a *speechiness* higher than 0.66 are most likely audio books[3]. To analyze the acoustic features of each playlist, we aggregate the acoustic features of the individual tracks for each playlist in the dataset using the arithmetic mean. To show the dispersion of the tracks forming a playlist, we state the mean as well as the mean absolute deviation (MAD) of each acoustic attribute in Table 7.1. We make use of the MAD as it is a robust measure with respect to outliers [67]. Table 7.1 shows that except for loudness, the variance of each of the acoustic characteristics of the tracks inside a playlist is low: We observe a MAD that is rarely higher than the mean. Thus, we conclude that aggregating the characteristics of the individual tracks to playlist characteristics using the mean is representative. Further, we argue that aggregating the loudness of the individual tracks to a playlist loudness is not reasonable: the variance among the loudness in the tracks of a playlist is too high. In 99.99% of all cases, the MAD is higher than the mean. Therefore, we drop the loudness characteristic for the conducted playlist analysis. Finally, we derive a matrix P where each row represents a playlist. Hence, each playlist is represented by a 7-dimensional feature vector $\vec{p}$ containing the features *tempo*, *energy*, *speechiness*, *acousticness*, *danceability*, *loudness*, *valence* and *instrumentalness*.

[3] `https://developer.spotify.com/web-api/get-audio-features/`, last visited November 29, 2017

Attribute	MAD >Mean	%
tempo	0	0.00%
energy	61	0.34%
speechiness	39	0.21%
acousticness	1,392	7.67%
danceability	2	0.01%
loudness	18,145	99.99%
valence	101	0.56%
instrumentalness	978	5.39%

Table 7.1: Aggregated Acoustic Features

7.3.2 Groups of Playlists

As pointed out in the beginning of this section, in order to analyze user-generated playlists, we conduct a PCA using SVD as the computational kernel on the playlist matrix P. Figure 7.1 depicts a biplot of the first two Principal Components (PCs), where each playlist is represented by a dot. This allows us to analyze 60.7% of the variation within the playlists dataset. The first PC on the x-axis distinguishes *acoustic* and *instrumental* playlists from playlists focusing on *tempo* and *energy* as well as playlists focusing on *valence* and *danceability*. This is, as the loading vector of PC1 only has negative signs for *acousticness* and *instrumentalness* and thus, contrasts those two attributes from the other attributes. By only using the first PC, we are able to explain 41.2% of the variation. Analogously, we observe that the second PC on the y-axis divides more *instrumental* playlists and playlists with high *tempo* and *energy* from playlists that are more *acoustic* as well as playlists with high *danceability*, *valence* and *speechiness* values. Again, this is as the loading vector of PC2 has negative signs for latter attributes, whereas the former three attributes are positively signed. By using the second PC, we are able to explain another 19.5% of the variation. We complement our analysis by looking at PC3, explaining 13.4%: PC3 separates tracks with high *speechiness* values from the remaining playlists. Using the first 3 PCs, we are able to explain 74.1% of the variance. Adding further PCs adds 10% and less explained variance and we observe a gap in the variance curve after PC3.

Based on the findings of the conducted PCA, we aim to partition our set of playlists into a set of playlist clusters: *instrumental* and *acoustic* playlists, playlists focusing on *valence* and *danceability* along with *speechiness* and playlist focusing on *tempo* and *energy*. Hence, we apply k-means clustering with $k = 3$ to $k = 7$. Clustering into 3 clusters yields clusters that are based on the first two PCs (as described above), whereas clustering into 7 clusters yields clusters based on each of to the 7 acoustical features. This is shown

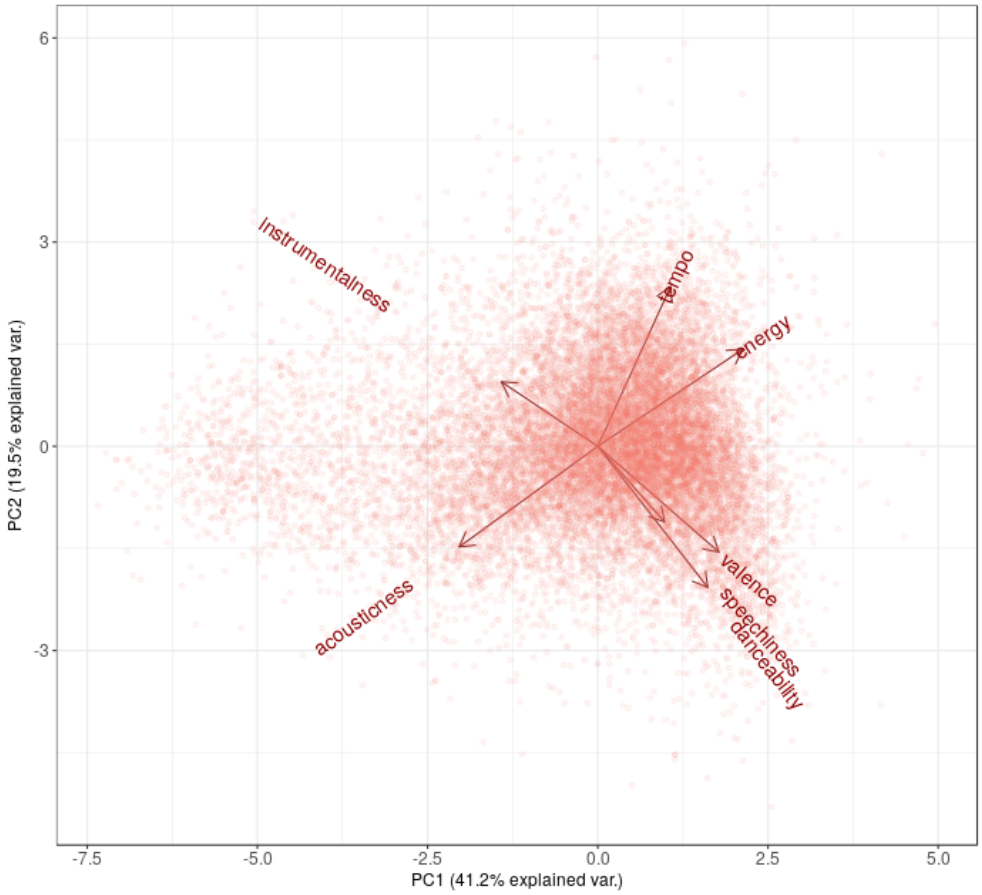

Figure 7.1: Biplot using PC1 and PC2

in Figure 7.2, where different k-means solutions are plotted for different k. Each point represents a playlist, plotted against PC1 and PC2. The color and shape of the points represent the cluster membership. We formally determine the optimal number of clusters utilizing the gap statistic [122]. This method is based on the "Elbow Curve"[17] or rather, on the idea how much the within-cluster sum of squares (WCSS) decreases with an increasing number of clusters, as the WCSS naturally decreases with the number of clusters. Utilizing the presented approach, the gap statistic indicated that 5 clusters are an appropriate solution. In a next step, we aim to get an overview of the acoustical attribute characteristics of the five clusters. Therefore, we visualize these as radar diagrams, one diagram for each cluster as shown in Figure 7.3. These diagrams show the different features and their manifestation in the five clusters.

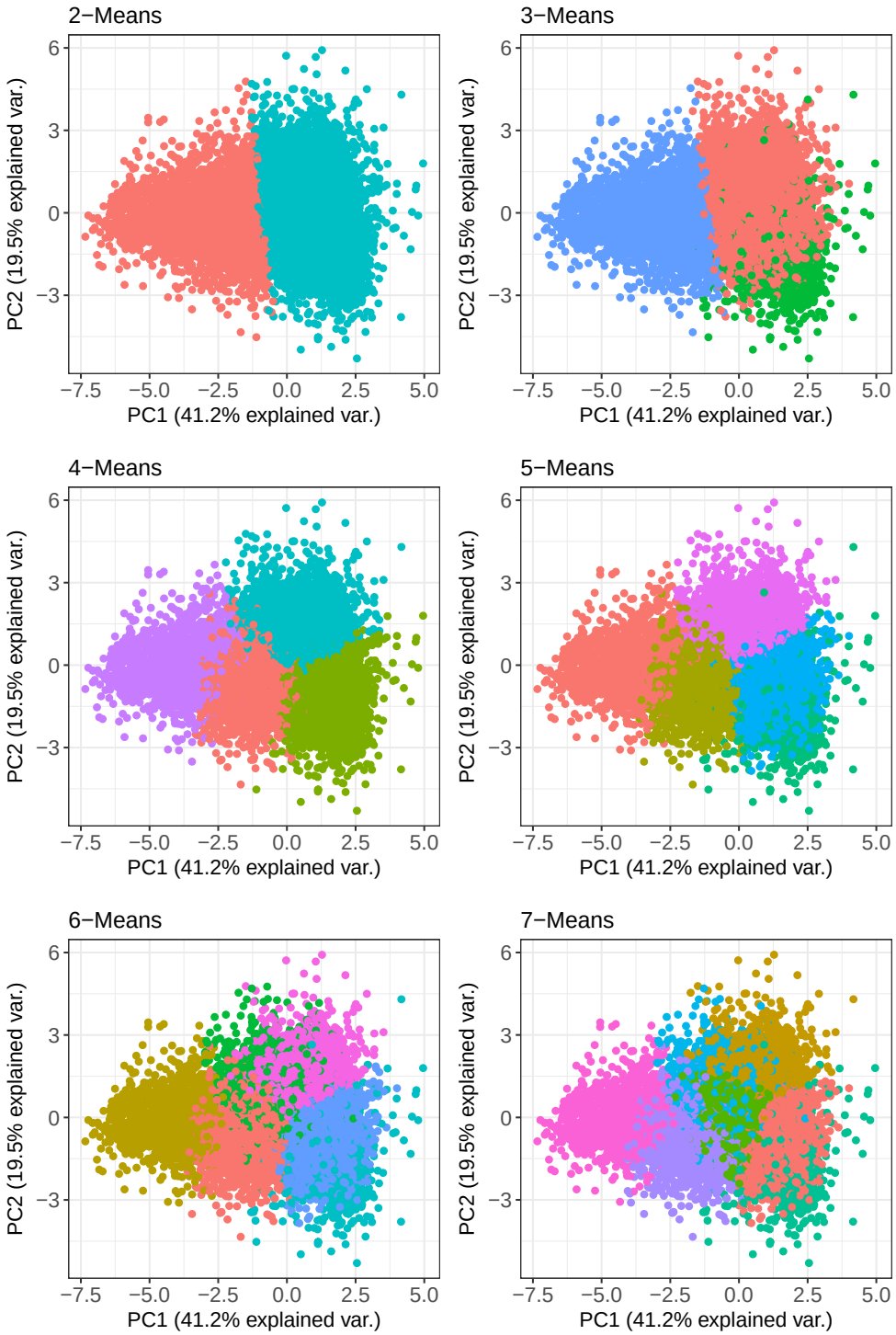

Figure 7.2: k-means for k between 2 and 7

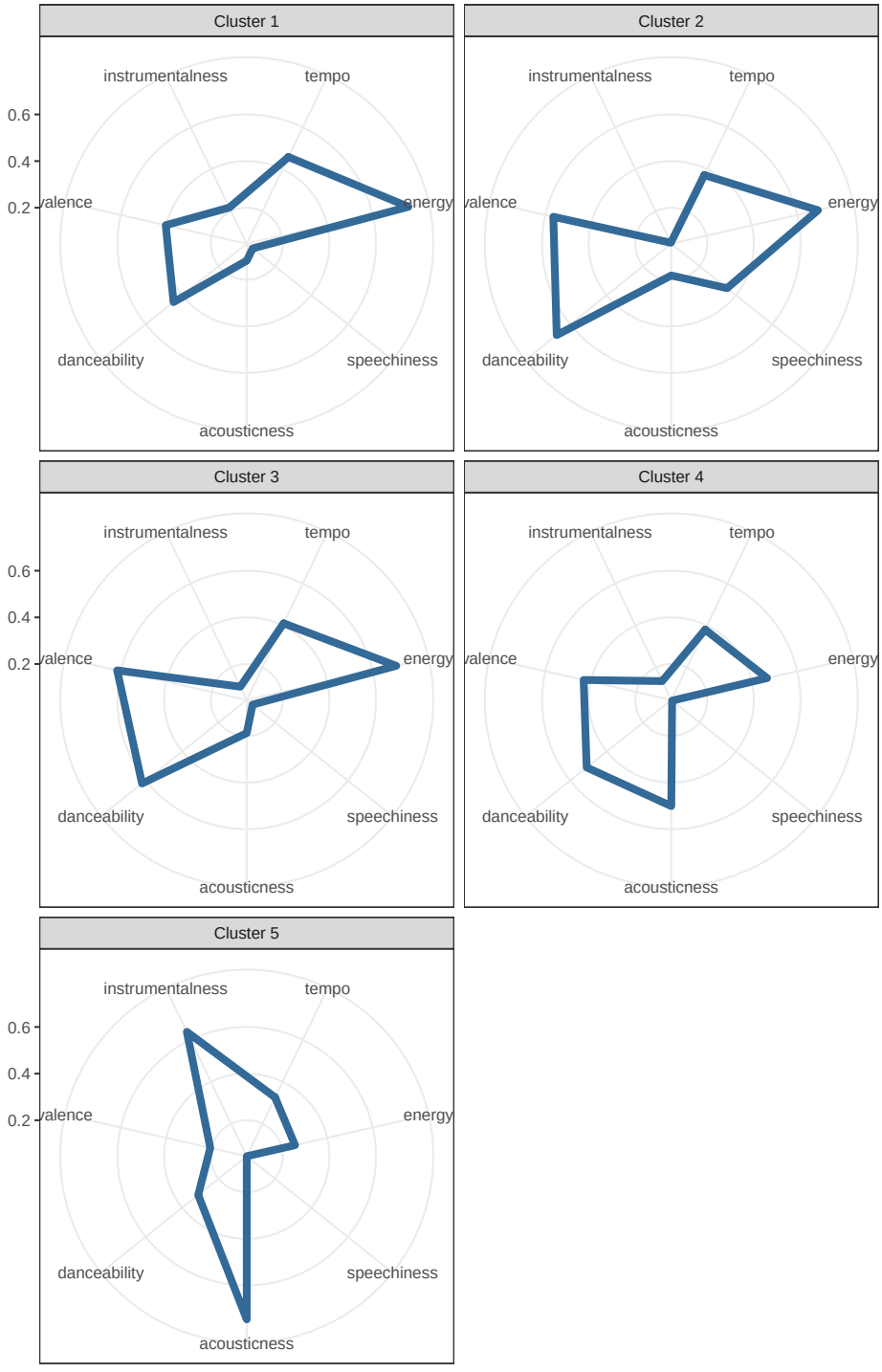

Figure 7.3: Acoustical Characteristics of the Clusters

Cluster 1 contains tracks focusing on *energy* and *tempo*, whereas cluster 2 contains tracks with high *speechiness*, *energy*, *valence* and *danceability* values. Cluster 3 is rather similar to cluster 2, besides the high *speechiness* values. This is, as the former contains mostly rap music, in contradiction to the latter, which contains different forms of pop music. This observation is underpinned by the genre distribution as discussed in Section 7.3.3. Furthermore, we witness that high *danceability* values correlate with high *valence* values (clusters 2 and 3). Cluster 5 contains tracks focusing on *acousticness* and *instrumentalness* as this cluster mostly contains classical music. Again, this is reflected in the genre distribution.

