universität
innsbruck

Department of
Computer Science

Cumulative Habilitation Thesis

Recommender Systems for Music Retrieval Tasks

Dr. Eva Zangerle

Faculty of Mathematics, Computer Science and Physics

December 2022

... to the Zagal family.

Contents

Part I.

Preface

1. Introduction

Music is ubiquitous in today's world—almost everyone enjoys listening to music. With the rise of streaming platforms, listeners now have access to more music than ever before. While users may seemingly benefit from this plethora of available music, at the same time, it has increasingly made it more difficult for users to explore and discover new music they like. Personalized access to music libraries and music recommender systems aim to help users discover and retrieve music they like and enjoy.

To this end, the field of *Music Information Retrieval (MIR)* strives to make music accessible to all by advancing retrieval applications such as music recommender systems, content-based search, the generation of personalized playlists, or user interfaces that allow visually exploring music collections [126, 414]. This includes tasks such as gathering machine-readable musical data, extracting meaningful features, developing data representations based on these features, and methodologies to process and understand this data [169, 414]. Retrieval approaches specifically leverage these representations for indexing music and providing search and retrieval services.

Personalized music retrieval and recommendation require incorporating information about users and their preferences into retrieval and recommendation algorithms. Most importantly, such user-centric MIR approaches need to capture aspects that influence the user's perception of music, a factor highly relevant to a user's preference for music. Aspects that influence human perception of music include music content (descriptors extracted from the audio signal, such as tempo or acousticness), music context (external factors describing the track or artist such as a track's lyrics), user properties (comparatively stable, long-term descriptors of the user, such as general music preferences), and user context (short-term, dynamic factors describing the user, such as the current activity, occasion or emotional state) [413, 416]. These aspects are typically captured by a *user model* [251]. To compute items that best meet the user's needs and preferences, these user models are compared with *item models*, which capture the characteristics of individual items.

Current user models for personalized retrieval tasks are typically modeled rather simplistically [10, 413], mostly focusing on single aspects of the user—for instance, individual contexts such as the user's current location [27, 96, 233, 234] or mood [33, 63, 178, 381]. As Schedl et al. [413] note, comprehensive user models are rare in MIR. Consequently, there is not only a lack of comprehensive user models but also a lack of retrieval and recommendation approaches that allow integrating and combining multi-faceted user and item models.

This habilitation thesis contributes to the field of (music) recommender systems in the following aspects: (1) We present novel comprehensive user and item models to capture characteristics of users, their context, and musical items. (2) We jointly leverage these user and item models in newly designed context-aware recommender systems. (3) We investigate biases and potential unfairness in state-of-the-art recommender systems based on the proposed multi-faceted user and item models. (4) We propose an evaluation framework to conceptualize the recommendation evaluation space, enabling a comprehensive assessment of the factors influencing recommendation performance.

The remainder of this habilitation thesis is structured as follows. Section 2 provides an overview of current approaches in the fields of music recommender systems and user modeling and identifies open challenges in these areas. In Section 3, we summarize the main contributions of the publications featured in this habilitation thesis and show how they address the challenges identified. In Part II, we present the selected papers as the core of this habilitation thesis.

2. Background and Open Challenges

In this chapter, we present background and related work within the scope of this habilitation thesis. Furthermore, we identify open challenges that are (partly) addressed in the scientific contributions of this thesis.

We first discuss core recommender systems, before focusing on user models for recommender systems and fairness concerns in recommender systems research. Lastly, we discuss the evaluation of recommender systems.

2.1. Recommender Systems

The main goal of recommender systems is to provide their users with personalized item recommendations [190, 384], helping them to deal with the choice overload problem [57]. The two fundamental algorithmic approaches toward computing recommendations are collaborative filtering and content-based recommender systems.

The most widely used approach for computing recommendations is collaborative filtering [402, 403], which assumes that users who had similar preferences in the past will also do so in future. The collective preferences of the user base are modeled in the so-called user-item matrix. This matrix captures users' past interactions with items, where interactions may be explicit ratings of items (e.g., on a scale from 0 to 5) or implicit actions like product views. The algorithmic task carried out is mainly that of matrix completion—predicting the missing ratings in the matrix (based on e.g., neighborhood-based approaches [335], matrix factorization [263], or machine- and deep-learning approaches [517]). For an overview of collaborative filtering approaches, we refer to [64, 132, 190, 402, 403].

Content-based recommender systems aim to recommend items that are similar to items the user has liked in the past [7, 328, 348]. Therefore, items are characterized by a set of features (for instance, in the movie domain, these features could be genre, actors, or the plot of the movie). Based on the features of items a user has liked, a user profile is built. Recommendations can then be obtained by computing the similarity between the given user profile and items; recommending the most similar, yet new and relevant, items to the user. Hybrid recommender systems aim to combine the advantages of collaborative filtering and content-based approaches [67].

Both of these types of approaches rely on rather simple user models and ignore that users interact with the system in specific contexts [10]. In contrast, context-aware recommender systems [9] tailor recommendations to the contextual situation of the user. Kaminskas and Ricci [232] categorize context information as follows: environment-related context (e.g., location, time, weather), user-related context (e.g., activity, demographic information, emotional state of the user), and multimedia context (e.g., text or pictures the user is currently viewing). The three algorithmic paradigms for computing context-aware recommendations are contextual pre- or post-filtering and contextual modeling [9]. Contextual pre- and post-filtering rely on traditional models (mostly, collaborative filtering models) and add a further data filtering step either before or after the actual recommendation computation (e.g., for recommendations for a given context, pre-filtering would remove all data collected in another context and use the remaining data as input to the recommender system). In contrast, contextual modeling incorporates context information directly into the model.

One of the main open challenges is to integrate available features describing users, items, and contexts into personalized (music) retrieval systems [413]. This involves not only comprehensive user and item models to characterize users, items, and contexts more accurately, but also requires recommendation algorithms to integrate and jointly leverage these models. Particularly for context-aware recommender systems, the main challenges, as Adomavicius et al. [9] note, are in (1) incorporating context information into canonical recommender systems, (2) leveraging context and hidden interactions between users, items, and context for improved rating prediction, and (3) identifying relevant contextual factors. Particularly, incorporating multiple relevant contextual signals into a music recommender system is still an open challenge [413].

2.2. Multi-faceted User and Item Models

To this end, recommender systems mostly rely on interaction data and neglect further information about the user and the items. While these systems are undeniably highly successful and have reshaped the landscape of recommender systems, they are agnostic to any further data on users and items (so-called side information [337]). For instance, we listen to music because of its content, which may include rhythmic, timbral, and lyrical qualities, or because of the emotions the music evokes in us. Also, we perceive music on different levels of semantics that comprise, for instance, the listened audio signal, but also textual or visual input [325]. These qualities are not considered in many of today's music recommender systems.

The facets that allow for describing users and items in the music domain more comprehensively, and hence, capturing the drivers of preference, are manyfold. Schedl et al. [413] organize the factors influencing human music perception into (1) music content, (2) music context, (3) user context, and (4) user properties. Music content refers to features that can be extracted and retrieved from the audio signal (from low-level frequency bands or Mel-frequency cepstrum coefficients (MFCCs) to high-level features such as danceability or tempo). Music context describes features that are not directly tied to the audio signal. This includes, for instance, the lyrics of a song, or further information about the artist's background. User context refers to dynamic factors that describe the user—the user's current emotional state, their current activity, or their spatio-temporal context. User properties, on the other hand, are more stable descriptors of a user, capturing the user's music taste, demographics, cultural background, personality, or musical experience.