To answer RQ1, there exist differences based on the audio characteristics of playlists. By conducting a PCA we are able to explain 74% of the variance using the first 3 PCs: We observe, that the first PC separates *acoustic* and *instrumental* playlists from the rest. The second PC separates playlists with high *valence* and *danceability* from the rest and the third PC separates tracks with high *speechiness* values. Based on these characteristics, we are able to cluster playlists into 5 different groups using k-means. These are already valuable insights, however aiming to get a better understanding of the different clusters, we explore the genre distribution among each of the clusters in the next section.

7.3.3 Genre Distribution

In the following section, we provide a detailed analysis of the genres within the presented clusters.

We obtain the genre information for each track by using the genre tags that are provided by Spotify. Next, we derive a genre distribution for each cluster by counting the number of appearances of each genre in each cluster.

In a further step, we look into whether there is a difference in the genre distribution among the clusters. Therefore, we rely on two traditional similarity measures, namely Jaccard [50] and Pearson Similarity [84] to compute similarities between the individual clusters in terms of contained genres. Thus, in a first step, we count how many times each of the distinct genres occurs in each cluster. In a second step, we apply the two similarity measures on all pairs of clusters. Please note that the Jaccard coefficient is scaled between 0 and 1, whereas the Pearson-Similarity-coefficient is scaled between -1 and 1. In Equation 7.1 we state how the Jaccard Similarity between cluster x and cluster y is computed. We denote G_x to the set of all genres contained in cluster x and analogously, G_y is the set of all genres contained in cluster y. In Equation 7.2, we show the computation of the Pearson-Similarity-Coefficient.

In this Equation, X and Y represent vectors containing the counts of the different genres in cluster x and respectively the counts of the genres in cluster y. As introduced in Section 2.3, we denote *cov* as the covariance between two vectors and σ as the standard deviation. The computation results are stated in Table 7.2 and Table 7.3.

$$Jaccard_{x,y} = \frac{G_x \cap G_y}{G_x \cup G_y} \tag{7.1}$$

$$Pearson_{x,y} = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)} \tag{7.2}$$

	1	**2**	**3**	**4**	**5**
1	1.00	0.46	0.74	0.67	0.48
2	0.46	1.00	0.42	0.45	0.43
3	0.74	0.42	1.00	0.68	0.44
4	0.67	0.45	0.68	1.00	0.54
5	0.48	0.43	0.44	0.54	1.00

Table 7.2: Genre Similarities between Clusters using Jaccard Similarity

	1	**2**	**3**	**4**	**5**
1	1.00	0.20	0.77	0.63	0.26
2	0.20	1.00	0.46	0.19	0.03
3	0.77	0.46	1.00	0.76	0.24
4	0.63	0.19	0.76	1.00	0.36
5	0.26	0.03	0.24	0.36	1.00

Table 7.3: Genre Similarities between Clusters using Pearson Correlation

In Table 7.3 we observe a high correlation between clusters 1 and 3 ($r = 0.77$), which we lead back to the different forms of pop-music genres in those two clusters. Additionally, we observe a moderate similarity of several clusters. This implies that the same genres, mainly different forms of pop music, appear among several clusters. E.g., we find the "popchristmas"-genre in all clusters. Hence, we argue that users do not necessarily group tracks in a genre based manner. In other words, users use the same genres in different playlists. In addition, we observe that the correlation coefficient is nearly 0 between cluster 2 (the "rap cluster") and cluster 5 (the "classical music cluster"), confirming that rap-style music is rather different from classical music. Please note that results are highly consistent with Pearson and Jaccard Similarity.

Besides analyzing the genre distribution of the playlist clusters, we also study the user distribution among the clusters. We present the results in the next section.

7.3.4 Users among Clusters

Complementary to analyzing the genre distribution, we analyze the user distribution among the clusters representing playlists with similar acoustic features.

Firstly, we investigate how many users create playlists in a single cluster only. In particular, we are interested whether there are users only listening to a single type of music in regards to acoustic features. Related to that, we are also interested in how many users create playlists in different clusters. In Table 7.4, we state the number of users and the number of clusters they created playlists in. We observe that 64% of the users organize their music in playlists belonging to 3 or more different clusters. About 17% of the users create playlists among all 5 clusters, the maximum. On average, a user is represented in 3.08 clusters with a median of 3 (SD=1.36). From the median and mean we derive that the number of clusters users create playlists in is equally distributed. The average number of users per cluster is 631.60 with a median of 183 (SD=232.39).

# Clusters	# Users	Relative Portion
≥ 1	1133	100.00%
≥ 2	923	81.47%
≥ 3	733	64.70%
≥ 4	478	42.19%
≥ 5	200	17.65%

Table 7.4: User and Number of different Clusters

We are also interested in whether we can find clusters, which are populated by the same users. I.e., whether if users that create a playlist in cluster A are also likely to create a playlist in another cluster B. Therefore, we look at the correlation between the clusters in terms of users having created playlists in those clusters. As the data can be considered ordinal or at least discrete between 1 and 54, which is the maximum number of playlists a user created within a cluster, we apply Spearman's rank correlation coefficient [120]. The result of this analysis is shown in Table 7.5.

We do not observe any strong correlations between the individual clusters ($\rho > 0.7$), but we detect several moderate correlations ($\rho > 0.5$) between clusters. It is worth to mention that cluster 2, the "rap cluster", does not have

	1	2	3	4	5
1	1.00	0.32	0.55	0.54	0.41
2	0.32	1.00	0.42	0.36	0.23
3	0.55	0.42	1.00	0.64	0.40
4	0.54	0.36	0.64	1.00	0.56
5	0.41	0.23	0.40	0.56	1.00

Table 7.5: User-Cluster Correlations

any moderate correlations with other clusters. Cluster number 5, the "classical music cluster", only shows a low moderate correlation with cluster number 4. However, every cluster except cluster 2 (rap) does have this moderate correlation to cluster number 4, the "folk cluster", a cluster containing different forms of folk music according to the genre distribution. Further, clusters 1 and 3 also show a moderate correlation. With respect to the acoustic attributes of these two clusters, they are rather similar, except for the fact that cluster number 3, containing pop music, shows higher values for *valence* and *danceability*. We interpret music belonging to this cluster as "feel good music".

Complementary to this, to estimate the overall popularity of the clusters, we compute the number of users and playlists in each cluster as shown in Table 7.6.

Cluster	Users	%[4]	Playlists	%[5]	Avg. Pls./Users
1	768	68%	5,129	28%	6.68
2	427	38%	1,423	8%	3.33
3	1,032	91%	7,967	44%	7.72
4	793	70%	4,623	25%	5.83
5	447	39%	1,534	8%	3.43

Table 7.6: Users and Playlist per Cluster

We find that 91% of all users create playlists in cluster number 3, the "feel good music"–cluster. Also, 44% of all playlists are located in this cluster. Interestingly, nearly 40% of all users create playlists in the "rap" or "classical music" clusters, however, playlists in those clusters only account for 7% and respectively 8% of all playlists. This means that a high number of users create playlists containing rap or classical music, while at the same time, the number of playlists with respect to the total number of playlist is low. This means, that classical music or rap music can be considered as niche music with respect to the number of playlists, but not with respect to the number of users.

[4]Number of users in this cluster compared to the total number of users in the dataset

[5]Number of playlists in this cluster compared to the total number of playlists in the dataset

7.3.5 Summary

In the preceding sections, we presented our analysis of user-generated playlists on the music streaming platform Spotify. Our main contribution is an explanation of differences and commonalities among user created playlists based on their distinct audio characteristics. We observe that acoustical features partition playlists differently than if the genre is used. Along with that, we show that "feel-good" popular music is serving as a common musical ground across all users. 91% of all users create at least one playlist in the "feel good music"-cluster. Additionally, we observe, that classical music and rap music can be considered as niche music with respect to the number of playlists, however not as niche music when considering the number of users. Related to this we observe that users creating playlists in both, the rap and the classical music cluster, are rare. Along with that, we find that users in general listen to different styles of music (or at least organize different styles of music in their libraries). We refer to these different styles, we find via clustering, as *playlist archetypes* and aim to exploit them for music recommendations: users listen to different styles of music in different situations. Using our playlist archetypes, we implement a music recommender system that is capable of modeling the interaction effects between a certain user, a certain situational cluster and his or her musical preferences in this cluster. We introduce this recommender system in the next sections.

7.4 Amplified Music Recommender System

As outlined in the beginning of this Chapter, information about the situational context of a user has not been linked to acoustic features directly or to groups of playlists based on acoustic characteristics. In the remaining sections of this chapter, we show how contextual information and audio characteristics can be jointly leveraged for track recommendations. As introduced, we propose to make use of Factorization Machines (FM) [93] as these allow for exploiting latent features as well as allow to model interactions between input variables. As shown in the previous chapter and in [89], the characteristics of FMs are very well suited for the task of context-aware recommendations in settings as ours. Particularly, exploiting the interaction effects between contextual clusters extracted from the names of playlists (our situational clusters introduced in Chapter 6) and acoustic feature-based clusters (playlist archetypes) computed by leveraging the audio characteristics of the tracks are highly beneficial for computing track recommendations.

In the following, we describe the input data along with the data processing steps for the amplified recommender system before elaborating on the details of the multi-context-aware recommendation model.

7.4.1 Acoustic Feature Clusters

As explained in the previous sections, our proposed approach relies on clusters of playlists that share similar acoustic features, for instance the tempo or the valence of the tracks contained. Analogously to our preliminary study presented in the previous sections of this chapter and as suggested in [88] as well as [90], we propose to apply a factorization approach to find principal components of the acoustic feature space of playlists and subsequently use these principal components to find clusters of acoustically similar playlists. This allows us capturing a user's preference for playlists that share common acoustic features. In the next section, we show the integration of these findings into the recommendation process of ELFC-MR.

In a first step, we process and aggregate the acoustic features of the playlist dataset presented in Section 4.5. The acoustic features provided by Spotify have been already introduced in Section 7.3.1 and comprise *danceability*, *energy*, *speechiness*, *acousticness*, *instrumentalness*, *tempo* and *valence*. To derive the acoustic feature matrix (AFM), which serves as the input for the factorization task, we aggregate the acoustic features of each track per playlist using the arithmetic mean and removed the outliers (cf. Section 7.3.1). The result of this aggregation step is a $m \times n$ acoustical feature matrix AFM, where each of the m rows represents a playlist and each of the n columns represents an acoustic feature. In a next step, we apply factorization to the centered matrix we refer to as AFM' (all columns have a mean value of 0 and a SD=1) as depicted in Equation 7.3. Finally, we apply SVD [52] to compute a PCA [84].

$$AFM' \approx U\Sigma V^T \tag{7.3}$$

Based on the principal components obtained by the conducted PCA (cf. Section 7.3.2), we explain differences in playlists and, more importantly, estimate the number of acoustic features clusters (ACs). We determine the number of clusters empirically by the elbow curve of the explained variance of each PC and hence, the squared singular values s_i^2, which is the diagonal of the matrix Σ (cf. Section 7.3.2). Having obtained the number of ACs ($k = 5$), we compute the clusters by applying k-means on the dimension-reduced matrix $\overline{AFM}$, which allows us to explain 7% more variance than clustering on the original matrix. The clustering assigns each playlist and hence, implicitly each track to a playlist archetype that allows to capture a user's musical preferences for certain types of playlists. We depict the result of this approach in Figure 7.4, where each playlist is represented by a number between 1 and 5 indicating the cluster assignment. Furthermore, the clusters are marked by individual colors and are annotated according to the acoustic feature. As pre-

sented in Section 7.3.2, we observe that playlists that are highly influenced by instrumental and acoustic features are separated from the remaining playlist by the first PC. Furthermore, PC1 and PC2 separate energetic playlists with a high tempo from the remaining playlists. Finally, we are also able to separate playlists with high valence and danceability characteristics by PC1 and PC2. PC3, not visible in Figure 7.4, separates playlists with high speechiness values from other playlists. The presented clusters allow us to capture a user's preference towards certain acoustic characteristics of playlists and we exploit this information in the computation of track recommendations as described in Section 7.4.3.

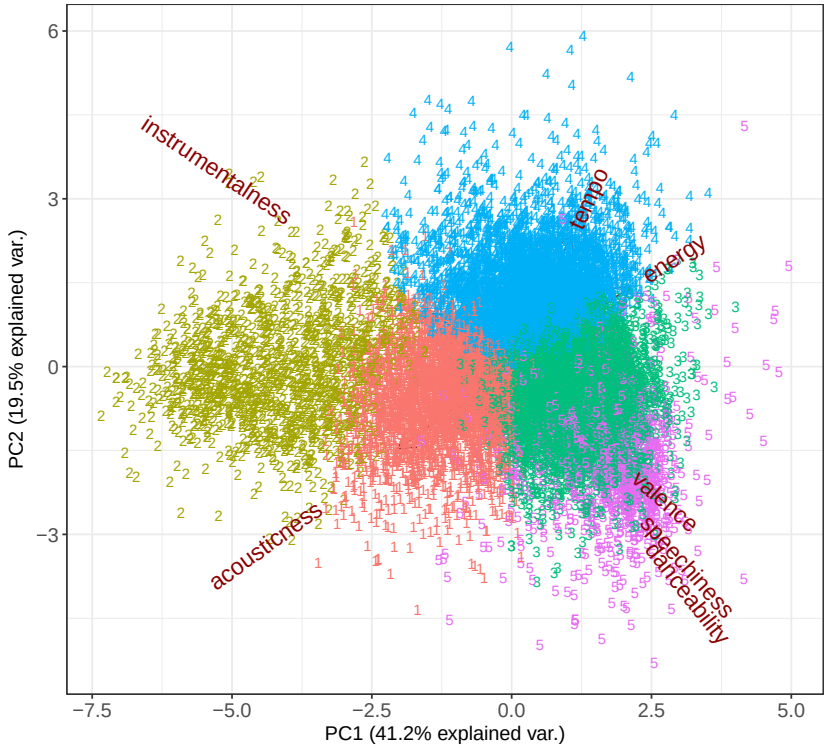

Figure 7.4: Latent Representation of Playlists and the five Archetypes

7.4.2 Situational Clusters

Besides capturing user preferences in terms of acoustic features, we also aim to contextualize playlists by extracting situational context information from the names of playlists using the approach described in Section 6.3. As we already elaborated on, the underlying assumption here is that the names of playlists provide information about the situational context in which the tracks

contained in the playlists are listened to. Examples might be "RUNNING", "Work Playlist", "Summer Fun", "Dustin's Workout Mix", or "Christmastime". Hence, by utilizing our proposed approach [87, 89], we mine for activities and other descriptors (seasons, events, etc.) in the names of playlists. A detailed description and evaluation of the approach is given in Section 6.3.

7.4.3 Recommendation Computation

The previous steps provide us with information about (i) a user's preference for playlist archetypes and (ii) the situational context in which a user listens to certain tracks. This information is extracted in the form of user-cluster assignments, where a situational cluster and a playlist archetype is assigned. We combine these clusters and the user listening history (all tracks a user interacted with) of the users to compute track recommendations.

Particularly, we propose to utilize factorization machines [93] to compute a predicted rating $\hat{r}$ for a given user i and a given track j, incorporating situational clusters (SC) and acoustic feature-based clusters (AC). We model the input for the rating prediction task as follows: in a preliminary step, $<user, track>$-pairs are enriched by assigning the corresponding situational cluster and acoustic feature-based cluster to each user-track pair, now forming $<user, track, SC, AC>$-quadruples. The information about the clusters is represented as nominal variables. By adding a fifth column rating to the dataset, we derive the input matrix R for our rating prediction problem to be solved by the factorization machine: for each unique $<user, track, SC, AC>$-quadruple, the rating $r_{u,i,s,c}$ is 1 if a user u has listened to a track i in situation (or context) s in a playlist belonging to archetype c. Our dataset does not contain any implicit feedback from users (i.e., play counts, skipping behavior, session durations or dwell times during browsing the catalog). Therefore, we cannot estimate a preference towards an item a user did not listened to as i.e., proposed by Hu et al. [49]. Thus, for each $<user, track, SC, AC>$-quadruple, for which we cannot obtain a rating for, analogously to the previous experiments in Section 6.5, we assume the rating to be $r = 0$ as proposed by Pan et al. [83]. The rating $r_{u,i,s,c}$ for each user u, track i, situational cluster s and acoustic cluster c can now be defined as stated in Equation 7.4.

$$r_{u,i,s,c} = \begin{cases} 1 & if\ u_u\ listened\ to\ t_i\ in\ AC_c\ in\ SC_s \\ -1 & otherwise \end{cases} \tag{7.4}$$

We already elaborated on that the class distribution of relevant and irrelevant tracks is highly unbalanced in the playlist dataset (as presented in Section 4.5). Therefore, we continue to rely on oversampling in order to achieve a 1:1 ratio

between relevant and irrelevant tracks. We train and test our classifiers on the oversampled dataset in order to avoid a bias towards negative values. For computing the predicted rating $\hat{r}_{u,i,s,c}$ based on the presented data, we extend the model of our context-aware recommender system presented in Chapter 6 (cf. Equation 6.3) to additionally model influence of the acoustic clusters c along with the influence of a user u, a track i and the situational cluster c on $\hat{r}$. Please note, that analogously to Model 6.3 we incorporate quadratic interaction effects ($\sum_{i=1}^{n}\sum_{j=i+1}^{n}\hat{w}_{i,j}b_ib_j$), where now $n \in \{1,2,3,4\}$ and similarly solve the factorization problem by applying a Markov Chain Monte Carlo (MCMC) solver [102] as proposed by Freudenthaler et al. [37].

$$\hat{r}_{u,i,s,c} = \mu + b_u + b_i + b_s + b_c + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\hat{w}_{i,j}b_ib_j \tag{7.5}$$

7.5 Experiments

In the following sections, we present the experiments conducted to assess the performance of the amplified recommender system. All experiments are based on the same experimental setup as presented in Section 6.5.4 and we use the same evaluation measures (cf. Section 6.5.3). However, as we integrate the acoustical features of the tracks in the recommendation process, we use an enriched version of the dataset. In particular, we add the seven acoustic features acquired via the Spotify API (cf. Section 7.3) to the dataset.

7.5.1 Data Modeling

For our experiments, we leverage the Spotify playlist dataset presented in Section 4.5. Analogously to the data modeling in Section 6.5.1, in a first step, we apply the proposed dimension reduction and clustering methods on the initial dataset. Hence, we reshape a dataset containing $<user, track, playlist\ name, acoustic\ features>$-quadruples into a dataset containing $<user, track, SC, AC, acoustic\ features>$-quintuples. We add the individual acoustic features for each track (AF), as we aim to leverage these features in a baseline approach in the course of our experiments. In a next step, we assign each track a rating value r as described in Section 7.4.3. The rating indicates whether a certain user listened to a certain track in a certain situational cluster ($r = 1$) or not ($r = 0$). Please note that a user might listen to the same song in different situations, whereas a track always belongs to the same acoustic feature-based cluster as naturally, these do not change.

For a better understanding, a fragment of the dataset is shown in Table 7.7. This excerpt shows that user 872 has listened to track 250,246 (belonging to acoustic cluster 1) in situational context cluster 0, whereas user 911 has listened to track 250,249 (belonging to acoustic cluster 4) in situational context 2. This dataset formed the foundation of our conducted experiments, which we present in the next section.

User	**Track**	**SC**	**AC**	**Rating**	$\mathbf{AF}_1$	...	$\mathbf{AF}_7$
872	309,275	0	3	1	0.24		0.16
872	309,275	1	3	0	0.24		0.16
872	250,246	0	1	1	0.10		0.12
911	250,246	0	1	0	0.10		0.12
911	250,249	2	4	1	0.77		0.32

Table 7.7: Data Set Fragment

The final dataset used for the presented evaluation contains 956 unique users who listened to 485,304 unique tracks. On average, a user in the dataset listens to 770.19 (SD=2,168.62, Median=264.50) tracks.

7.5.2 Evaluated Recommender Systems

To assess the effects of incorporating different contextual information encoded as clusters into a recommender system, we propose to evaluate a theoretical random baseline, three baseline approaches and a set of different extended models. Please note that we refrain from evaluating other classifier-based approaches besides FMs, as we have already shown a FMs superior performance in Chapter 6.

Firstly, we evaluate which models outperform the same random baseline as introduced in Section 6.5.2. To outperform the random baseline, the values for RMSE and MAPE have to be lower than 0.5. This is, as presented in Section 7.4.3, a track in the test data is relevant or not and hence, the real rating is 0 or 1. The chance of correctly guessing the correct rating in the sample space $\Omega = \{0, 1\}$ is $P(0) = P(1) = 0.5$ for each track. Based on this assumption, on average every second guess is wrong and hence, the error of every second guess is 1. Averaging this error among the number of recommendations gives an average error of 0.5. As for the random baseline of the rating prediction, for the top-n recommendations we assume that the probability of correctly guessing the rating of a track is $P = 0.5$. Hence, for the precision measure, the random baseline is 0.5. For the recall measure, the baseline is dependent on the number of recommendations n along with the number of relevant tracks $|T_r|$ and can be stated as $\frac{1}{2}\frac{n}{|T_r|}$ assuming that every second guess ($\frac{1}{2}n$) is a hit. Besides the random baseline, to assess and contextualize

the performance of our proposed approach, we employ a set of three baseline methods. Firstly, we utilize a non-model based approach that recommends the most popular tracks (MP) for each situational cluster. Furthermore, we evaluate a baseline that incorporates the users' listening histories as input to the FM and refer to this user-track model as the user-track model (UT). This UT model is then extended with the acoustic features (AF) of the tracks, as this is known to work well [75, 123]. Please note that we use the individual acoustic features of all tracks and do not rely on acoustic feature clusters in this model. We refer to this model as AF and consider it as a more advanced, but still context-agnostic baseline.

Naturally, we also evaluate our proposed approaches. Therefore, we derive a set of extended models utilizing the situational clusters mined from the playlist names (SC) and/or the acoustic feature clusters (AC). Table 7.8 gives an overview of all evaluated models according to the input data. Firstly, we evaluate a model extending the UT baseline by incorporating the situational clusters mined from playlist names (SC). Analogously, we evaluate a model extending the UT baseline by incorporating the playlist context (AC). Next, we evaluate a multi-context-aware model that additionally combines both clusters (SC+AC). Further, we evaluate a model extending the baseline AF model with the acoustic clusters (AC) and refer to this model as AF+AC. Finally, we evaluate a model incorporating the acoustical features (AF) along the situational clusters (SC) mined from the playlist names, the AF+SC model.

Model	**UT**	**AF**	**AC**	**SC**
MP				✓
UT	✓			
AF	✓	✓		
AC	✓		✓	
SC	✓			✓
AF+AC	✓	✓	✓	
AF+SC	✓	✓		✓
SC+AC	✓		✓	✓

Table 7.8: Overview of Evaluated Models

In the next section, we present the reader the results of the experiments and hence, the performance comparison of all models. A detailed description of the experimental setup and the evaluation metrics was already given in Section 6.5.3 and Section 6.5.4.

7.5.3 Experimental Results

In this section, we present the results of the evaluation of the models presented in Table 7.8. We firstly discuss the results of the top-n recommendations evaluation followed by an evaluation of the rating prediction task.

Top-n Recommendations

Analogously to our prior experiments (c.f. Section 6.5), for assessing the top-n recommendations, we state the *precison@100*, *recall@100*, *adapted recall@100* as well as the *adapted F_1-measure@100* for all evaluated models. Furthermore, to visualize the results for $n \in \{1..100\}$, we plot the corresponding adapted F_1-curves in Figure 7.5. We observe that all model-based approaches outperform the non-model-based MP baseline (recommending the most popular items in the according cluster) in regards to precision, recall and F_1. We explain this behavior by the fact that there is a natural cap rooted in the long-tailed distribution of the play counts in our data (cf. Section 2.1). Hence, popular tracks with high play counts among several users are rare [6]. This is why the set of "good" recommendations of the MP approach is limited to this small number of popular tracks across all users, naturally limiting its performance.

Model	Precision	Recall	Adapted Recall	Adapted $\mathbf{F_1}$
SC+AC	**0.75**	**0.52**	0.91	**0.82**
AF+SC	0.73	0.51	0.90	0.80
SC	0.70	0.50	0.89	0.78
AF+AC	0.66	0.49	**0.95**	0.78
AF	0.60	0.49	0.81	0.69
AC	0.52	0.49	0.84	0.64
UT	0.41	0.46	0.64	0.50
MP	0.48	0.40	0.50	0.49

Table 7.9: Top-n Recommendations Performance@100 ordered by F_1

Besides that, we observe a superior performance of our proposed multi-context-aware SC+AC approach incorporating both, acoustic clusters (AC) and situational clusters (SC). For the top-100 recommendations evaluated in this experiment, in terms of the F_1-measure, the SC+AC model performs 5.57% better across all n than the AF+AC approach, which is the best performing approach that does not leverage situational clusters. Along with that, according to the F_1-measure, our approach is 5.19% more accurate than a model solely exploiting situational clusters (SC) and 2.79% more accurate than a model exploiting acoustic features (no clustering) along with situational clusters (AF+SC). This finding is important because of two results: Firstly, we are able to model the acoustical characteristics of the tracks using our clusters as accurate as directly leveraging the acoustical features. Secondly, leveraging

the acoustic clusters allows us to compute recommendations more efficiently. We only have to estimate the interaction effects between the SCs and ACs and not between the SCs and all seven acoustical features. This makes the FM faster, as the complexity of the underlying model is much lower. Finally, a model solely exploiting acoustical clusters (AC) is 31.15% more accurate than the MP baseline and a model solely leveraging situational clusters (SC) outperforms the MP baseline by 59.99%. Furthermore, we observe, that the context-agnostic baseline solely considering the user listening history as input (UT) is outperformed even if contextual clusters are incorporated into the model in isolation by 64.52% (SC) and 28.52% (AC) respectively. Besides that, as outlined in the introduction, Celma [26] found, that collaborative filtering-based recommender systems do not necessarily exploit the long tail as well as content-based systems. This is reflected in our experiments: the highest recall values are achieved by the solely content-based AF+AC model. However, tough covering the long tail well by a 4.40% higher recall, the SC+AC model yields a 13.64% higher precision.

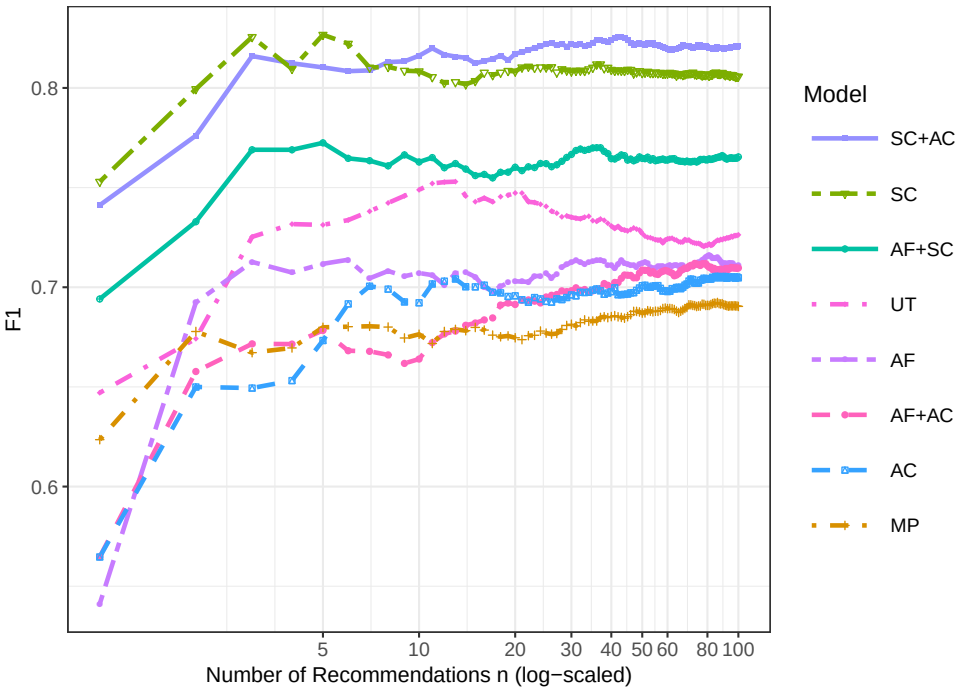

Figure 7.5: Adapted F_1 Curves

According to Bollen et al. [19], user satisfaction is highest when presenting the user with a short list of items naturally assuming that this recommendation list contains a sufficient number of relevant items. Therefore, we argue that the precision is highly important. We observe, that the SC+AC model computes

recommendations with the highest precision. I.e., it is 2.73% higher than the precision of the AF+SC model and 7.14% higher than the SC model.