The major challenges for user and item models are the elicitation and modeling of all four factors influencing human music perception [414]. However, user models in (music) information retrieval and recommender systems are very simplistic [10, 413]—despite the fact that a rich set of item descriptors is available in the music domain. Schedl et al. [423] consider incorporating psychological constructs such as personality and emotion as a central future direction of music information retrieval. They furthermore note the challenge of situation-aware music recommender systems, which require a multi-faceted user model to describe contextual and situational preferences (as also discussed in Section 2.1). A further challenge for the personalization of music recommendations is the fact that the way we perceive music is also shaped by our cultural background, which requires music recommender systems to also personalize on a cultural level [423].

2.3. Fairness in Recommender Systems

Beyond the core performance of recommender systems, there are also further factors that need to be considered in the recommender systems ecosystem. Among those factors, the concept of fairness is a highly significant one. Fairness captures whether a recommender system does not discriminate on the individual or on the group level, or, put differently, whether all individuals and groups (regardless of age, race, gender, etc.) are served with recommendations of the same quality [134, 483]. Assessing fairness also requires identifying the different stakeholders of a recommender system (from end-users to platform providers or, for instance, artists) to integrate the stakeholders' perspectives and fairness concerns [2].

From a data perspective, biases (and potential resulting unfairness) are introduced if we work on data that is not representative of the full population. Chen et al. [89] distinguish four types of data biases in the process of collecting data: (1) selection bias: users are free to choose the items they rate, resulting in a non-representative sample of ratings; (2) exposure bias: users are only exposed to a fraction of all available items and hence, any unobserved user-item interaction does not necessarily represent a negative preference; (3) conformity bias: users tend to behave similarly to other users, therefore their feedback does not necessarily reflect their true preferences; and (4) position bias: users tend to prefer items at a higher position in the list of recommendations irrespective of the item's real relevance.

From an algorithmic perspective, recommender systems may suffer from biases at the user and item level (Chen et al. [89] consider these as biases in results). On the item level, items from the long tail (i.e., unpopular items with a low number of user interactions) are recommended less frequently as recommender algorithms favor well-known items [4, 78, 279], known as popularity bias [3, 4]. This effect (also referred to as the Matthews effect) is further amplified by the recommender system. On the user level, biases have been shown to provide, for instance, users of a certain gender [145], age group [138], or country [415] with less accurate recommendations.

The major open challenge in this context is that we lack an understanding of the potential biases involved and their impact on the recommender system. As Burke et al. note [70], the standard procedure for establishing fairness in bias and fairness research is to first identify discriminated individuals and groups and subsequently develop algorithms that remove the identified bias. Therefore, the first step is to get a deeper understanding of groups that are discriminated against. Such an understanding is the prerequisite for designing recommender systems that allow mitigating these biases by carefully considering all stakeholders of the system [2, 68, 438]. Suitable metrics also need to be developed to formalize and quantify system fairness during the development and evaluation process [438].

2.4. Evaluation of Recommender Systems

Evaluation of a system in development is crucial throughout the lifetime of a system—from initial prototypes, to deployment, maintenance, and updates. Jannach et al. [217] describe recommender systems evaluations as “methods for choosing the best technique based on the specifics of the application domain, identifying influential success factors behind different techniques, or comparing several techniques based on an optimality criterion”.

In the evaluation of recommender systems, we differentiate three types of experiments: offline experiments, user studies, and online experiments [42, 44, 154, 174, 190]. Offline experiments are based on a historic, pre-collected dataset of user-item interactions. User behavior is simulated by removing parts of the interactions (test set), computing recommendations on the remaining data (training set) and subsequently comparing the predicted ratings to the original ratings. Offline experiments aim to compare recommendation algorithms and settings and focus on system-centric aspects [174, 190, 399]. In contrast, user studies are conducted by recruiting human users who perform pre-defined tasks in a laboratory setting. By observing the users’ interactions with the system and collecting direct feedback (e.g., via surveys before, during, or after the task), the user experience with the system is evaluated—including interaction behavior. The third experiment type, online experiments, are deployed in a real-world setting, where users perform self-selected tasks. These experiments allow for evaluating realistic scenarios by recording user interactions and collecting direct feedback. Hence, user studies and online experiments can be considered user-centric evaluations [174, 190, 399].

The major challenges we (still) face in evaluating recommender systems are in performing rigorous, systematic, and hypothesis-driven evaluations [25, 143, 207, 448]. In today's scientific practice, mostly a single experiment type is applied (mainly offline experiments). This substantially limits the generalizability of the obtained results and may leave further aspects unclear. Particularly, Jannach et al. [214] argue that "we should more often follow a research approach that is guided by clear and explicit hypotheses. These hypotheses then determine the experimental design and in particular the used metrics." This also requires the application of holistic and comprehensive evaluation protocols that extend beyond offline evaluations and that integrate multiple system- and user-centric perspectives and experiments [104, 254].

3. Contributions

The following chapter presents an overview of the papers contained in this habilitation thesis. We describe the contributions of the individual papers, particularly regarding the extent to which these works contribute to the problems and challenges outlined in Section 2. The papers are grouped thematically by the challenges addressed (as introduced in Section 2). Finally, we present further publications not included in the thesis; albeit most of these publications also contribute to the fields of recommender systems and music information retrieval.

3.1. Recommender Systems

Publications

[C1] M. Pichl, E. Zangerle, and G. Specht. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 201–208. ACM, 2017. DOI: 10.1145/3078971.3078980

[C2] M. Pichl and E. Zangerle. User models for multi-context-aware music recommendation. *Multimedia Tools and Applications*, 80(15):22509–22531, 2021. DOI: 10.1007/s11042-020-09890-7

Note that [C2] is an extended version of the following publication[1]:

[C3] M. Pichl and E. Zangerle. Latent Feature Combination for Multi-Context Music Recommendation. In *2018 International Conference on Content-Based Multimedia Indexing*, CBMI '18, pages 1–6. IEEE, 2018. DOI: 10.1109/CBMI.2018.8516495

These works all aim to improve context-aware recommender systems by incorporating context information directly in the recommendation algorithm in a contextual modeling approach, allowing to incorporate one or multiple contexts to further tailor recommendations towards the user, their preferences, and their current context.

[1] This paper won the Best Student Paper award at the International Conference on Content-Based Multimedia Indexing (CBMI) 2018.

In [C1] (Chapter 4), we advance context-aware music recommender systems by proposing a recommendation approach that directly incorporates contextual information in the computation of recommendations. In contrast to the widely used pre-filtering approaches, this approach does not require filtering user-item interactions based on the user's context and hence, allows using the full set of interactions for the computation of recommendations. We extract situational information from user playlists and their names (e.g., "party", "workout", or "my summer playlist") by extracting clusters of contextually similar tracks. This allows modeling all interactions of users with the system by <user, track, cluster, rating> vectors. We use these vectors as input for our recommender system and employ Factorization Machines for the prediction of ratings, which extend traditional factorization approaches by also factorizing the interaction of variables into a lower-dimensional space. The main contribution of this work is that it extends the previously prevalent pre-filtering approaches for context-aware recommender systems by incorporating context information directly into the core recommendation approach to leverage all information available. We show that our proposed factorization machine-based recommender system substantially outperforms context-agnostic recommender systems, pre-filtering context-aware recommender systems as well as classification-based context-aware recommender systems.

[C2] (Chapter 5) further extends the work in [C1] by proposing to simultaneously leverage multiple user and item contexts in a multi-context-aware recommender system. Our approach leverages situational and acoustic context information to describe users and items. We extract the situational context from playlist names (see also [C1]) to capture the situation in which certain tracks are listened to by a user. To further capture a user's music preferences, we compute musical archetypes by clustering tracks via their high-level acoustic features to describe a user's inclination and preference for such archetypes. Notably, we investigate how contextual and audio characteristics (captured by the proposed multi-context user model) may jointly be leveraged for track recommendations. The main contributions of this paper are as follows: We propose a multi-context-aware user model and recommender system that allows capturing a user's preference towards certain archetypes of music (acoustic context) and contexts in which users listen to certain tracks (situational context). We exploit interaction effects between the input variables (user listening history, acoustic feature-based playlist archetypes, and situational context) by introducing Factorization Machines for the task; i.e., we model the influence of a certain context on the choice of tracks for a given user. In several experiments, we show that a recommender system leveraging this proposed model substantially outperforms a context-aware recommender system that relies on either context- or acoustic feature-based clusters individually.

3.2. Multi-faceted User and Item Models

Publications

[C4] E. Zangerle and M. Pichl. Content-based User Models: Modeling the Many Faces of Musical Preference. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, ISMIR '18, pages 709–716. ISMIR, 2018. DOI: 10.5281/zenodo.1492515

[C5] E. Zangerle, C.-M. Chen, M.-F. Tsai, and Y.-H. Yang. Leveraging Affective Hashtags for Ranking Music Recommendations. *IEEE Transactions on Affective Computing*, 12(1):78–91, 2021. DOI: 10.1109/TAFFC.2018.2846596

[C6] E. Zangerle, M. Pichl, and M. Schedl. User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues. *Transactions of the International Society for Music Information Retrieval*, 3(1), 2020. DOI: 10.5334/tismir.37

All of these publications propose rich, multi-faceted user models to describe users and their (context-specific) preferences more accurately, ultimately improving recommendation performance.

In [C4] (Chapter 6), we particularly address the lack of comprehensive content-based user models. We introduce a set of user models that capture the musical preferences of users by using content descriptors of tracks that a user has listened to. These user models aim to capture not only the overall musical preferences of users. To describe users and their musical preferences, we capture the musical preferences of users via content descriptors of tracks. This is one of the rare works that aim to devise comprehensive user models based on content descriptors. Most notably, we model user preferences probabilistically by employing Gaussian mixture models. Our experiments show that a user model based on a user's specific preferences regarding different types of music models via a Gaussian mixture model, complemented by a user's general musical preferences achieves the best results.

In [C5] (Chapter 7)[2], we leverage tweets that describe the track a user is currently listening to and also feature a hashtag describing their emotional state (for instance, in "#nowplaying Crazy For You by Adele #Happy"). We study the impact of adding affective context information on ranking contextual affection-aware music recommendations tailored to the user's current emotional state and musical preferences. Therefore, we propose modeling users, tracks, and affective hashtags in a graph and computing a latent,

[2] This manuscript won the Women in RecSys Journal Paper of the Year Award at the 16th ACM Conference on Recommender Systems in 2022.

low-dimensional representation for each node by applying a graph embedding method. Based on these representations, we propose several novel ranking methods. We show that in a ranking task, comparing the latent representations of users and tracks is sufficient to capture the user's general preferences. However, for context-aware ranking of recommendations based on the user's current affective context, sentiment information extracted from the user's hashtags is vital and contributes to improved, personalized rankings and hence, recommendations. The contributions of this work lie in novel ranking methods that integrate affective information extracted from social media. Particularly, we learn user, item, and context representations by employing graph embedding methods and incorporating these into the proposed ranking methods. Furthermore, we propose an evaluation setup that allows us to investigate the extent to which a ranking or recommender algorithm is able to capture the general preferences compared to the context-specific preferences of users. Furthermore, our study is based on large-scale, real-world data, whereas existing work mostly relies on laboratory experiments, with low to medium sample sizes.

[C6] (Chapter 8) extends content-based user models by introducing socio-economic and cultural aspects to the user model to also capture the cultural backgrounds of listeners. Consequently, it becomes possible to uncover music-cultural patterns of listening that describe the interrelationship between users, their cultural background, and the characteristics of the music they listen to. Particularly, we propose a novel music-cultural user modeling approach that allows leveraging music-cultural listening patterns in a recommender system. Therefore, we integrate information about the acoustic qualities of the music users have listened to, and culture-specific information derived from the users' location/country to describe the user's likely cultural background. The two main contributions of this work are: (1) We jointly use acoustic song features and culture-related features to describe the user's musical preferences and cultural background, and (2) we utilize these features in a culture-aware user model and show their contribution to performance in a music recommendation task based on a dataset of 55k users and 395k listening events.

3.3. Fairness in Recommender Systems

Publications

[C7] D. Kowald, P. Müllner, E. Zangerle, C. Bauer, M. Schedl, and E. Lex. Support the underground: characteristics of beyond-mainstream music listeners. *EPJ Data Science*, 10(1):14, 2021. DOI: 10.1140/epjds/s13688-021-00268-9

[C8] A. B. Melchiorre, E. Zangerle, and M. Schedl. Personality Bias of Music Recommendation Algorithms. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 533–538. ACM, 2020. DOI: 10.1145/3383313.3412223

These publications both contribute to our understanding of potential fairness issues and biases in recommender systems.

In [C7] (Chapter 9) we investigate the characteristics of beyond-mainstream music and its listeners and the quality of music recommendations served to beyond-mainstream listeners to inform future user modeling and recommendation tasks. Therefore, we capture a user's mainstreaminess at the artist level by computing the correlation between a user's artist playcount and the global playcount per artist. In an exploratory study, we investigate the characteristics of beyond-mainstream music, its listeners, and compare the recommendation performance for mainstream- and non-mainstream listeners. The main contributions of this work are as follows: (1) We show that recommendations provided to beyond-mainstream music listeners are of significantly lower recommendation accuracy than those served to mainstream music listeners; (2) based on a novel dataset, we identify different types of beyond-mainstream music based on their acoustic features; (3) we identify subgroups of beyond-mainstream music listeners and investigate the relationship between openness and diversity of these subgroups and the recommendation accuracy for these groups.

In [C8] (Chapter 10), we investigate to which extent state-of-the-art recommender algorithms are prone to personality bias (i.e., providing recommendations of different quality to user groups with different personality traits). Particularly, we analyze the performance of these algorithms and how it differs across user groups in terms of personality traits. This is particularly interesting as personality traits have been shown to correlate with music preference and usage of music. We describe the personality of users by the OCEAN model, which describes personality traits along five dimensions: openness to experience (conventional vs. creative thinking), conscientiousness (disorganized vs. organized behavior), extraversion (engagement with the external world), agreeableness (need for social harmony), and neuroticism (emotional instability). With this work, we advance our understanding of personality bias by evaluating a set of state-of-the-art recommendation approaches for user groups exhibiting different personality profiles. The main finding of this analysis is that there are indeed statistically significant differences in terms of accuracy metrics (recall@k, NDCG@k) for all traits. We observe particularly pronounced differences in the traits of neuroticism and openness.

3.4. Evaluation of Recommender Systems

Publications

[C9] E. Zangerle and C. Bauer. Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 2022. ISSN: 0360-0300. DOI: 10.1145/3556536

In [C9] (Chapter 11), we survey and consolidate the current state of the art in recommender systems evaluation. Most importantly, we propose the Framework for EValuating Recommender systems (FEVR) that conceptualizes the evaluation space of recommender systems evaluation. With FEVR, we categorize the evaluation design space and provide a systematic overview of the essential aspects of RS evaluation and their application. The proposed FEVR framework encompasses a wide variety of facets required when evaluating recommender systems and can also accommodate comprehensive evaluations that address the various multi-faceted dimensions. FEVR provides a structured basis for adopting and describing appropriate evaluation configurations, as well as integrating multiple evaluations and accounting for repeatability and reproducibility. It provides a guide for systematic RS evaluation that the RS research community can build on.