To conclude, we observe that models incorporating acoustic features along with situational clusters provide the best performance independent of the number of recommendations n. This is why we argue that a model combining classical CF with acoustical features represents the user well, but integrating a user's situational context allows to capture the user's preferences even more precise in this scenario. Along with that, we see that for a small number of recommendations n ($n \leq 5$), incorporating situational clusters is highly important and incorporating solely situational clusters provides results comparable with those reached by additionally leveraging acoustical features. However, for suggesting a larger number tracks, more personalized recommendations are needed and hence, the integration of these acoustical features.

Evaluation of the Predicted Ratings

To get a deeper understanding of the recommendations computed by the FMs with respect to the different models used, we also evaluate the rating prediction task. This allows us to compare the recommendation confidence of the individual models, as the FM-component in our recommender system computes a predicted rating $\hat{r}$ that is the probability of a user listening to a certain track in a certain situational cluster. Hence, the more precise the rating prediction, which means that $\hat{r} = r = 0$ for irrelevant tracks and $\hat{r} = r = 1$ for relevant tracks, the more confident are the results of the underlying model. Furthermore, the probability impacts the ranking of items, as we use the predicted rating $\hat{r}$ for ranking (cf. Section 7.4.3). For this, we consider all items with a predicted rating $\hat{r} < 0.5$ as irrelevant and sort the remaining items in descending order. To compute deviations between $\hat{r}$ and r, we use the error measures presented in Section 6.5.3. In particular, we interpret $\hat{r}$ as the proxy for the perceived usefulness of a user towards an item and compute the RMSE and MAPE to compute a model performance.

We provide the results of the rating prediction measures computed using the top-100 recommendations in Table 7.10. Our results show models including the situational clusters SC or the SC along with the acoustic clusters (AC) achieve the lowest error rates across all error measures. Furthermore, we observe that models incorporating acoustic features represented by acoustical clusters and situational clusters (SC+AC) outperform a model solely using the situational clusters (SC) by 2.22% and a model using acoustic-features and situational clusters (AF+AC) by 18.52%, respectively. Along with the evaluation of the top-n recommendations in the prior experiment, these findings strongly support our initial hypothesis that clusters and the interaction effects between the input variables are highly beneficial for context-aware track recommen-

dations. To precisely estimate these effects, we compare our proposed FM to a FM without any interaction effects in a further experiment in the next section.

Interestingly, the most popular (MP) approach outperforms several models including the AC- as well as the AF-model. This is, as the MP approach assigns the top-n most popular tracks with a predicted rating of $\hat{r} = 1$ and the remaining (unpopular) items with no rating. For the latter, we assume a predicted rating of $\hat{r} = 0$. In contrast, the model-based FM approaches compute $\hat{r}$, which is the probability whether a given user has listened to a given track in a given situational cluster. Ultimately, for non-relevant and correctly classified tracks in the test set, the error is 0 for the most popular approach, whereas there is an error for the model-based approaches, though the track is correctly classified. This is, as all tracks with a predicted rating $\hat{r} < 0.5$ are classified as irrelevant which yields a true positive for the classification-based measures, but the rating prediction measures indicate an error due to the discrepancy of r and $\hat{r}$ between 0 and 0.5.

Model	RMSE	MAPE
SC+AC	**0.44**	**0.27**
SC	0.45	**0.27**
UT	0.52	0.37
MP	0.53	0.32
AF+SC	0.54	0.35
AF+AC	0.61	0.42
AC	0.61	0.42
AF	0.62	0.43

Table 7.10: Rating Prediction Measures ordered by RMSE

Summing up, analogously to the prior experiment, this experiment shows that incorporating the situational context (SC) is highly beneficial for the recommendation accuracy. Thus, we investigate the impact of the interaction effects between situational context and acoustical features in a final experiment.

Estimating the Interaction Effects

Finally, we are also interested in estimating the impact of the interaction effects on the recommendation quality. Therefore, we compare the performance of a FM that does not exploit any interaction effects and a FM that leverages interaction effects. Our experiments show that adding interaction effects allows for an 8% higher F_1-score (0.81 vs. 0.75) and hence, again strengthens our hypothesis that interaction effects are beneficial in such a scenario. This is also reflected in the RMSE of 0.44 for a model incorporating interaction

effects and an RMSE of 0.53 for a model not incorporating these. However, in this scenario, the interactions are constrained to 2-way interactions. We aim to weaken this constraint by applying higher order factorization machines (HOFM) in future work.

7.6 Summary and Contribution

The main contribution to the field music information retrieval presented in this chapter is two-fold. First of all, the presented study of Spotify users is the first study aiming at understanding user-curated playlists. Hence, we shift the focus from automatic playlist generation towards characterizing playlists based on their audio-features to get a deeper understanding of the user behavior on streaming platforms. Secondly, we are able to leverage the findings of the study in a multi-context-aware music recommender system. To the best of our knowledge, this is the first approach that is capable of jointly exploiting (i) information about a user's situational context and (ii) information about playlist archetypes that feature a set of typical acoustic characteristics. This allows modeling which kind of music is listened by which user in which situational contexts. In a profound offline evaluation based on users of the music streaming platform Spotify, we show that (i) the integration of situational context as described above improves the precision of music recommender systems and that (ii) acoustical features and thereby, a user's musical taste is particularly suitable to retrieve tracks a user likes from the long tail. We moreover show that a multi-context-aware recommendation model, leveraging both, situational context and clustered content-based features along with the interaction effects delivers the most accurate recommendations and simultaneously covers the long tail well.

CHAPTER 8

ELFC-MR III: Cultural Context[1]

In the related work in Chapter 3 we discussed that the music information retrieval and the recommender system community agree upon the fact that to build personalized music retrieval and recommender systems, context is essential. In particular, context needs to be considered along with the user listening history of a user (often referred to as the user profile). Also, the experiments conducted in this thesis show the importance of different contextual dimensions (cf. Chapters 6 and 7). Besides the integration of the situational context as presented in Chapter 6, the current location of the user can be leveraged as an additional source of contextual information. Under the term location-based context we subsume (i) points of interest (POI) and (ii) geographic locations. Whereas the first can be sights on a sightseeing tour or facilities users visit during their daily life, e.g., their workplace, a gym or a restaurant, the latter are geographic entities as raw GPS coordinates or higher geographical entities as the city, the country or even the culture a user is embedded in.

[1]This chapter is based on and content is partly reused from the following paper:
M. Pichl, E. Zangerle, G. Specht and M. Schedl. Mining Culture-Specific Music Listening Behavior from Social Media Data. In Proceedings of the 19th IEEE Symposium on Multimedia (ISM 2017), pages 208-215. IEEE, 2017.

Today, we observe a rise in research activities in the field of location-aware music information retrieval as recently data became available: As an increased amount of music is listened to via streaming services on mobile devices equipped with a GPS sensor, geolocalized listening profiles for a substantial amount of social media users who share their listening preferences and habits via social media platforms as Twitter are available. Prior works in the field of location-aware MIR were concerned with the visualization of local music listening behavior [42] as well as recommending music fitting to a certain POI a user is currently located at [20]. This was done via mobile applications developed for this special purpose and enabled the application's integrated recommender system to recommend tracks fitting to the current locational context. To give an example, classical music in front of a cathedral is recommended by such a system [20]. More recent works are concerned with integrating the geographic distance as a proxy for cultural similarity in the computation of the user similarity for CF-based recommender systems [113, 114]. However, those studies conclude that it is hardly possible to derive cultural (music) similarities among users using GPS coordinates or countries and continents. This is why we focus our research on culture-aware music recommendation and present our research in the following sections. For this, in a first step, we invent a novel culture-aware user similarity incorporating socio-economic and cultural features instead of raw GPS coordinates. In a second step, we evaluate the impact of this novel similarity on track recommendations using an amplified version of our multi-context-aware music recommender system presented in the previous chapter.

8.1 Research Overview

While a location as point of interest plays an important role to describe a listener's context [20, 12, 13, 126] and is partly covered by our situational clusters (i.e. music for going into the gym), the use of raw GPS coordinates to approximate the cultural context of a user may be misleading. This is, as they do not necessarily reflect differences in culture. Even worse, exploiting GPS coordinates to model *similarity* between listeners, which is key to build recommender systems, leads to systems that are agnostic to cultural characteristics as geographically far users might have a very similar cultural background. A common approach to approximate culture is to map GPS coordinates to countries. However, the underlying assumption that culture matches with political borders neglects the existence of ethnic groups within and beyond country borders. Therefore, a measure that integrates musical similarity and cultural similarity beyond countries' geographical borders is called for.

We close this research gap by our approach to model the user similarity by integrating two dimensions: Firstly, we integrate personal listening habits

described by acoustic properties mined from Twitter and Spotify. Secondly, we integrate cultural characteristics based on socio-economic and cultural factors available from the World Happiness Report[2]. Employing this similarity allows us to compute recommendations using a music-cultural-aware model. Our research to develop this music-culture-aware model is guided by the following four research questions:

RQ1 How can we find culture-specific music listening patterns among users?

RQ2 To which extent do a user's musical preferences, a user's cultural embedding and a user's geographical location influence the proposed model?

RQ3 What are the characteristics of the identified cultural groups in terms of musical taste?

RQ4 To which extend do the computed cultural groups influence track recommendations?

8.2 Analyzing Cultural Music Listening Behavior

8.2.1 Data Acquisition, Cleaning and Aggregation

For our analysis aiming at answering the research questions presented above, we require information about the (i) music listening behavior of users, (ii) their geolocation and finally the (iii) cultural characteristics of their home countries. We describe the process of acquiring this information in the following.

Listening Behavior

As the main data source for our research, we continue to use the Spotify playlist dataset as presented in Section 4.5. To detect the location of the contained users, we exploit that Spotify provides the means to share the tracks a user is currently listening to on Twitter and that people often send such #nowplaying tweets via their GPS-enabled mobile devices automatically adding the geolocation information to the tweet. An example of such a Tweet is given in Figure 4.1 in Chapter 4. Consequently, we search for Spotify user names on Twitter, which allows us to crawl geo-locatable tweets of each user using only exactly matching user names to reduce the number of false positive matches. Moreover, we apply a second measure to prevent false positives: We compare the #nowplaying tweets of the user (holding information about the music the user listened to) to the contents of his/her Spotify playlists. If we can find

[2] `http://worldhappiness.report`

the according tracks in the playlists of the user, we assume that we correctly matched the user's Spotify and Twitter handles. With this approach, we are able to match 22.73% of all user names contained in the playlist dataset. To map each user to a distinct position, we neglect location shifts beyond the first decimal point of the longitude and latitude values. In particular, we locate the users in rectangles of the size 11.1 km x 7 km, as these rectangles capture changes of the first decimal of the GPS position. Using this method, we observe that 80% of the users in the dataset constantly tweet from the same area (i.e., no location shifts beyond the first decimal). To determine the location for the remainder of the users, we apply a majority voting approach based on the grid rectangles and consider the rectangle in which most tweets were sent as the user's location. As a result, we can determine a unique country for 2,872 of the 3,335 users and remove the remaining users from the dataset. We could not determine a location for these users as some of the coordinates are located above sea. This can have several reasons: malfunctioning devices, devices sending fake GPS coordinates or people tweeting while traveling on ships and on airplanes. Furthermore, we restrict our dataset to countries with listening events of at least 10 distinct users (a total of 25 countries). The resulting dataset contains 104,390 listening events by 2,724 distinct users having listened to 62,104 distinct tracks. The top-10 countries with respect to the number of users are given in Table 8.1.

Country	**Users**	**Tracks**	**TPU**	**SD**
United States	1,131	35,560	31.44	124.51
Spain	417	16,133	38.69	81.12
United Kingdom	279	8,708	31.21	84.62
Mexico	233	15,073	64.69	84.01
Netherlands	91	1,881	20.67	36.10
Sweden	84	2,031	24.18	41.65
France	73	1,533	21.00	27.98
Italy	61	2,963	48.57	102.86
Germany	48	1,441	30.02	47.70
Chile	35	4,406	125.89	174.28

Table 8.1: Top-10 Countries (TPU=tracks per user, SD=standard deviation)

While the dataset is rather small compared to other available datasets (e.g., [106]), it nevertheless contains the necessary information in sufficient volume. This is backed by the fact that we find statistically significant differences in features of the different clusters using an analysis of variance (ANOVA) in Section 8.2.3.

As for modeling personal listening taste, we leverage the acoustic features of the playlist dataset (cf. Section 4.5) presented in Section 7.3. Furthermore,

in Chapter 7 and previous works [88, 90], we have already shown that these audio descriptors can be exploited for clustering tracks based on their audio features and subsequently, can contribute to improved context-aware music recommendations.

Cultural and Socio-Economic Data

To complement our model with cultural and socio-economic characteristics of countries, we rely on the World Happiness Report (WHR) [44]. We argue that people's cognitive and affective evaluations of their daily life and hence, their subjective well-being [33] provide a good indicator for cultural aspects as these have been shown to be directly influenced by cultural factors [116]. The WHR provides a set of aggregated measures capturing the perceived happiness of 156 countries: *gdp* is the real gross domestic product per capita; *freedom* measures the freedom to make life choices, *healthy life expectancy* states the healthy life expectancy at birth in the country, *generosity* specifies whether people in a country are willing to spend money to a charity; *social support* states if people have people helping them if they need support (i.e., relatives or friends); *corruption* and *happiness* measure the perceived corruption and happiness of citizens.

Finally, we put both, a user's personal music listening and the variables of the WHR into a single feature vector. Firstly, analogously to the analysis of playlists in Chapter 7 and to characterize the musical preferences of each user, we compute the arithmetic mean for each of the acoustic features of the songs contained in the user's playlists. Secondly, for the approximation of the cultural embedding of the users, we rely on the variables of the WHR as described previously. We add these variables to the feature vector as we aim to find cultural listening patterns by computing cultural similarity between users based on these variables. We assume that these variables reflect cultural similarity better than the mere geographic similarity. Finally, to homogenize values across all variables, we perform centering and scaling such that all elements of the vectors exhibit a mean of 0 and standard deviation of 1 for each of the acoustic and cultural variables. The feature vector representing a user then consists of two parts: (i) a user's individual music preferences captured by acoustic features as well as the (ii) user's cultural embedding approximated by socio-economic aspects extracted from the WHR.

8.2.2 User Models and Impact of Components

Aiming at answering RQ2, namely, to which extent do a user's musical preferences, a user's cultural embedding and a user's geographical location influence the proposed model, similar to the analysis of Spotify playlists (cf. Chapter 7),

we perform a PCA. As we are interested in the influence of the cultural and acoustic features and also aim to evaluate the suitability of GPS coordinates for the localization of users (in contrast to mapping a user's GPS location to the country level), we perform a PCA on a set of different user models:(i) user feature vectors as described in the previous section holding cultural and musical features; (ii) user feature vectors holding both cultural and musical features complemented with longitude and latitude information of the location of each user; (iii) user feature vectors solely containing musical features and the longitude and latitude information and hence, neglecting any cultural features in this model. For the conducted PCA, we set a minimum threshold of 75% explained variance, which is reached between the sixth and ninth PC depending on the model. In Tables 8.2, 8.3, and 8.4 we show the loadings of the PCs along with the respective dimensions.