3.5. Synopsis

This habilitation thesis addresses the topic of recommender systems for music retrieval tasks. We have identified four distinct challenges in Chapter 2: core recommendation algorithms, multi-faceted user and item models, fairness in recommender systems, and the evaluation of recommender systems. Addressing the first challenge, we have contributed contextual modeling recommender algorithms that allow us to directly integrate multiple contexts. In the field of multi-faceted user and item models, we have contributed novel data sources that allow incorporating affective and cultural contexts, and also proposed novel modeling techniques for rich user and item characteristics. In the context of fairness in recommender systems, we have deepened our understanding of two potential user characteristics that may lead to biases: user personality and their tendency to listen to mainstream music. This understanding is the prerequisite for mitigating such biases and resulting unfairness in the next step. As a final contribution, we propose a framework for conceptualizing the design space of recommender systems evaluations based on a survey on recommender system evaluation.

3.6. Further Contributions

Since receiving my PhD in 2013, I have also co-authored the following peer-reviewed publications which are not included in this habilitation thesis.

[F1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, and E. Zangerle. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, Style Change Detection, and Trigger Detection. In *Advances in Information Retrieval - 44th European Conference on IR Research*, ECIR '22, pages 331–338. Springer, 2022. DOI: 10.1007/978-3-030-99739-7_42.

[F2] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, and E. Zangerle. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 13th International Conference of the CLEF Association*, volume 13390 of *LNCS*, pages 382–394. Springer, 2022. DOI: 10.1007/978-3-031-13643-6_24.

[F3] R. Binna, E. Zangerle, M. Pichl, G. Specht, and V. Leis. Height Optimized Tries. *ACM Transactions on Database Systems*, 47(1), 2022. DOI: 10.1145/3506692.

[F4] M. Moosleitner, G. Specht, and E. Zangerle. Co-rating Attacks on Recommendation Algorithms. In *Proceedings of the 32nd GI-Workshop Grundlagen von Datenbanksysteme (GvDB'21)*, volume 3075 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022.

[F5] M. Moscati, E. Parada-Cabaleiro, Y. Deldjoo, E. Zangerle, and M. Schedl. Music4All-Onion – A Large-Scale Multi-Faceted Content-Centric Music Recommendation Dataset. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pages 4339–4343. ACM, 2022. DOI: 10.1145/3511808.3557656.

[F6] M. Vötter, M. Mayerl, G. Specht, and E. Zangerle. HSP Datasets: Insights on Song Popularity Prediction. *International Journal of Semantic Computing*:1–23, 2022. DOI: 10.1142/S1793351X22400104.

[F7] E. Zangerle, C. Bauer, and A. Said. Report on the 1st Workshop on the Perspectives on the Evaluation of Recommender Systems (PERSPECTIVES 2021) at RecSys 2021. *SIGIR Forum*, 55(2), 2022. DOI: 10.1145/3527546.3527565.

[F8] E. Zangerle, M. Mayerl, M. Potthast, and B. Stein. Overview of the Style Change Detection Task at PAN 2022. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum*, volume 3180 of *CEUR Workshop Proceedings*, pages 2344–2356. CEUR-WS.org, 2022.

[F9] J. Bevendorff, B. Chulvi, G. L. D. la Peña Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, and E. Zangerle. Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association*, volume 12880 of *LNCS*, pages 419–431. Springer, 2021. DOI: 10.1007/978-3-030-85251-1_26.

[F10] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, and E. Zangerle. Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection. In *Advances in Information Retrieval – 43rd European Conference on IR Research*, ECIR '21, pages 567–573. Springer, 2021.

[F11] M. Vötter, M. Mayerl, G. Specht, and E. Zangerle. Novel Datasets for Evaluating Song Popularity Prediction Tasks. In *IEEE International Symposium on Multimedia*, ISM '21, pages 166–173. IEEE, 2021. DOI: 10.1109/ISM52913.2021.00034.

[F12] E. Zangerle, C. Bauer, and A. Said. *Perspectives on the Evaluation of Recommender Systems (PERSPECTIVES)*. In *Proceedings of the 15th ACM Conference on Recommender Systems*. RecSys '21. ACM, 2021, pages 794–795. DOI: 10.1145/3460231.3470929.

[F13] E. Zangerle, M. Mayerl, M. Potthast, and B. Stein. Overview of the Style Change Detection Task at PAN 2021. In *CLEF 2021 Labs and Workshops, Notebook Papers*, CEUR Workshop Proceedings, pages 1760–1771. CEUR-WS.org, 2021.

[F14] J. Bevendorff, B. Ghanem, A. Giachanou, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Wiegmann, and E. Zangerle. Overview of PAN 2020: Authorship Verification, Celebrity Profiling, Profiling Fake News Spreaders on Twitter, and Style Change Detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association*, volume 12260 of *LNCS*, pages 372–383. Springer, 2020.

[F15] J. Bevendorff, B. Ghanem, A. Giachanou, M. Kestemont, E. Manjavacas, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Wiegmann, and E. Zangerle. Shared Tasks on Authorship Analysis at PAN 2020. In *Advances in Information Retrieval – 42nd European Conference on IR Research*, ECIR '20, pages 508–516. Springer, 2020.

[F16] C. Hörtenhuemer and E. Zangerle. A Multi-Aspect Classification Ensemble Approach for Profiling Fake News Spreaders on Twitter—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[F17] M. Liu, E. Zangerle, X. Hu, A. Melchiorre, and M. Schedl. Pandemics, Music, and Collective Sentiment: Evidence from the Outbreak of COVID-19. In *Proceedings of the 21st International Society for Music Information Retrieval Conference 2020*, ISMIR '20, pages 157–165. ISMIR, 2020.

[F18] M. Vötter, M. Mayerl, G. Specht, and E. Zangerle. Recognizing Song Mood and Theme: Leveraging Ensembles of Tag Groups. In *Working Notes Proceedings of the MediaEval 2020 Workshop*, volume 3181 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[F19] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, and B. Stein. Overview of the Style Change Detection Task at PAN 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, Sept. 2020.

[F20] C. Bauer and E. Zangerle. Leveraging Multi-Method Evaluation for Multi-Stakeholder Settings. In *Proceedings of the 1st Workshop on the Impact of Recommender Systems co-located with 13th ACM Conference on Recommender Systems (ACM RecSys 2019)*, volume 2462 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[F21] W. Daelemans, M. Kestemont, E. Manjavacas, M. Potthast, F. M. R. Pardo, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Tschuggnall, M. Wiegmann, and E. Zangerle. Overview of PAN 2019: Bots and Gender Profiling, Celebrity Profiling, Cross-Domain Authorship Attribution and Style Change Detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 10th International Conference of the CLEF Association*, volume 11696 of *LNCS*, pages 402–416. Springer, 2019. DOI: 10.1007/978-3-030-28577-7_30.

[F22] H.-T. Hung, Y.-H. Chen, M. Mayerl, M. Vötter, E. Zangerle, and Y.-H. Yang. MediaEval 2019 Emotion and Theme Recognition task: A VQ-VAE Based Approach. In *Working Notes Proceedings of the MediaEval 2019 Workshop*, volume 2670 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[F23] M. Mayerl, M. Vötter, H.-T. Hung, B. Chen, Y.-H. Yang, and E. Zangerle. Recognizing Song Mood and Theme Using Convolutional Recurrent Neural Networks. In *Working Notes Proceedings of the MediaEval 2019 Workshop*, volume 2670 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[F24] M. Mayerl, M. Vötter, E. Zangerle, and G. Specht. Language Models for Next-Track Music Recommendation. In *Proceedings of the 31. GI-Workshop Grundlagen von Datenbanken*, volume 2367 of *CEUR Workshop Proceedings*, pages 15–19. CEUR-WS.org, 2019.

[F25] G. Rampl, E. Gruber, G. Hiebel, B. Larl, C. Posch, and E. Zangerle. "Nobody Climbs Mountains for Scientific Reasons" – Semantic Mountaineering History: Names and Activities in Mountaineering Discourse. In *10th International Corpus Linguistics Conference*, 2019.

[F26] A. Saeed, S. Ilic, and E. Zangerle. Creative GANs for generating poems, lyrics, and metaphors. *NeurIPS Workshop Machine Learning for Creativity and Design*, 2019. DOI: abs/1909.09534.

[F27] M. Schmidt and E. Zangerle. Article Quality Classification on Wikipedia: Introducing Document Embeddings and Content Features. In *Proceedings of the 15th International Symposium on Open Collaboration*, OpenSym '19, 13:1–13:8. ACM, 2019. DOI: 10.1145/3306446.3340831.

[F28] M. Vötter, E. Zangerle, M. Mayerl, and G. Specht. Autoencoders for Next-Track Recommendations. In *Proceedings of the 31. GI-Workshop Grundlagen von Datenbanken*, volume 2367 of *CEUR Workshop Proceedings*, pages 20–25. CEUR-WS.org, 2019.

[F29] E. Zangerle, R. Huber, M. Vötter, and Y.-H. Yang. Hit Song Prediction: Leveraging Low- and High-Level Audio Features. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ISMIR '19, pages 319–326. ISMIR, 2019.

[F30] E. Zangerle, M. Tschuggnall, G. Specht, B. Stein, and M. Potthast. Overview of the Style Change Detection Task at PAN 2019. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[F31] C. Bauer and E. Zangerle. Information Imbalance and Responsibility in Recommender Systems. In *Workshop Proceeding of the 2nd Workshop on Green (Responsible, Ethical and Social) IT and IS—the Corporate Perspective (GRES-IT/IS)*. Department für Informationsverarbeitung und Prozessmanagement, WU Vienna University of Economics and Business, 2018. URL: https://epub.wu.ac.at/7681/.

[F32] R. Binna, E. Zangerle, M. Pichl, G. Specht, and V. Leis. HOT: A Height Optimized Trie Index for Main-Memory Database Systems. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 521–534. ACM, 2018. DOI: 10.1145/3183713.3196896.

[F33] C. Esswein, M. Schedl, and E. Zangerle. geMsearch: Personalized Explorative Music Search. In *Joint Proceedings of the ACM IUI 2018 Workshops co-located with the 23rd ACM Conference on Intelligent User Interfaces*, volume 2068 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[F34] B. Larl and E. Zangerle. Leiwand Oida: Geolocating Regional Linguistic Variation of German on Twitter. In *Proceedings of the 6th Conference on Computer-Mediated Communication (CMC) and Social Media Corpora*, 2018.

[F35] M. Pichl, B. Pichl, and E. Zangerle. Carl: Sports Award Recommender. In *The SIGIR 2018 Workshop On eCommerce co-located with the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, volume 2319 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[F36] A. Poddar, E. Zangerle, and Y.-H. Yang. #Nowplaying-rs: a new benchmark dataset for building context-aware music recommender systems. In *Proceedings of the 15th Sound and Music Computing Conference*. SMC, 2018. DOI: 10.5281/zenodo.1422565.

[F37] E. Zangerle and C. Müller-Birn. Recommendation-Assisted Data Curation for Wikidata. In *Wiki Workshop 2018, co-located with The Web Conference*. 2018. DOI: 10.5281/zenodo.1194790.

[F38] E. Zangerle, M. Pichl, and M. Schedl. Culture-Aware Music Recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, pages 357–358. ACM, 2018. DOI: 10.1145/3209219.3209258.

[F39] E. Zangerle, M. Tschuggnall, S. Wurzinger, and G. Specht. ALF-200k: Towards Extensive Multimodal Analyses of Music Tracks and Playlists. In *Advances in Information Retrieval - 39th European Conference on IR Research*, ECIR '18, pages 584–590. Springer, 2018. DOI: 10.1007/978-3-319-76941-7_48.

[F40] B. Laner, A. Stauder, E. Zangerle, and T. Hug. Visualization Strategies for Digital Archives. The Case of the Ernst-von-Glasersfeld-Archive. In *4th Digital Humanities Conference Austria (DHA17)*, 2017.

[F41] B. Larl and E. Zangerle. Geolocating German on Twitter - Hitches and Glitches of Building and Exploring a Twitter Corpus. In *9th International Corpus Linguistics Conference 2017*, 2017.

[F42] 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*, volume 1984 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[F43] B. Murauer, E. Zangerle, and G. Specht. A Peer-Based Approach on Analyzing Hacked Twitter Accounts. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, HICSS '17, pages 1841–1850. IEEE, 2017.

[F44] 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*, 8(4):44–59, 2017. DOI: 10.4018/IJMDEM.2017100103.

[F45] M. Pichl, E. Zangerle, G. Specht, and M. Schedl. Mining Culture-Specific Music Listening Behavior from Social Media Data. In *2017 IEEE International Symposium on Multimedia*, ISM '17, pages 208–215. IEEE, 2017. DOI: 10.1109/ISM.2017.35.

[F46] E. Zangerle, M. Tschuggnall, and G. Specht. Analyzing Coherent Characteristics in Music Playlists. In *4th Digital Humanities Conference Austria (DHA17)*, 2017.

[F47] B. Larl and E. Zangerle. Geolocating German on Twitter Hitches and Glitches of Building and Exploring a Twitter Corpus. In *4th Conference on CMC and Social Media Corpora for the Humanities*, 2016.

[F48] M. Pichl, E. Zangerle, and G. Specht. Understanding Playlist Creation on Music Streaming Platforms. In *IEEE International Symposium on Multimedia*, ISM '16, pages 475–480. IEEE, 2016. DOI: 10.1109/ISM.2016.0107.

[F49] E. Zangerle, W. Gassler, 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 '16. ACM, 2016. DOI: 10.1145/2957792.2957804.

[F50] E. Zangerle, M. Illecker, and G. Specht. SentiStorm: Realtime Sentiment Detection von Tweets. *HMD Praxis der Wirtschaftsinformatik*, 53(4):514–529, 2016. DOI: 10.1365/s40702-016-0237-6.

[F51] 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 '16. ISMIR, 2016.

[F52] E. Zangerle, G. Schmidhammer, and G. Specht. Analysing the Usage of Wikipedia on Twitter: Understanding Inter-Language Links. In *49th Hawaii International Conference on System Sciences*, HICSS '16, pages 1920–1929. IEEE, 2016. DOI: 10.1109/HICSS.2016.243.

[F53] M. Pichl, E. Zangerle, and G. Specht. #Nowplaying on #spotify: leveraging spotify information on twitter for artist recommendations. In *Current Trends in Web Engineering, 15th International Conference, ICWE 2015 Workshops (Revised Selected Papers)*, pages 163–174. Springer, 2015. DOI: 10.1007/978-3-319-24800-4_14.

[F54] M. Pichl, E. Zangerle, and G. Specht. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name? In *15th IEEE International Conference on Data Mining Workshops*, ICDM '15, pages 1360–1365. IEEE, 2015. DOI: 10.1109/ICDMW.2015.145.

[F55] E. Zangerle, G. Schmidhammer, and G. Specht. #Wikipedia on twitter: analyzing tweets about wikipedia. In *Proceedings of the 11th International Symposium on Open Collaboration*, OpenSym '15, 14:1–14:8. ACM, 2015. DOI: 10.1145/2788993.2789845.

[F56] W. Gassler, E. Zangerle, and G. Specht. Guided Curation of Semistructured Data in Collaboratively-built Knowledge Bases. *Journal on Future Generation Computer Systems*, 31:111–119, 2014. DOI: 10.1016/j.future.2013.05.008.

[F57] 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)*, volume 1313 of *CEUR Workshop Proceedings*, pages 35–40. CEUR-WS.org, 2014.