To begin with, in Table 8.2 we present the PCs for the user model containing cultural (WHR) and acoustic data (AF) including the explained variance of each PC and the relative loadings of the cultural and acoustic features reflecting the feature's impact. We observe that the mean impact across all eight PCs of the WHR data is 41% and the mean impact of the acoustic features is 59%. Table 8.3 presents the PCA analysis for the user model complemented with GPS coordinates of users (GEO). The average impact of cultural data drops from 41% to 36% as the users in the same country feature the same WHR values and hence, a geographic similarity is implicitly covered, while it is now explicitly covered by the impact of the coordinates themselves. I.e., the cultural components act as a proxy in this scenario. The geographic distance based on the GPS coordinates has an average impact of 12%. We observe this low value, as part of the information is already implicitly covered by the variables of the WHR. Particularly for small countries, it holds that users originating from the same country have similar GPS coordinates and naturally, identical WHR variables. To complement this analysis, we also apply a PCA to a restricted dataset solely containing musical features and GPS coordinates. This analysis is aimed at getting a deeper understanding of the extent to which GPS coordinates may act as a proxy and allow for approximating cultural information in terms of user modeling. Therefore, in this analysis, we neglect any cultural features. The results of this PCA is shown in Table 8.4. We observe that in such a setting, the GPS coordinates explain 22% of the variance. Therefore, we argue that the geographic distance explains a substantially smaller fraction of the variance in comparison to the WHR data representing cultural aspects (41%). This is also reflected in the relative loadings of the PCs shown in Tables 8.3 and 8.4: In Table 8.3, there is no loading in the geographic dimension that substantially influences any PC. Although 27% in PC5 is not small, it is less than half of the variation explained by the WHR (60%). In Table 8.4, solely PC3 is substantially influenced by the geographic dimension.

PC/Dimension	Explained Variance	WHR	AF
PC1	14.96%	92.81%	7.19%
PC2	11.36%	10.82%	89.18%
PC3	10.57%	84.84%	15.16%
PC4	9.41%	10.58%	89.42%
PC5	7.84%	18.82%	81.18%
PC6	7.54%	16.63%	83.37%
PC7	7.32%	13.34%	86.66%
PC8	6.27%	76.27%	23.73%
Sum	75.27%	—	—
Mean Impact	—	40.52%	59.49%

Table 8.2: Explained variance for world happiness report features (WHR) and acoustic features (AF)

PC/Dimension	Explained Variance	WHR	AF	GEO
PC1	13.78%	83.72%	5.76%	10.52%
PC2	10.24%	45.72%	38.38%	15.90%
PC3	9.99%	35.04%	55.10%	9.86%
PC4	8.34%	14.98%	84.63%	0.39%
PC5	7.49%	60.41%	12.63%	26.95%
PC6	6.96%	15.55%	80.93%	3.52%
PC7	6.68%	8.95%	87.13%	3.92%
PC8	6.49%	8.06%	87.80%	4.14%
PC9	5.95%	54.43%	13.92%	31.65%
Sum	75.92%	—	—	—
Mean Impact	—	36.21%	51.81%	11.87%

Table 8.3: Explained variance for world happiness report features (WHR), acoustic features (AF) and GPS coordinates of users (GEO)

PC/Dimension	Explained Variance	AF	GEO
PC1	16.53%	95.63%	4.37%
PC2	13.78%	93.25%	6.75%
PC3	12.25%	18.93%	81.07%
PC4	11.41%	86.67%	13.33%
PC5	11.20%	73.36%	26.64%
PC6	10.69%	84.34%	15.66%
Sum	75.86%	—	—
Mean Impact	—	77.94%	22.06%

Table 8.4: Explained variance for acoustic features (AF) and GPS coordinates of users (GEO)

To conclude, we argue that in the current form, where cultural aspects are modeled on a country-level, adding a geographic user similarity in terms of GPS coordinates does not improve the result substantially. This is why we conduct the main analysis on the dataset comprising WHR data and acoustic features.

8.2.3 Cultural Clusters

To compute groups of users sharing common listening patterns as well as a common cultural background for our recommender system, we rely on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [35]. To visualize the data and the clusters accordingly, we rely on t-SNE [124], which is a state of the art data visualization method. In contrast to a PCA, t-SNE allows us to capture non-linear relationships. In preliminary experiments we found that DBSCAN applied to the t-SNE dimensionality reduced data provides the best results in regards to the variance of the acoustic attributes. As we aim to find cultural *listening patterns* across both underlying feature dimensions, we naturally strive to maximize the variance of the acoustic features between the clusters. The sum of all standard deviations (SD) of all acoustical attributes for DBSCAN is $SD = 4.25$, compared to k-means ($SD = 2.33$) and spectral clustering ($SD = 3.62$) and hence, we choose to utilize DBSCAN for the clustering of users. We moreover use the maximization of cluster variance for determining the DBSCAN parameters $minPts$ and ϵ. The first parameter ($minPts$) defines how many points have to be within the range ϵ, such that those points are considered as core points and form a cluster. Hence, in our setting, $minPts$ defines how many users have to be grouped inside the range ϵ by DBSCAN to actually form a cultural cluster. Accordingly, we tune the parameters of DBSCAN by maximizing cluster variance in a grid search. We find variance optimal parameters with $minPts = 20$ and $\epsilon = 2$ for the presented dataset.

For examining the characteristics of the nine obtained clusters in terms of musical taste and cultural characteristics, we provide an interactive web interface[3] which allows to compute clusters based on various clustering algorithms and settings and to interactively explore and visualize the obtained clusters and their characteristics. We state the highlights found via this web interface in the remainder of this chapter. We begin with discussing the results of the clustering step and subsequently focus on the individual characteristics of culture-specific listening patterns across and within individual countries.

[3] `http://dbis-mcc.uibk.ac.at`

Country-Cluster Assignments

In Figure 8.1, we depict the clusters resulting from applying DBSCAN on the user model containing both acoustic and cultural features and represented in a two-dimensional space computed using the t-SNE algorithm [124]. Moreover, we provide a world map depicting which countries belong to which cluster in Figure 8.2. In this plot, we indicate the user-cluster assignments by individual colors. For countries where users belong to several clusters, we apply a majority vote, where the country is assigned to the cluster to which most of its users belong. Here we can observe that, e.g., countries in Northern Europe mostly share similar listening characteristics with Canada, the United States, and Australia. Furthermore, we find a South American cluster, but Chile and Peru forming own clusters.

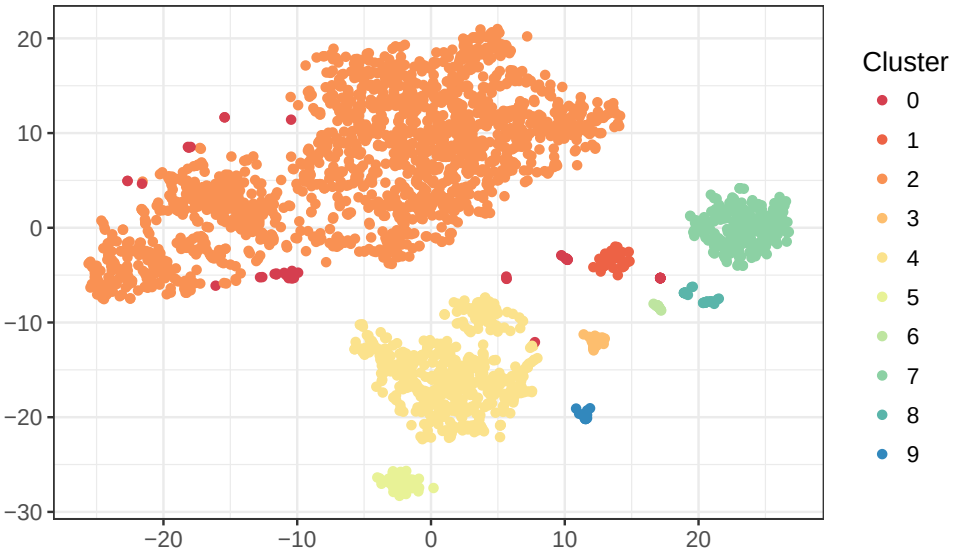

Figure 8.1: t-SNE and DBSCAN applied to Cultural- and Acoustic Features

Feature Distribution

To get a deeper understanding of the individual acoustic and cultural characteristics of all clusters, we provide bar plots for all clusters in Figures 8.3 and 8.4. Particularly, Figure 8.3 depicts the average cultural aspects of the nine clusters detected and Figure 8.4 presents the average acoustic features and their distribution across all clusters. By applying an ANOVA analysis, we find that the differences between clusters are significant across all features (p-value < 0.01).

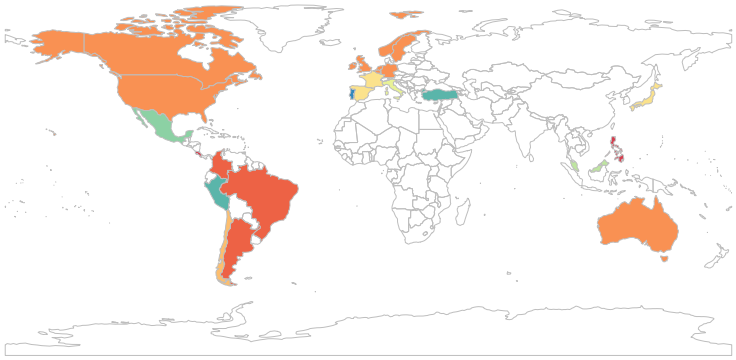

Figure 8.2: Country-Cluster Assignments

Feature Correlation and Cluster Analyses

To analyze listening patterns based on the proposed user model, we perform a correlation analysis of acoustic and cultural features. Using Pearson's correlation coefficient [84], we compare acoustic and cultural features and depict the obtained results in Figure 8.5. Besides several low positive and negative correlations between the acoustic and cultural features, we observe that happiness correlates well with valence (correlation coefficient $\rho = 0.61$). When inspecting clusters with respect to their valence and happiness values using the bar plots in Figures 8.3 and 8.4, clusters 1, 2, and 7 feature the highest values for both of these features. cluster 1 groups users from Argentina, Brazil, and Columbia; cluster 2 groups users from Northern Europe, the United States, and Canada, whereas cluster 7 solely contains Mexican users. Hence, these countries (stemming from three different clusters) feature high happiness values and tend to listen to high-valence music. Besides valence and happiness, in Figure 8.5, we additionally observe a moderate correlation between happiness and danceability with $\rho = 0.45$. We detect particularly high danceability values for clusters 7 and 9. cluster 7 contains users solely from Mexico, whereas cluster 9 solely contains Portuguese users. Italian and Portuguese users have similar listening patterns as other western users, however additionally consume music characterized by high danceability.

Focusing on the cultural dimension, we plot the cultural characteristics of the individual clusters in Figure 8.3. In this analysis, we detect that cluster 2 (Northern European countries, U.S. and Canada) features the highest values for the gdp-feature along with high values for generosity, health, and freedom. Generally, we observe these characteristics in most western countries. In contrast, cluster 7 (Mexico) features especially low values for social support and

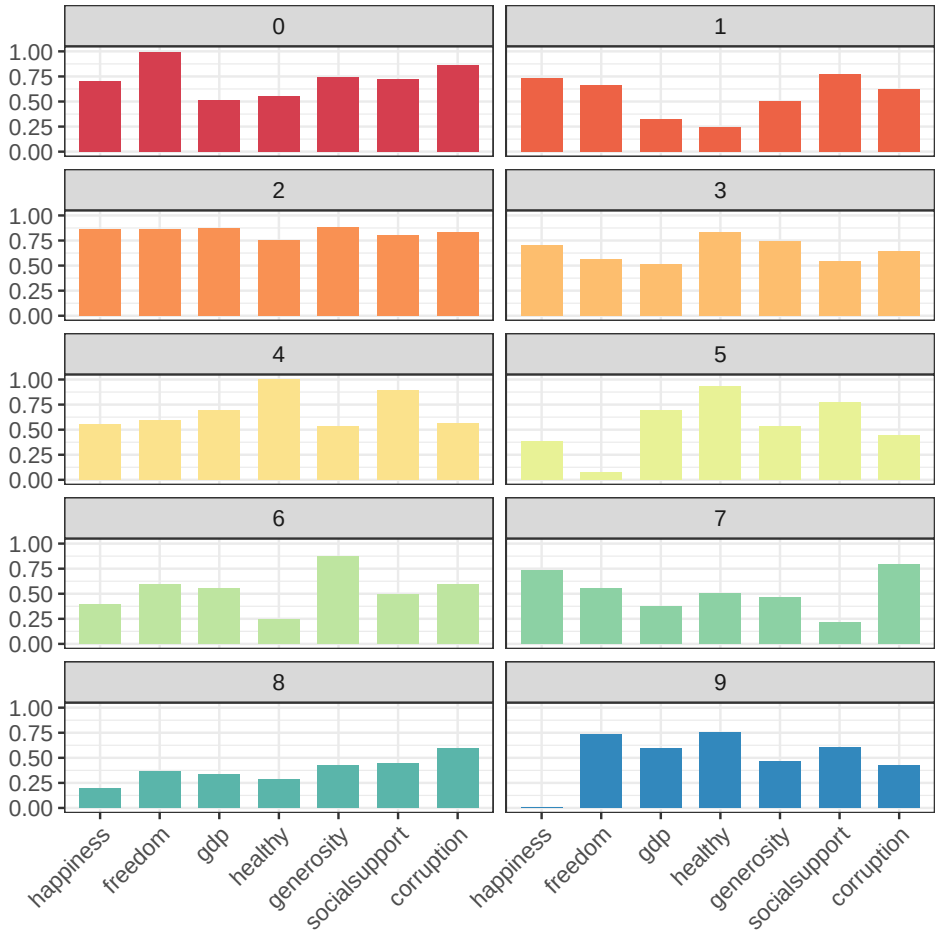

Figure 8.3: Cultural Characteristics of Clusters

high corruption compared to both cluster 1 (Argentina, Brazil, and Colombia) and cluster 2 (Northern European countries, U.S., and Canada). Besides these cultural differences, we moreover observe differences in the acoustic features plotted in Figure 8.4: compared to cluster 2, cluster 1 features lower instrumentalness and speechiness values. Hence, the slightly lower speechiness and instrumentalness values of cluster 1 are accompanied by lower generosity values. Cluster 7 is relatively similar to cluster 2 with respect to the acoustic characteristics, however, has lower speechiness values. Along with that, we find that cluster 2 features the highest speechiness and instrumental values among all clusters. Solely cluster 4 also features high values for speechiness *and* instrumentalness. This cluster contains users from France, Italy, Japan, and Spain.