[F58] 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 '14, pages 21–26. ACM, 2014. DOI: 10.1145/2661714.2661719.

[F59] E. Zangerle and G. Specht. “Sorry, I was hacked": A Classification of Compromised Twitter Accounts. In *Proceedings of the 29th ACM Symposium on Applied Computing*, SAC '14, pages 587–593. ACM, 2014. DOI: 10.1145/2554850.2554894.

[F60] E. Zangerle and G. Specht. Cybercrime on Twitter: Shifting the User Back into Focus. In *Proceedings of the WebScience Cybercrime / Cyberwar Workshop, co-located with WebSci14*, 2014.

[F61] E. Zangerle, W. Gassler, and G. Specht. On the impact of text similarity functions on hashtag recommendations in microblogging environments. *Social Network Analysis and Mining*, 3(4):889–898, 2013. DOI: 10.1007/s13278-013-0108-x.

Part II.

Selected Papers

4. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach

Publication

M. Pichl, E. Zangerle, and G. Specht. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In *Proceedings of the 2017 ACM International Conference on Multimedia Retrieval*, ICMR '17, pages 201–208. ACM, 2017. DOI: 10.1145/3078971.3078980

Abstract

Over the last years, music consumption has changed fundamentally: people switch from private, mostly limited music collections to huge public music collections provided by music streaming platforms. Thus, the amount of available music has increased dramatically and music streaming platforms heavily rely on recommender systems to assist users in discovering music they like. Incorporating the context of users has been shown to improve the quality of recommendations. Previous approaches based on pre-filtering suffered from a split dataset. In this work, we present a context-aware recommender system based on factorization machines that extracts information about the user's context from the names of the user's playlists. Based on a dataset comprising 15,000 users and 1.8 million tracks we show that our proposed approach outperforms the pre-filtering approach substantially in terms of accuracy of the computed recommendations.

4.1. Introduction

Recently, we are facing a fundamental change in the way people consume music: more and more people switch from private, mostly limited music collections to public music streaming collections containing several millions of tracks [276]. People increasingly do not store music locally on CDs and hard drives anymore. Instead, they access millions of tracks offered by cloud-based streaming services using various devices. To increase usability, streaming platforms heavily rely on recommender systems to help users in discovering music they like. Previous research has shown that the context of a user (i.e., occasion, event or emotional state) plays an important role for providing personalized music recommendations [241, 275]. Kamalzadeh et al. [228] showed that people listen to different music during different activities and found that people organize tracks in their music collections by the intended use (i.e., working or exercising). This finding is backed up by Cunningham et al. [109], who found that people create playlists that are intended for certain activities.

Over the last years, data for quantitatively validating these studies became available: music streaming platforms provide means for "social playlist generation"—playlists that are shared among friends or to the public. Particularly public playlists serve as an essential new data source for music recommender systems. For Spotify[1], a popular music streaming service, all user-created playlists are public by default[2] and thus can be crawled using the Spotify API[3]. Pichl et al. [361] propose an approach for clustering contextually similar playlists by exploiting the names of these playlists. The clusters are then leveraged in a collaborative filtering recommender system (CF) with pre-filtering [11], hence CF is applied to each cluster individually. Thus, the recommender system is applied to different parts of the dataset in isolation, a method that has drawbacks: the user profiles are split up among the different clusters and thus, there is no holistic view on the user. In addition, recommendation accuracy substantially varies among clusters, as these are different in size.

In this work, we follow up and complement the research of Pichl et al. [361] by utilizing their proposed playlist aggregation pipeline to implement a novel recommender system to overcome the drawbacks of contextual pre-filtering. Particularly, we are interested in how contextual clusters may be leveraged for music recommendations while ensuring that the drawbacks of the pre-filtering approach can be avoided. Therefore, we propose to make use of Factorization Machines (FM) [371] that are directly able to incorporate the contextual clusters extracted from the names of playlists for the computation of recommendations.

[1] http://www.spotify.com
[2] https://developer.spotify.com/web-api/working-with-playlists/#public-private-and-collaborative-status
[3] http://developer.spotify.com/web-api/

In several empirical experiments using k-fold cross-validation we show that our proposed factorization machine-based recommender system outperforms context-agnostic recommender systems, pre-filtering context-aware recommender systems as well as classification-based context-aware recommender systems substantially in terms of recall, precision and the F_1-measure. Our experiments show that factorization machines are particularly capable of tackling the major issue of the pre-filtering approach (i.e., splitting up the dataset). To foster reproducibility and repeatability, we make both our code and data used publicly available by publishing our recommender system and the evaluation framework utilized in this paper on GitHub[4].

The remainder of this paper is structured as follows. In the next section, we focus on related work before presenting our recommendation approach in Section 4.3. After that, we introduce the reader to our conducted experiments aiming to benchmark different recommendation systems including our proposed recommender system. In the subsequent sections, we present the results of the experiments and discuss them in Section 4.5. Finally, we wrap up our work in Section 4.6.

4.2. Related Work

We classify related work into two main fields of research: context-aware music recommender systems and approaches concerned with leveraging new data sources for music recommendations.

It is widely agreed upon the fact that the user's context improves personalized recommendations [11]. This is why we can see a shift from purely content- or CF-based approaches towards more user-centric approaches incorporating the user's context [413]. In the field of music recommender systems, studies showed that users often seek for music that matches their current context (i.e., occasion, event or emotional state) [241, 275]. As for the different types of contexts, Kaminskas and Ricci [232] distinguish environment-related context (location, time, weather), user-related context (activity, demographic information, emotional state of the user) and multimedia context (text or pictures the user is currently reading or looking at). Examples for contextual information that is leveraged for music recommendations are emotion and mood [33, 63, 178, 381], the user's location [27, 96, 233, 234] or recommending music fitting to documents on the web a user reads at the moment [71]. As for the integration of contextual information into a recommender system, Adomavicius et al. [11] classify approaches modeling the user's context into contextual pre-filtering, contextual post-filtering and contextual modeling approaches. We consider the approach presented in this work as a contextual modeling approach as we do not filter the input or output data of the system.

[4] https://github.com/dbis-uibk/MusicRecommenderEvaluator/

As for music recommender systems based on novel publicly available data, Zangerle et al. [507] propose a music recommender system based on association rules computed based on user listening behavior extracted from #nowplaying tweets (tweets in which users state which musical track they are listening to at the moment). Moreover, context-aware approaches for music recommendations that are based on information extracted from public data sources have been proposed. Schedl and Schnitzer exploit #nowplaying tweets enhanced with acoustic features extracted from 7digital[5] and extract context information about these tracks by utilizing a web search on the track and artist [420]. In [422], Schedl et al. explore the use of geospatial information for a set of collaborative filtering approaches. Furthermore, also LastFM has been utilized for analyzing the listening behavior of users [182, 408]. Pichl et al. [361] extract contextual information from the names of playlists of Spotify users and incorporate these in the process of recommending tracks. The work presented in this paper builds upon this approach and aims to address the problems of the pre-filtering approach (as proposed by Pichl et al.) by using factorization machines. To the best of our knowledge, this is the first factorization machine-based recommendation approach for integrating contextual clusters derived from playlist names into a music recommender system.

4.3. Methods

In this section, we present our proposed recommendation algorithm. First, we introduce the approach taken for computing clusters of contextually similar tracks. In a next step, we present the proposed recommendation framework, which leverages the information provided by these contextual clusters. Figure 4.1 depicts the overall workflow for the computation of music recommendations utilizing contextual clusters.

As the approach taken for computing contextual clusters relies on the work of Pichl et al. [361], we naturally utilize the same dataset for evaluating our approach (and comparing it to the original approach). This dataset contains 143,528 unique playlists created by 15,345 unique users who listened to 1,878,457 tracks in the form of <user, track, artist, playlist>-quadruples.

4.3.1. Playlist Aggregation and Cluster Generation

In a first step, we compute clusters of contextually similar playlists based on the context information extracted from the names of playlists. Therefore, we follow the method introduced by Pichl et al. [361], which we will shortly sketch in the following. As depicted in Figure 4.1, we firstly stem all playlist names and lemmatize the tokens in a first step. In a next step, we remove non-contextual terms such as genre, artist and track names as well as general stop words, as these do not contain any contextual information. We use the resulting bags of lemmata describing each playlist to compute the term frequency-

[5] http://www.7digital.com

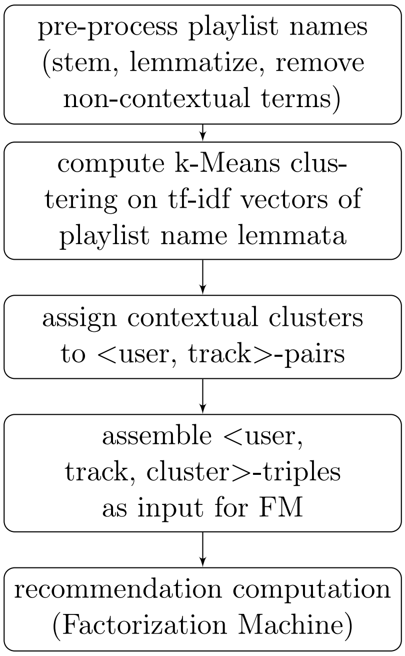

pre-process playlist names (stem, lemmatize, remove non-contextual terms)
compute k-Means clustering on tf-idf vectors of playlist name lemmata
assign contextual clusters to <user, track>-pairs
assemble <user, track, cluster>-triples as input for FM
recommendation computation (Factorization Machine)

Figure 4.1.: Pipeline for Computing Recommendations.

inverse document frequency (tf-idf) [440] for each bag of lemmata representing a playlist name. Using tf-idf, we represent each playlist as a vector containing the tf-idf weights. This allows us to compute playlist similarities by computing the pairwise cosine similarity of the playlist vectors. Using the computed similarities, we span a distance matrix and finally find contextually similar playlists by applying k-Means to the playlists in the matrix. As we evaluate our approach using the same dataset as Pichl et al. [361], we set the number of clusters to $k = 23$, as proposed in the original approach. In the next step, we integrate the contextual clusters in the recommendation computation as presented in the following section.

4.3.2. Recommendation Computation

Our proposed recommendation approach aims to provide track recommendations for a given user in a given context. Particularly, we aim to model users by the tracks they listened to and enrich this information with the contexts in which each individual user has listened to those tracks. For the given input dataset, we assume that by adding a track to a playlist, the user expresses some preference for the track. For means of simplicity, we will describe a user-track interaction extracted from a playlist as "a given user listened to a given track". Furthermore, we infer from previous findings [109, 228], that user create playlists to listen to the contained tracks in the context specified by the playlist name.

The initial input dataset contains $<user, track, playlist>$-triples. We transform this dataset into a set of $<user, track, context\_cluster>$-triples by applying the clustering method presented in Section 4.3.1 and assigning each user-track pair with one of the 23 contextual clusters in which the given user has listened to the given track. By adding a fourth factor *rating* to the dataset, we transform the recommendation computation task into a rating prediction task: for each unique $<user, track, context\_cluster>$-triple, the *rating* r_{ijk} is 1 if the user u_i has listened to the track t_j in cluster c_k. Our dataset does not contain any implicit feedback by users (i.e., play counts, skipping behavior, session durations or dwell times during browsing the catalog). Therefore, we cannot estimate any preferences towards an item a user not listened to as proposed by [199]. Thus, for each $<user, track, context\_cluster>$ combination for which we cannot obtain a rating for, we assume the rating to be $r = -1$ (as proposed by [199]). The rating r_{ijk} for each user u_i, track t_j and cluster c_k can now defined as stated in Equation 4.1. Although there is a certain bias towards negative values as some missing values might be positive, Pan et al. [341] found that this method for rating estimation works well.

$$r_{ijk} = \begin{cases} 1 & if\ u_i\ listened\ to\ t_j\ in\ c_k \\ -1 & otherwise \end{cases} \tag{4.1}$$

To get a better understanding of the resulting dataset, we depict a sample of the dataset in Table 4.1. Based on this dataset, we train a classifier that decides whether a user has listened to a track in a contextual cluster or not. For this computation, we require a given user, track and cluster as input.

As for the actual computation of recommendations, we opt for factorization machines (FM) [371, 373], as these can be considered as state-of-the-art recommendation approach and have been shown to perform well for recommender systems [373]. 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 dimension). As we aim to exploit the interaction effects of users, tracks and clusters with this recommender system, we chose to utilize a FM of the order $d = 2$ modeling all single and pairwise interactions between input variables as depicted in Equation 4.2.

$$\hat{r}_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j \tag{4.2}$$

Equation 4.2 shows that a FM computes rating predictions by modeling a global bias (w_0), the influence of the user, track as well as the clusters ($\sum_{i=1}^{m} w_i x_i$) along with the quadratic interaction effects of those ($\sum_{i=1}^{m} \sum_{j=i+1}^{m} \langle \vec{v_i}, \vec{v_j} \rangle$). However, instead of learning all weights $w_{i,j}$ for the interaction effects, a FM relies factorization to model the interaction as the inner product of low dimensional vectors ($\langle \vec{v_i}, \vec{v_j} \rangle$) [373].

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

4.4. Experiments

In this section, we introduce the experiments conducted to evaluate the proposed approach and the baseline approaches aiming at answering our research questions. We start with a description of the dataset used for the evaluation before focusing on the experimental setup and the evaluation measures.

4.4.1. Dataset

For our experiments, we apply the proposed clustering method on the initial dataset and reshape the input dataset into a set containing <user, track, context_cluster, rating>-quadruples. We assign each track in a playlist with a rating value as described in Section 4.3.2. The rating indicates whether a certain user listened to a certain track in a certain cluster ($r = 1$) or not ($r = -1$). A fragment of the dataset is shown in Table 4.1. This excerpt shows that user 872 has listened to track 250246 in contextual cluster 0, whereas user 911 has listened to track 250246 in context 2. This dataset forms the foundation for our experiments, which are presented in the next section.

User	Track	Contextual Cluster	r
872	309275	0	1
872	309275	1	-1
911	250246	0	-1
911	250246	0	-1
911	250246	2	1

Table 4.1.: Dataset Fragment.

4.4.2. Baseline Recommender Systems

We compare our proposed FM approach to three baseline recommender systems: a CF-based system, a SVD-based system and a classification-based system. To incorporate context information in the CF- and SVD-based baseline approaches, we apply pre-filtering [11], where the computation of recommendations (CF or SVD) is performed on each contextual cluster individually. I.e., we compute the recommendations on a subdataset of the dataset restricted to a certain cluster. The classification-based system uses the computed contextual clusters as an input feature to the classifier. With those systems, we benchmark classical CF, approaches facilitating latent features (considered as state-of-the art in recent years) and a classification-based approach against our proposed factorization machine-based recommender system.

The first recommender system to benchmark is a collaborative filtering approach [7]. The idea behind CF is to recommend items the k-nearest neighbors of a user interacted with. For determining the nearest neighbors, we compute pairwise user similarities by computing the Jaccard Coefficient [204] 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 as depicted in Equation 4.3, where we denote S_i as the set of tracks a user i has listened to.

$$Jaccard_{i,j} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{4.3}$$

The second baseline recommender system is based on singular value decomposition (SVD) [262]. SVD predicts ratings by extracting a number of latent features from the user-item matrix R. In our setting, this is a sparse matrix containing all the binary ratings r_{ij} (cf. Equation 4.1) of all users u_i and the tracks t_j they listened to. These latent features, characterizing types of tracks, are computed by factoring the user-item matrix R into two matrices U and V, which represent the user and item factors. Hence, R is the cross product of U and V ($R = UV$). We approximate U and V by minimizing the error to the known ratings r_{ij} using stochastic gradient descent optimization (SGD) [262].