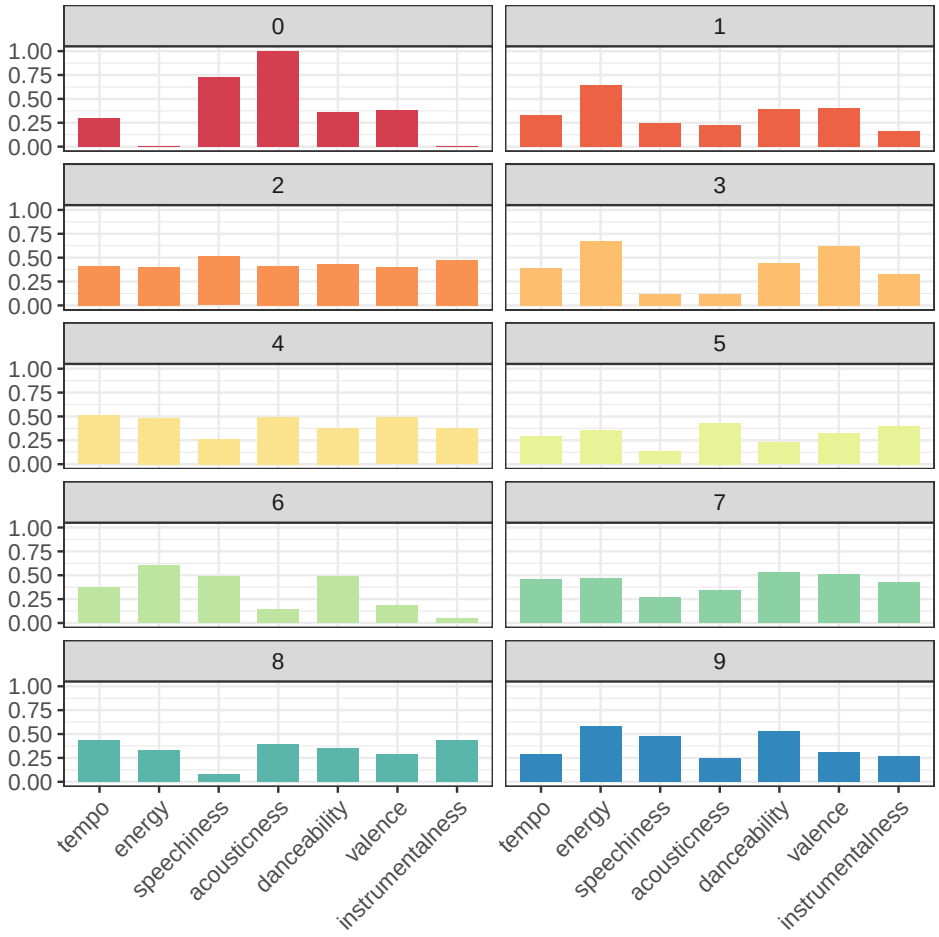

Figure 8.4: Musical Characteristics of Clusters

Furthermore, we see a strong correlation between French and Japanese users based on the cultural features ($\rho = 0.91$). While most of the feature values differ (but correlate), we find similar values for freedom and corruption. We argue, that in this special case, the variables of the WHR represent cultural aspects where these two countries appear similar, however, we assume that for Japan there exist other cultural aspects where those countries differ but are not captured by the WHR.

Another interesting finding is the observation that high speechiness values correlate strongly ($\rho = 0.81$) with high freedom values in the dataset. We presume that clusters 2 and 4 containing western countries are characterized by a culture where freedom is important and along with that, music with high speechiness values and hence, mostly rap music is popular and important.

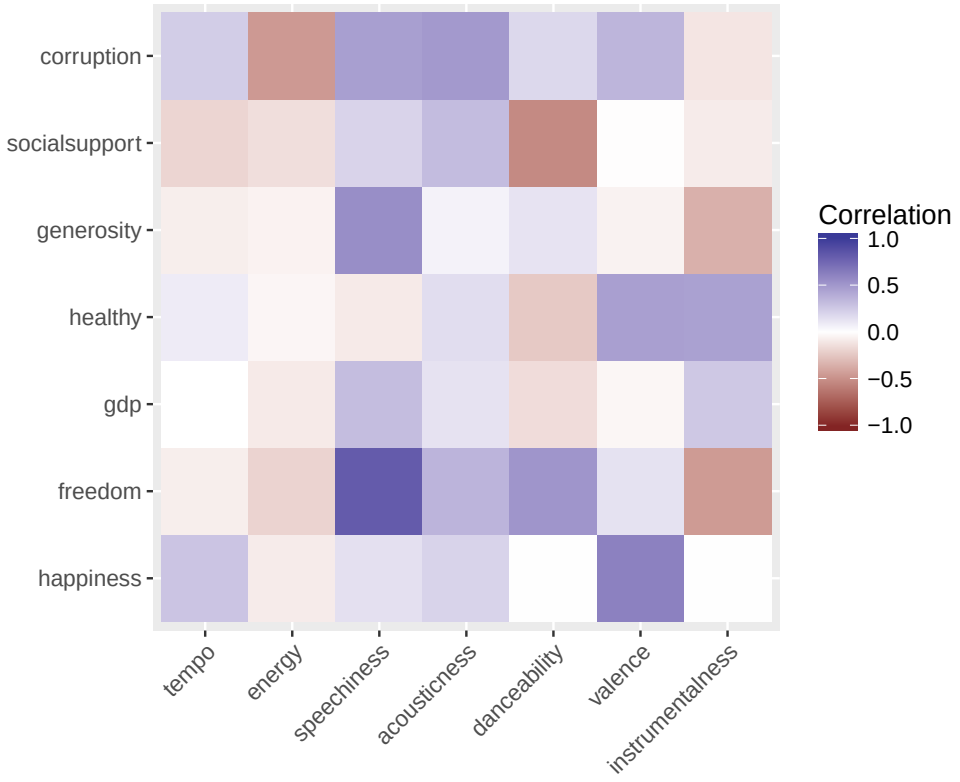

Figure 8.5: Correlations between Cultural- (Y-Axis) and Acoustic (Y-Axis) Features across all Clusters

We detect a similar pattern for the medium correlation between healthiness and instrumentalness ($\rho = 0.45$): clusters 2 and 4 group countries, where instrumental music is more popular than in other countries (and clusters). At the same time, cluster 4 (grouping European and Japanese users) as well as cluster 2 (grouping Northern Europe users, U.S., Canada, and Australia) are characterized by the highest health values. In fact, cluster 4 is characterized by the highest health values among all clusters.

Country-Specific Patterns

Besides looking at patterns that span across countries as presented in the previous section, we also aim to detect country-specific listening patterns. Therefore, we have a deeper look at the clusters solely grouping users of a single country in the remainder of this section. Particularly, we are interested in the features that make these countries stand out and hence, form individual clusters.

Cluster 3, a cluster solely grouping users from Chile, exhibits similar patterns to cluster 1 (grouping users from Argentina, Brazil, and Colombia), besides 28% higher instrumentalness values. Italian users are located in clusters 4 (4.92% of all Italian users) and 5 (95.08%, respectively). In contrast to cluster 4, where other European users are located, Italian users in cluster 5 are characterized by lower tempo (-2.23%), valence (-4.35%) and danceability (-2.41%) values. Cluster 6, containing users from Malaysia, is characterized by the lowest acousticness values among all clusters, which are 13.17% lower than the mean computed. This is accompanied by 10.42% lower instrumentalness values. Finally, Mexican users form their own cluster and hence, have their own music listening pattern. The music listening behavior of Mexican users is similar to the western cluster 2, but we observe 13.05% lower speechiness values along with 5.32% lower acousticness values.

8.3 Summary

We consider the result of the conducted analysis still as early, as we are aware of the fact that our findings based on Twitter and Spotify users do not necessarily represent the world's music taste aside of music streaming. Further, we note that the number of users analyzed in this study is still limited and hence, naturally affects the generalizability of our study. Nevertheless, the results of our clustering approach, bringing together the cultural embedding of a user with his or her musical preferences, suggests that there exist several culture-specific music listening patterns. We categorize those listening patterns into two groups: we observe country-specific listening patters as well as cross-country listening patterns that span across several countries. The latter are not restricted to neighboring countries or continents, as we see in the bias towards instrumental and rap music for western countries reaching from Australia over Europe to the United States and Canada or commonalities in the music listening between Europe and Japan. Besides those cross-country listening patterns, we find country-specific patterns, for instance, the bias of Italian and Portuguese users towards music with high danceability values. Our findings show that Japan and France feature a high correlation based on cultural features. While this might not seem obvious, according to the socio-economic features contained in the WHR data, the correlation holds. However, this also signals that characterizing a user's cultural embedding by WHR data only does not fully capture the cultural background of users. Therefore, we also aim to extend the description of the cultural embedding with further characteristics to improve precision in this regards in future work.

To conclude, based on our findings, we argue that a music recommender or retrieval system incorporating a user's cultural background allows for providing more fine-grained and personalized results. We particularly consider the find-

ing that there are on the one hand clusters that span across multiple countries, and on the other hand, clusters that only comprise users of a single country as highly relevant. Furthermore, given that cultural information explains 41% of the variance in our dataset, therefore nearly as much as music content information (59%), we argue that cultural information is an important contextual variable that allows for better characterizing users.

As we believe that these findings can contribute to providing more personalized and culture-aware music recommendations by integrating country-specific listening patterns and cultural information, we amplify the ELFC-MR approach to consider music-cultural clusters.

8.4 Amplified Music Recommender System

As outlined in the previous sections, we aim to provide culture-aware music recommendations by integrating country-specific listening patterns and cultural information into the recommendation process. In the remaining sections of this chapter, we show how music-cultural clusters (MCCs) can be leveraged for track recommendations and benchmark a recommender system incorporating these clusters for music recommendations to the approach prior presented in Chapter 7. As introduced, we propose to make use of our FM-based recommendation approach (ELFC-MR) to answer the research question to which extend music-cultural clusters are beneficial for track recommendations. In the following, we shortly describe the input data of the recommender system before elaborating on the details of the culture-aware recommendation model.

8.4.1 Music Cultural Clusters

As shown in Section 8.2.3, to compute groups of users sharing common listening patterns as well as a common cultural background, DBSCAN [35] is a suitable clustering algorithm. In prior experiments we found, that applying DBSCAN on the t-SNE dimensionality reduced data with a minimum number of users per cluster $minPts = 20$ within the range $\epsilon = 2$ provides the best results in regards to the variance of the acoustic attributes. We aim to maximize the variance of the acoustic features between the clusters, as we want to find cultural listening patterns. These clusters allow us to capture a user's music-cultural embedding, contextual information we exploit for the computation of track recommendations as described in the next section.

8.4.2 Recommendation Computation

In the previous Chapter, we presented how to leverage information about (i) a user's preference for playlist archetypes and (ii) the situational context in which a user listens to certain tracks for track recommendations. In this section, we present how we additionally model and incorporate (iii) a user's music-cultural embedding.

Analogously to Chapter 7, we propose to utilize factorization machines [93] to compute a predicted rating $\hat{r}$ for a given user u and a given track i, incorporating situational clusters (SC), acoustic feature-based clusters (AC) and music-cultural clusters (MCC). Relying on this information, we model the input for the rating prediction task as follows: in a preliminary step, we assign the corresponding situational cluster, acoustic feature-based cluster and music-cultural cluster to each $<user, track>$-pair, to form $<user, track, SC, AC, MCC>$-quintuples. The information about the clusters is represented as nominal variables. By adding a rating column to the dataset, we derive the input matrix R for our rating prediction problem to be solved by the factorization machine: for each unique $<user, track, SC, AC, MCC>$-quintuple, the rating $r_{u,i,s,c,m}$ is 1 if a user u embedded in music-culture m has listened to a track i in situation (or context) s in a playlist belonging to archetype c. As we build our experiment on the same data as our previous experiments, we do not have any implicit feedback as we already discussed in Section 7.4.3. Finally, the rating $r_{u,i,s,c,m}$ for each user i, track j, situational cluster s, acoustic cluster c and music-cultural cluster u can be defined as stated in Equation 8.1.

$$r_{u,i,s,c,m} = \begin{cases} 1 & if\ u_u\ in\ MCC_m\ listened\ to\ t_i\ in\ AC_c\ in\ SC_s \\ -1 & otherwise \end{cases} \tag{8.1}$$

As we continue to use the unbalanced playlist dataset (cf. Section 4.5), we rely on oversampling in order to achieve a 1:1 ratio between relevant and irrelevant tracks to avoid a bias towards negative values. As described in Section 8.2.1, we use a subset as we could not determine a location for all users in the playlist dataset. For computing the predicted ratings $\hat{r}_{u,i,s,c,m}$ based on the presented data, we extend the model of our context-aware recommender system presented in Chapter 7 to additionally model and estimate the influence of music-cultural clusters m along with the influence of a user u, a track i, the situational cluster s and the acoustical cluster c on $\hat{r}$. Analogously to the model in Equation 7.5, we incorporate quadratic interaction effects $(\sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j)$, where $n \in \{1, 2, 3, 4, 5\}$. Finally, to solve the factorization problem, we stick to applying a Markov Chain Monte Carlo (MCMC) solver [102].

$$\hat{r}_{u,i,s} = \mu + b_u + b_i + b_s + b_c + b_m + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} b_i b_j \tag{8.2}$$

8.5 Experiments

In the following sections, we present the experiments conducted to assess the impact of the MCCs. All experiments are based on the same experimental setup as presented in Section 6.5.4 and we use the same evaluation measures as in Section 6.5.3. However, as we integrated the MCCs in the recommendation process, we only consider the users in the dataset for whom we could determine a location.

8.5.1 Data Modeling

For our research on a culture-aware user similarity we presented in Section 8.2 and for the experiments conducted in this section, we enrich the Spotify playlist dataset with the locations of the users along with socio-economic and cultural features (cf. Section 8.2.1). Analogously to the data modeling in Sections 6.5.1 and 7.5.1, in a first step, we apply the proposed dimension reduction and clustering methods on the initial dataset. Thus, we reshape a dataset containing $<user, track, playlistname, acoustic\ features, whr\ features>$-quintuples into a dataset containing $<user, track, SC, AC, MCC>$-quintuples. For this, we compute situational clusters as described in Section 6.3, compute the playlist embedding as shown in Section 7.3.2 and compute music-cultural clusters as described in Section 8.2.3. In a next step, we assign each track a rating value r as described in Section 8.4.2. The rating indicates whether a certain user listened to a certain track in a certain situational cluster ($r = 1$) or not ($r = 0$). Please note that a user embedded in a certain music-culture might listen to the same song in different situations, whereas a track always belongs to the same acoustic feature-based cluster as naturally, these do not change.

For a better understanding, a fragment of the dataset is shown in Table 8.5. This excerpt shows that user 872 embedded in music-cultural cluster 5 has listened to track 250,246 (belonging to acoustic feature-based cluster 1) in situational context cluster 0, whereas user 911 embedded in music-cultural cluster 6 has listened to track 250,249 (belonging to acoustic feature-based cluster 4) in situational context 2. This dataset forms the foundation for our experiments, which we present in the next section.

User	Track	SC	AC	MCC	Rating
872	309,275	0	3	5	1
872	309,275	1	3	5	0
872	250,246	0	1	5	1
911	250,246	0	1	6	0
911	250,249	2	4	6	1

Table 8.5: Dataset Fragment

8.5.2 Evaluated Recommender Systems

To assess and contextualize the performance effects of incorporating our proposed music-cultural clusters (MCC) into the recommendation process, analogously to the previous chapter, we evaluate a set of extended UT models. These models utilize the situational clusters mined from the playlist names (SC), playlist context derived from acoustic feature clusters (AC) and the newly introduced music-cultural clusters (MCC). Furthermore, we propose a set of three baseline approaches and finally compare the results to the SC+AC model introduced in the previous chapter.

In Table 8.6, we give an overview of all evaluated models. Firstly, we evaluate a model extending the UT baseline by incorporating the music-cultural clusters mined from socio-economic, cultural and acoustic features (MCC). Secondly, we evaluate a model extending the MCC model with situational clusters (SC) mined from the playlist names refer to this model as SC+MCC. Thirdly, we evaluate a model incorporating the music-cultural clusters (MCC), the situational clusters (SC) and the content-based clusters (AC) and refer to this model as the MCC+SC+AC model. As for the baselines, we propose to substitute the MCCs of a user with the corresponding home country of the user. In particular, we leverage the country name as a factor variable. Similar to the clusters, the country is encoded as a nominal variable. We evaluate a Country, a Country+AC and a Country+SC model. Finally, we compare all the models to the SC+AC model, the best performing model yet.

Model	UT	Country	MCC	AC	SC
Country	✓	✓			
Country+AC	✓	✓		✓	
Country+SC	✓	✓			✓
MCC	✓		✓		
SC+MCC	✓		✓		✓
MCC+SC+AC	✓		✓	✓	✓
SC+AC	✓		✓	✓	

Table 8.6: Overview of Evaluated Models

After giving an overview of the evaluated models, we present the results of the experiments in the subsequent section. A detailed description of the evaluation metrics is given in Section 6.5.3 and of the experimental setup is given in Section 6.5.4.

8.5.3 Experimental Results

In this section, we present the results of our conducted experiments of the models presented in Table 8.6. We firstly present the results of the top-n recommendations evaluation followed by an evaluation of the rating prediction task.

Top-n Recommendations

Analogously to our prior experiments (c.f. Section 6.5 and Section 7.5), for assessing the top-n recommendations, we state the *precison@100*, *recall@100*, *adapted recall@100* as well as the *adapted* F_1*-measure@100* of the evaluated models in Table 8.7. Along with that, in Figure 8.6, we plot the corresponding adapted F_1-curves for $n \in \{1...100\}$. As already shown and discussed in the experiments in Section 7.5, all model-based approaches outperform the non-model-based MP baseline also in this scenario.

Model	Precision	Recall	Adapted Recall	Adapted $\mathbf{F_1}$
SC+AC+MCC	**0.97**	0.51	**0.99**	**0.98**
SC+MCC	0.88	0.49	**0.99**	0.92
SC+AC	0.87	**0.52**	0.98	0.91
Country+SC	0.81	0.50	0.97	0.84
MCC	0.64	**0.52**	0.96	0.73
Country	0.69	0.50	0.95	0.77
Country+AC	0.68	0.51	0.97	0.76

Table 8.7: Top-n Recommendations Performance@100 ordered by F_1

.

Most importantly, we observe a superior performance of all our presented multi-context-aware models in this dataset. We observe that SC+AC+MCC, SC+MCC, SC+AC, outperform all other models including the country baselines. We further observe that models leveraging situational clusters outperform all other models. This repeatedly highlights how important situational context is. Along with that, we detect that by adding music-cultural clusters, the precision of recommendations can be improved substantially. In particular, a model incorporating situational clusters along with music-cultural clusters (SC+AC+MCC) performs 6.52% better than a model incorporating acoustic-

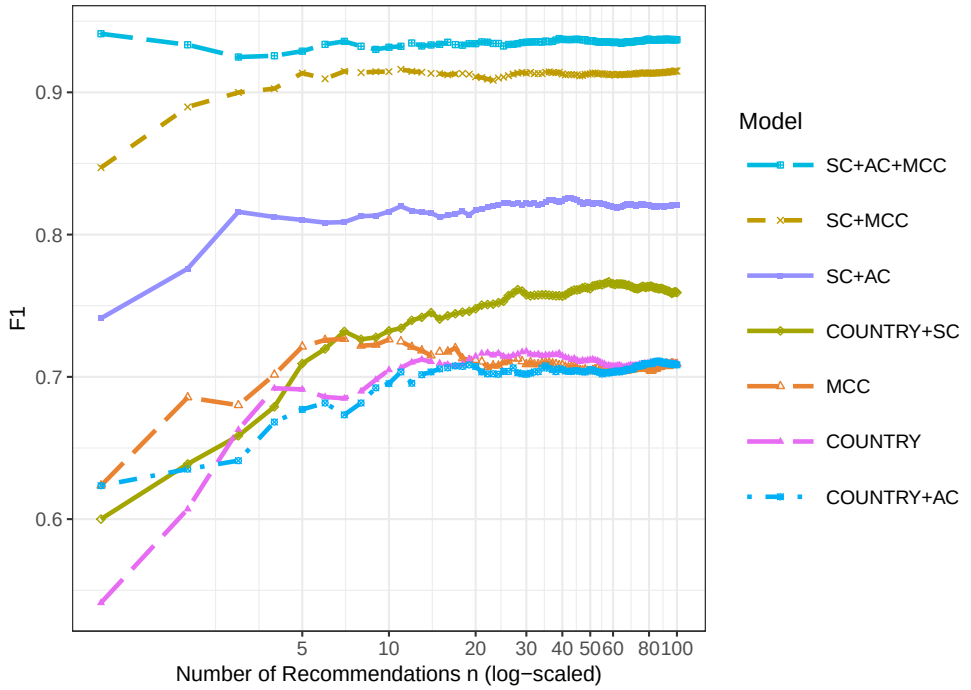

Figure 8.6: Adapted F_1 Curves

and situational clusters (SC+AC) in terms of the adapted F_1-measure. In Table 8.7 we moreover see that by replacing acoustic clusters (AC) with music-cultural clusters (MCC) the precision slightly increases by 1.10% while recall slightly drops by 5.77%. This indicates that music-cultural clusters already capture listening patterns (AC) very well. The comparison of the SC+AC model to the model incorporating MCCs (SC+AC+MCC) shows, that there exist music-cultural listening patterns that can be leveraged for music recommendation.

To conclude, we detect that musical cultural clusters substantially improve the recommendation precision, which is the most important measure in a setting as ours [19, 118]. Further, we observe that music-cultural clusters already cover a large portion of the users' music listening habits. I.e., adding further content-based clusters only results in a small precision improvement, although they give a substantial improvement when not incorporating MCCs (c.f. Section 7.5.3). Because of this and the fact that the SC+MCC model is superior to the SC+COUNTRY model, we argue that computing MCCs is a reasonable approach to improve the recommendation precision. In particular, we are able to show the existence of music-cultural listening patterns and their applicability for music recommendation. Next, to get an even more fine-grained understanding of the computed recommendations, we have a look at how ac-

curate the different FM models compute the probabilities that a certain track was actually listened or not.

Evaluation of the Predicted Ratings

In this section, we give detailed insights into the performance of the different FM models. In particular, we assess how accurate the probabilities computed by the FM indicate whether a track was listened or not. In Table 8.8, we state the RMSE and MAPE computed by comparing $\hat{r}$, which is the probability whether a song was listened, to the actual rating r which can be either 0 (listened) or 1 (not listened).

Model	RMSE	MAPE
SC+AC+MCC	**0.24**	**0.10**
SC+MCC	0.32	0.14
SC+AC	0.44	0.27
Country+SC	0.53	0.35
MCC	0.61	0.41
Country	0.61	0.42
Country+AC	0.61	0.42

Table 8.8: Rating Prediction Measures ordered by RMSE

In the results of this experiment, we observe, that the most precise recommendations in terms of the stated probability are computed by the SC+AC+MCC model. Hence, we can validate the findings of the previous experiment. Moreover, we detect that the SC+AC+MCC model provides substantially more accurate predicted ratings than the SC+MCC model. The RMSE is 25.00% lower. Compared to the SC+AC model, the RMSE is 45.45% lower. Furthermore, neither the MCC nor the country-based models can compete the random baseline of 0.5. To conclude this experiment, we observe that MCCs in isolation cannot compete with other approaches, however, in a combined model serving as an additional contextual variable, they substantially improve the recommendation results.

8.6 Summary and Contribution

Triggered by the finding that the music-cultural embedding of a user explains a substantial portion of the variance in the music listening behavior of Spotify users, in a future work, we are particularly interested in analyzing cultural music listening patterns beneath the country level. An analysis beneath the country-level may mitigate the weakness of performing a majority vote to assign a country to a cluster. Moreover, it allows a more fine-grained analysis

of cultural listening patterns, probably revealing regional listening patterns. Whereas analyzing regional acoustical patterns is possible due to precise GPS coordinates, the cultural analysis is challenging as we are not aware of a consistent and sufficiently extensive data source observing cultural aspects beneath the country-level. As currently, cultural features are leveraged on the country level, our analysis indicated that the incorporation of precise GPS coordinates is not beneficial at this stage. Besides a sub-country analysis, it would indeed be interesting to optimize the current user model: With the current approach, we use the arithmetic mean of each individual acoustic feature for aggregating values of each track, which implies the simplification that each user follows a single, homogeneous listening pattern. Hence, our analysis may benefit from a more comprehensive user model revealing multiple listening patterns per user.

Despite the aforementioned limitations of this work, we find that by integrating music-cultural clusters into the recommendation model, we can substantially improve the recommendation precision. Hence, we contribute a multi-context-aware recommendation model leveraging the music-cultural embedding of a user to the MIR community. We find that the integration the music-cultural embedding as a further contextual information improves the recommendation precision substantially. I.e., a model incorporating music-cultural cluster, acoustical clusters and situational clusters computes 11.49% more precise recommendations than a model incorporating acoustical clusters and situational clusters. We find that music-cultural clusters combined with situational clusters and/or music preferences are highly beneficial, whereas music-cultural clusters considered in isolation cannot outperform the model leveraging situational context presented in Chapter 7.

CHAPTER 9

Conclusion

In this work, we present novel *mining methods* and *user models* applicable for multi-context-aware music recommendation. By profound offline experiments relying on the presented multi-context-aware music recommender system prototype, we show that the situational context, the playlist context and the music-cultural context of a user can substantially contribute to precise track recommendations. Naturally, the current situation while listening to music highly influences the perceived usefulness of recommended tracks. We could validate this assumption by answering the first of our four guiding research questions (RQs) (c.f. Section 1.2), namely *how do people organize the increased amount of tracks available in the music streaming era?* To answer this RQ, we conduct a quantitative analysis of Spotify users. We find that (amongst others) users organize their tracks after the intended use. Hence, tagging tracks with a certain "use" allows us to compute highly fitting track recommendations. With respect to our second RQ, *how to model the music listening behavior of music streaming users?*, we find that factorization machines (FMs) work well. Using our FM-based music recommender system prototype, we show that the integration of the situational context is highly beneficial for the accuracy of our track recommendations. Hence, our prior assumption about the situational context is reflected in the results of our offline exper-

iments. In addition to that, by computing a content-based playlist context, which is to compute groups playlists based on their audio characteristics, we could extend our model to reflect that users listen to a certain kind of music in a certain situation. In particular, we find that modeling which user prefers to listen to which type of tracks in which situation via the interaction effects of both contexts is highly beneficial. Conducting a set of offline experiments using this model allows us to estimate the effects of both contextual dimensions. These experiments answered our fourth RQ, namely *what is the impact of different contextual dimensions to the recommendation actuary?* Incorporating situational context combined with playlist context into the user model allows us to compute 12% more accurate track recommendations compared to a model without any context. This shows that besides the *personalization aspect*, that is to find tracks a user likes, there is the equally important *suitability aspect* that has to be considered, which is to present the right track in the right moment.

Besides modeling situational context, we are also interested in whether there exist cultural music listing patterns allowing us to further refine our recommendations (cf. RQ5). By inventing a novel culture-aware user similarity based on socio-economic, cultural and acoustic features we are able to find cultural music listening patterns. In a second step, we leverage these patterns in our track recommender system. In particular, we extend our FM-based approach by a third contextual facet: the music-cultural embedding of a user. This allows us to further increase the recommendation precision by at least 11.49% compared to the situational- and acoustic context model and by 51.56% compared to FM-based collaborative filtering. Along with that, we observe that music-cultural clusters combined with situational clusters and/or musical preferences are highly beneficial, whereas considered in isolation cannot outperform baseline models. Hence, we argue that music-cultural context is only useful as *additional* contextual information. Even though these results are promising, we consider our research on cultural music information retrieval still as early work. In a future work, we want to conduct experiments on a larger dataset, integrate more culture-related variables (for instance Hofstede's cultural dimensions [48]) in order to refine the cluster computation and finally develop novel retrieval applications leveraging music-cultural context. I.e., for future work, we plan to use music-cultural context to integrate novelty and serendipity in our recommender system. In particular, a possible approach for integrating novelty would be to recommend tracks that are similar to the tracks a user likes but are unpopular in the culture the user is embedded in.

To conclude, this dissertation highlights the importance of considering different types of contextual information for music recommendation and retrieval. We introduce several techniques to mine user context and developed novel models to represent the user behavior on music streaming platforms. Along

with that, we stress the importance of culture-aware music information retrieval by showing the applicability for novel applications and use cases. From our point of view, multi-context and in particular culture-aware user models enable visionary music recommendation and exploration systems. Luckily, due to the rise of music streaming and social media, today an unprecedented amount of new (contextual) information is waiting to be exploited.

List of Figures

Bibliography

[1] G. Adomavicius, B. Mobasher, Francesco, and A. Tuzhilin. Context-aware Recommender Systems. *AI Magazine*, 32:67–80, 2011.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 7, pages 217–253. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[4] A. Ajesh, J. Nair, and P. S. Jijin. A random forest approach for rating-based recommender system. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1293–1297, 2016.

[5] M. Alghoniemy and A. H. Tewfik. A network flow model for playlist generation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2001)*, pages 329–332, 2001.

[6] C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.

[7] A. Ankolekar and T. Sandholm. Foxtrot: a soundtrack for where you are. In *Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications (IwS 2011)*, pages 26–31. ACM, 2011.

[8] J. S. Armstrong and F. Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, 8(1):69–80, June 1992.

[9] J.-J. Aucouturier and F. Pachet. Scaling up music playlist generation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, pages 105–108. IEEE, 2002.

[10] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[11] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In C. Huemer and T. Setzer, editors, *E-Commerce and Web Technologies*, volume 85 of *Lecture Notes in Business Information Processing*, pages 89–100. Springer, 2011.

[12] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, K.-H. Lüke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *Proceedings of the 12th International Conference on Electronic Commerce and Web Technologies-Web (EC-Web 2011)*, August 2011.

[13] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526, 2012.

[14] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, pages 301–304, 2011.

[15] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter - Special issue on visual analytics*, 9(2):75–79, Dec. 2007.

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

[17] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata. Higher-order factorization machines. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3351–3359. Curran Associates, Inc., 2016.

[19] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys 2010)*, pages 63–70, 2010.

[20] M. Braunhofer, M. Kaminskas, and F. Ricci. Recommending music for places of interest in a mobile travel guide. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, pages 253–256, New York, NY, USA, 2011. ACM.

[21] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 43–52, 1998.

[22] D. Brillinger. *Time Series: Data Analysis an Theory*. Holden-Day Series in Time Series Analysis. Holden-Day, 1981.

[23] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov. 2002.

[24] R. Cai, C. Zhang, C. Wang, L. Zhang, and W.-Y. Ma. Musicsense: Contextual music recommendation using emotional allocation modeling. In *Proceedings of the 15th ACM International Conference on Multimedia (MM 2007)*, 2007.

[25] P. Cano, M. Koppenberger, and N. Wack. Content-based music audio recommendation. In *Proceedings of the 13th ACM International Conference on Multimedia (MM 2005)*, pages 211–212, 2005.

[26] O. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[27] C.-M. Chen, M.-F. Tsai, J.-Y. Liu, and Y.-H. Yang. Using emotional context from article for contextual music recommendation. In *Proceedings of the 21st ACM International Conference on Multimedia (MM 2013)*, pages 649–652, 2013.

[28] Z. Cheng and J. Shen. Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *Proceedings of the 16th ACM International Conference on Multimedia Retrieval (ICMR 2014)*, 2014.

[29] C. Cleverdon and M. Kean. Factors determining the performance of indexing systems. Aslib Cranfield Research Project, Cranfield, England, 1968.

[30] S. J. Cunningham, D. Bainbridge, and A. Falconer. More of an art than a science': Supporting the creation of playlists and mixes. In *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR 2006)*, 2006.

[31] S. J. Cunningham, M. Jones, and S. Jones. Organizing Digital Music for Use: An Examination of Personal Music Collections. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, 2004.

[32] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis. A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701, 2007.

[33] E. Diener. Subjective well-being: The science of happiness and a proposal for a national index. *American Psychologist*, 55(1):34, 2000.

[34] J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, 2003.

[35] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (SIGKDD 1996)*, pages 226–231, 1996.

[36] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist generation using start and end songs. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2008)*, 2008.

[37] C. Freudenthaler, L. Schmidt-Thieme, and S. Rendle. Bayesian factorization machines. In *Proceedings of the NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011.

[38] A. Goel, M. Sheezan, S. Masood, and A. Saleem. Genre classification of songs using neural network. In *Preceedings of the 5th International Conference on Computer and Communication Technology (ICCCT 2014)*, pages 285–289. IEEE, 2014.

[39] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[40] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. , J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference (AAAI 1999)*, pages 439–446, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[41] B.-j. Han, S. Rho, S. Jun, and E. Hwang. Music emotion classification and context-based music recommendation. *Multimedia Tools and Applications*, 47(3):433–460, 2010.

[42] D. Hauger and M. Schedl. Exploring Geospatial Music Listening Patterns in Microblog Data. In *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR 2012)*, 2012.

[43] D. Hauger, M. Schedl, A. Košir, and M. Tkalčič. The Million Musical Tweets Dataset: What Can We Learn From Microblogs. In *Proceedings of the 14th International Symposium on Music Information Retrieval (ISMIR 2013)*, Curitiba, Brazil, November 2013.

[44] J. F. Helliwell, H. Huang, and S. Wang. The distribution of world happiness. *World Happiness*, page 8, 2016.

[45] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd International Conference on Research and Development in In-*

formation Retrieval (SIGIR 1999), pages 230–237, New York, NY, USA, 1999. ACM.

[46] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS 2004)*, 22(1):5–53, Jan. 2004.

[47] F. Hernández del Olmo and E. Gaudioso. Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35(3):790–804, Oct. 2008.

[48] G. Hofstede, G. J. Hofstede, and M. Minkov. *Cultures and Organizations: Software of the Mind*, volume 3. McGraw-Hill, 2010.

[49] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pages 263–272, 2008.

[50] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[51] N. Japkowicz. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In *Papers from the Association for the Advancement of Artificial Intelligence Workshop (AAAI Technical Report WS-00-05)*, pages 10–15, 2000.

[52] I. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[53] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys 2016)*, pages 43–50. ACM, 2016.

[54] M. Kamalzadeh, D. Baur, and T. Mller. A survey on music listening and management behaviours. In *Proceedings of the 13th International Symposium on Music Information Retrieval (ISMIR 2012)*, 2012.

[55] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In *User Modeling, Adaption and Personalization*, pages 183–194. Springer Berlin Heidelberg, 2011.

[56] M. Kaminskas and F. Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2):89–119, 2012.

[57] M. Kaminskas, F. Ricci, and M. Schedl. Location-aware music recommendation using auto-tagging and hybrid matching. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys 2013)*, pages 17–24, 2013.

[58] D. Kim and B.-J. Yum. Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, 28(4):823–830, May 2005.

[59] J.-Y. Kim and N. J. Belkin. Categories of music description and search terms and phrases used by non-music experts. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR 2002)*, volume 2, pages 209–214, 2002.

[60] P. Knees, M. Schedl, T. Pohle, K. Seyerlehner, and G. Widmer. Supervised and Unsupervised Web Document Filtering Techniques to Improve Text-Based Music Retrieval. In *Proceedings of the 11th International Symposium on Music Information Retrieval (ISMIR 2010)*, Utrecht, the Netherlands, August 2010.

[61] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2008)*, pages 426–434, 2008.

[62] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2009)*, pages 447–456, New York, NY, USA, 2009. ACM.

[63] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer Journal*, 42(8), 2009.

[64] J. H. Lee and J. S. Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, volume 2004, page 5th, 2004.

[65] J. H. Lee, Y.-S. Kim, and C. Hubbles. A look at the cloud from both sides now: an analysis of cloud music service usage. In *Proceedings of the 17th International Symposium on Music Information Retrieval (ISMIR 2016)*, 2016.

[66] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 5th International Conference on Data Mining (SIAM 2005)*, 2005.

[67] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764 – 766, 2013.

[68] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[69] C. X. Ling and C. Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD 1998)*, pages 73–79, 1998.

[70] B. Logan. Content-Based Playlist Generation: Exploratory Experiments. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR 2002)*, 2002.

[71] D. Lübbers and M. Jarke. Adaptive multimodal exploration of music collections. In *Proceedings of the 10th International Symposium on Music Information Retrieval (ISMIR 2009)*, pages 195–200, 2009.

[72] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (1967)*, pages 281–297, 1967.

[73] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18:50–60, 1947.

[74] K. Mardia, J. Kent, and J. Bibby. *Multivariate Analysis*. Probability and mathematical statistics. Acad. Press, London [u.a.], 1979.

[75] B. McFee, L. Barrington, and G. Lanckriet. Learning content similarity for music recommendation. *IEEE transactions on audio, speech, and language processing*, 20(8):2207–2218, 2012.

[76] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web (WWW 2012)*, pages 909–916, New York, NY, USA, 2012. ACM.

[77] A. N. Mikhail Trofimov. tffm: Tensorflow implementation of an arbitrary order factorization machine. `https://github.com/geffy/tffm`, 2016.

[78] A. Mild and T. Reutterer. Collaborative filtering methods for binary market basket data analysis. In *Proceedings of the 6th International Computer Science Conference on Active Media Technology (AMT 2001)*, pages 302–313, 2001.

[79] A. Mild and T. Reutterer. An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services*, 10(3):123 – 133, 2003. Model Building in Retailing and Consumer Services.

[80] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[81] A. v. d. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, pages 2643–2651, USA, 2013. Curran Associates Inc.

[82] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In *Proceedings of the 8th International Conference Machine Learning and Data Mining in Pattern Recognition (MLDM 2012)*, pages 154–168, 2012.

[83] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pages 502–511, 2008.

[84] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(6):559–572, 1901.

[85] M. Pichl, E. Zangerle, and G. Specht. Combining Spotify and Twitter Data for Generating a Recent and Public Dataset for Music Recommendation. In *Proceedings of the 26nd Workshop Grundlagen von Datenbanken (GvDB 2014)*, pages 35–40, 2014.

[86] M. Pichl, E. Zangerle, and G. Specht. #nowplaying on #Spotify: Leveraging Spotify Information on Twitter for Artist Recommendations. In *Proceedings of the 2nd International Workshop on Mining the Social Web in conjunction with the 15th International Conference on Web Engineering (ICWE 2015)*, 2015.

[87] M. Pichl, E. Zangerle, and G. Specht. Towards a context-aware music recommendation approach: What is hidden in the playlist name? In *Proceedings of the 15th IEEE International Conference on Data Mining Workshops (ICDM 2015)*, pages 1360–1365, 2015.

[88] M. Pichl, E. Zangerle, and G. Specht. Understanding Playlist Creation on Music Streaming Platforms. In *Proceedings of the 18th IEEE Symposium on Multimedia (ISM 2016)*, pages 475–480, 2016.

[89] M. Pichl, E. Zangerle, and G. Specht. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In *Proceedings of the 7th ACM on International Conference on Multimedia Retrieval (ICMR 2017)*. ACM, 2017.

[90] M. Pichl, E. Zangerle, and G. Specht. Understanding user-curated playlists on spotify: A machine learning approach. *International Journal of Multimedia Data Engineering and Management IJMDEM*, 8(4), 2017.

[91] M. Pichl, E. Zangerle, G. Specht, and M. Schedl. Mining Culture-Specific Music Listening Behavior from Social Media Data. In *Proceedings of the IEEE Symposium on Multimedia (ISM)*, pages 208–215, 2017.

[92] S. Rendle. Factorization machines. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2010)*, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.

[93] S. Rendle. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST 2012)*, 3(3):57:1–57:22, May 2012.

[94] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[95] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 635–644. ACM, 2011.

[96] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 81–90, New York, NY, USA, 2010. ACM.

[97] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 5th ACM Conference on Computer Supported Cooperative Work (CSCW 1994)*, pages 175–186, 1994.

[98] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, Mar. 1997.

[99] S. Rho, B.-j. Han, and E. Hwang. Svr-based music mood classification and context-based music recommendation. In *Proceedings of the 17th ACM International Conference on Multimedia (MM 2009)*, pages 713–716, 2009.

[100] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[101] A. Said and A. Bellogín. Comparative recommender system evaluation: Benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys 2014)*, pages 129–136, New York, NY, USA, 2014. ACM.

[102] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 880–887, New York, NY, USA, 2008. ACM.

[103] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[104] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW 2001)*, pages 285–295, 2001.

[105] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems: A case study. In *Proceedings of the WebKDD Workshop at the 6th ACM International*

Conference on Knowledge Discovery and Data Mining (SIGKDD 2000), 2000.

[106] M. Schedl. The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR 2016)*, New York, USA, June 2016.

[107] M. Schedl. Investigating country-specific music preferences and music recommendation algorithms with the LFM-1b dataset. *International Journal of Multimedia Information Retrieval*, 6(1):71–84, 2017.

[108] M. Schedl, A. Flexer, and J. Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, 2013.

[109] M. Schedl, E. Gómez, and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3):127–261, Sept. 2014.

[110] M. Schedl, D. Hauger, and J. Urbano. Harvesting microblogs for contextual music similarity estimation - a co-occurrence-based framework. *Multimedia Systems*, 20(6):693–705, May 2013.

[111] M. Schedl, T. Pohle, P. Knees, and G. Widmer. Exploring the Music Similarity Space on the Web. *Transactions on Information Systems*, 29(3), July 2011.

[112] M. Schedl and D. Schnitzer. Hybrid retrieval approaches to geospatial music recommendation. In *Proceedings of the 35th International Conference on Research and Development in Information Retrieval (SIGIR 2013)*. ACM, 2013.

[113] M. Schedl and D. Schnitzer. Location-Aware Music Artist Recommendation. In *Proceedings of the 20th International Conference on MultiMedia Modeling (MMM 2014)*, Dublin, Ireland, January 2014.

[114] M. Schedl, A. Vall, and K. Farrahi. User Geospatial Context for Music Recommendation in Microblogs. In *Proceedings of the 37th International Conference on Research and Development in Information Retrieval (SIGIR 2014)*, 2014.

[115] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th*

International Conference on Research and Development in Information Retrieval (SIGIR 2002), pages 253–260, 2002.

[116] U. Schimmack, P. Radhakrishnan, S. Oishi, V. Dzokoto, and S. Ahadi. Culture, personality, and subjective well-being: integrating process models of life satisfaction. *Journal of Personality and Social Psychology*, 82(4):582, 2002.

[117] A. Schindler and A. Rauber. Capturing the temporal domain in echonest features for improved classification effectiveness. In *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR 2012)*, pages 214–227, 2014.

[118] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 8, pages 257–297. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[119] M. Slaney and W. White. Measuring playlist diversity for recommendation systems. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing for Multimedia (AMCMM 2006)*, pages 77–82, 2006.

[120] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[121] G. Specht and T. Kahabka. Information filtering and personalisation in databases using gaussian curves. In *Proceedings of the 4th International Database Engineering and Applications Symposium (IDEAS 2000)*, pages 16–24, 2000.

[122] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[123] A. van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.

[124] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[125] E. Vorhees. The trec-8 question answering track report. In *Proceedings of TREC-8*, 1999.

[126] X. Wang, D. Rosenblum, and Y. Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the ACM International conference on Multimedia (MM 2012)*, pages 99–108. ACM, 2012.

[127] X. Wang and Y. Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22Nd ACM International Conference on Multimedia (MM 2014)*, pages 627–636, New York, NY, USA, 2014. ACM.

[128] E. Zangerle, W. Gassler, and G. Specht. Exploiting twitter's collective knowledge for music recommendations. In *Proceedings of the 2nd Workshop on Making Sense of Microposts (#MSM2012)*, 2012.

[129] E. Zangerle, M. Pichl, W. Gassler, and G. Specht. #nowplaying music dataset: Extracting listening behavior from twitter. In *Proceedings of the 1st ACM International Workshop on Internet-Scale Multimedia Management (ISMM 2014)*, pages 21–26, New York, NY, USA, 2014. ACM.

[130] H.-R. Zhang and F. Min. Three-way recommender systems based on random forests. *Knowledge-Based Systems*, 91:275 – 286, 2016.