Thirdly, we aim to compare our proposed approach with a classification-based recommendation approach as the performed recommendation computation can also be considered as a one-class classification problem [341]. Therefore, we implement a random forest classifier [287] as it has two main advantages: firstly, we only have to tune one parameter: the number of trees [339]. Secondly, all trees can be computed in parallel and the algorithm scales linearly with the number of trees.

Furthermore, we compare all recommender system to a random-choice baseline. The assumption behind this baseline is that the fundamental chances of guessing whether a track was listened by a user ($r = 1$) or not ($r = -1$) is 50%. Thus, the random baseline for the *precision* measure is 0.5. The same holds for RMSE and MAPE, where the random baseline is also 0.5. For the *recall* measure we cannot state a single baseline value, as recall is dependent on the number of recommendations n as explained in Section 4.4.4 and shown in Equation 4.8.

A detailed description of the evaluation is given in the next section.

4.4.3. Experimental Setup

To evaluate the performance of the different recommender systems, we conduct a 5-fold cross-validation. 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 for the folds, each fold contains an arbitrary number of relevant and irrelevant items. The relevant items are tracks a user has listened to within a certain cluster, whereas the latter are items a user did not listen to at all within a 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 predicted ratings $\hat{r}$ as well as the actual ratings r in the test set, we compute the evaluation measures as described in Section 4.4.4. These evaluation measures are computed for each fold separately and before computing the measures, we perform a min-max scaling. For the results in Section 4.5, we compute the average across all folds.

For evaluating the top-n recommendations performance, we sort the result 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 Rendl et al. [371]. Generally, we tuned 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 1,000 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 CF, $n = 30$ and for SVD, $k = 50$ are suitable parameter options.

4.4.4. Evaluation Measures

In this section, we elaborate on the evaluation measures used for assessing the performance of the different recommendation algorithms.

For assessing the rating prediction task, we compute the different widely used error measures: root mean square error (RMSE) as well as the mean absolute percentage error (MAPE) as stated in Equations 4.4 and 4.5, where $\hat{r}$ is the predicted rating and r the actual rating as contained in the test set. For the results stated Table 4.2, we compute the average error among all ratings r_i in the test set. Please note that for computing the error measures, we scaled the predicted rating $\hat{r}$ between 0 and 1 using min-max scaling to be able to directly compare the evaluated approaches.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{n}} \tag{4.4}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{r_i - \hat{r}_i}{r_i}\right| \tag{4.5}$$

For measuring the performance of the top-n recommendations, we rely on *recall*, *precision* and the F_1-measure. For computing the *recall*-measure, we have to classify the tracks in the test set into relevant and non-relevant items. We consider an item as relevant, if the user has listened to this track in a certain cluster and thus $r = 1$. An item is considered as non-relevant for a given user if the user did not listen to it in a given cluster and thus, $r = -1$. For a certain user, a track can be relevant in certain clusters and simultaneously not relevant in other cluster. In case of the FM-based recommender, we have to transform the rating prediction task into a one-class classification task [341] on whether a given track relevant or not relevant for a given user in a given context to be able to compute the top-n measures. Therefore, we consider $\hat{r}$ as 1 if the computed probability that a user interacted with an item $P(r = 1)$ is higher than 50% as stated in Equation 4.6. As for the ranking, we rely on the predicted rating for ranking the recommendations to be able evaluate the top-n recommendations.

$$\hat{r} = \begin{cases} 1 & if\ P(r=1) \geq 0.5 \\ -1 & otherwise \end{cases} \tag{4.6}$$

In Equations 4.7 and 4.8 we state how precision (P) and recall (R) are computed. Precision measures the number of true positives (TP) in relation to the number of recommendations n, which is the number of true positives plus the number of false positives (FP). We consider all items where $r = \hat{r} = 1$ as true positives. In contrast, Recall measures the ratio of true positives and the number of relevant items in the test set (RIT). These relevant items are the items a user has listened to in the given context and hence, have the rating $r = 1$. This recall computation implies that there is natural a cap of the recall determined by the number of recommendations n. The maximum recall is $\frac{n}{RIT}$. Hence, a low number of recommendations n naturally implies a low recall R.

$$P = \frac{TP}{TP + FP} \tag{4.7}$$

$$R = \frac{TP}{RIT} \tag{4.8}$$

For assessing the overall *precision*, *recall* and F_1-measure of the evaluated recommender systems, we compute the measures for each individual fold and compute the average among all users in a final step. We elaborate on the results of the presented evaluation in the following section.

4.5. Results and Discussion

Based on the evaluation setup and measures described in the preceding section, we assess the performance of the following recommender systems: a pure CF-based recommender system (CF), context-aware CF with pre-filtering (PR-CF) as proposed by Pichl et al. [361], a SVD-based recommender system (SVD), a context-aware SVD-based recommender system with pre-filtering (PR-SVD), a context-aware random forest classifier-based recommender system (RF) as well as our proposed context-aware FM-based recommender system (FM). As outlined in Section 4.4.2, we consider the first five recommender systems as baseline approaches to our FM-based recommender. Additionally, we compare all recommender system against the random baseline (RB).

As described in Sections 4.4.3 and 4.4.4, we evaluate the rating prediction task and the top-n recommendations. Analogously to the previous section, we start with discussing the rating prediction before analyzing the top-n recommendations task.

Recommender	RMSE	MAPE
CF	0.921	0.424
Pre-filtering CF	0.914	0.418
SVD	0.913	0.417
Pre-filtering SVD	0.914	0.418
RF	0.520	0.209
FM	0.560	0.282

Table 4.2.: Evaluation of the Rating Prediction Task (all Tracks).

The results of the rating prediction task applied to all items in the test set are stated in Table 4.2. We find that with respect to the rating prediction task, the presented classifier-based context-aware approaches (RF and FM) clearly outperform all other approaches. RF and FM reach a RMSE of 0.520 and 0.560 and a MAPE of 0.209 and 0.282, respectively. The proposed baseline approaches reach RMSE values of > 0.9 and MAPE values of > 0.4. However, we also observe that none of the algorithms outperforms the random baseline of 0.5 w.r.t. RMSE (in contrast to MAPE). We lead this back to the fact that as RMSE naturally is more sensitive to high deviations between r and $\hat{r}$. Furthermore, the high error rate can also be explained by the fact that there are far more tracks a user did not listen to in a given cluster than tracks a user did actually listen to in a given cluster (i.e., the underlying matrix is highly sparse). Therefore, computing the error measures incorporating all tracks in the data leads to results biased towards imprecise rating predictions of low ranked (and hence, irrelevant) items. As the majority of tracks within our dataset are not relevant for a given user in a given context, evaluating RMSE and MAPE of all tracks within the dataset naturally includes tracks are not relevant for a user. Our recommender systems considers all tracks with a predicted rating $\hat{r} < 0.5$ as irrelevant to the user and these tracks are naturally not shown in the list of recommendations. We argue that the error for tracks with ratings $\hat{r} < 0.5$ are irrelevant for ranking the tracks on the recommendation list. To illustrate this bias we repeat the experiment for all tracks the recommendation algorithms consider as relevant for the user (i.e., tracks with a predicted rating of $\hat{r} \geq 0.5$ after the min-max scaling). The results of this evaluation are depicted in Table 4.3.

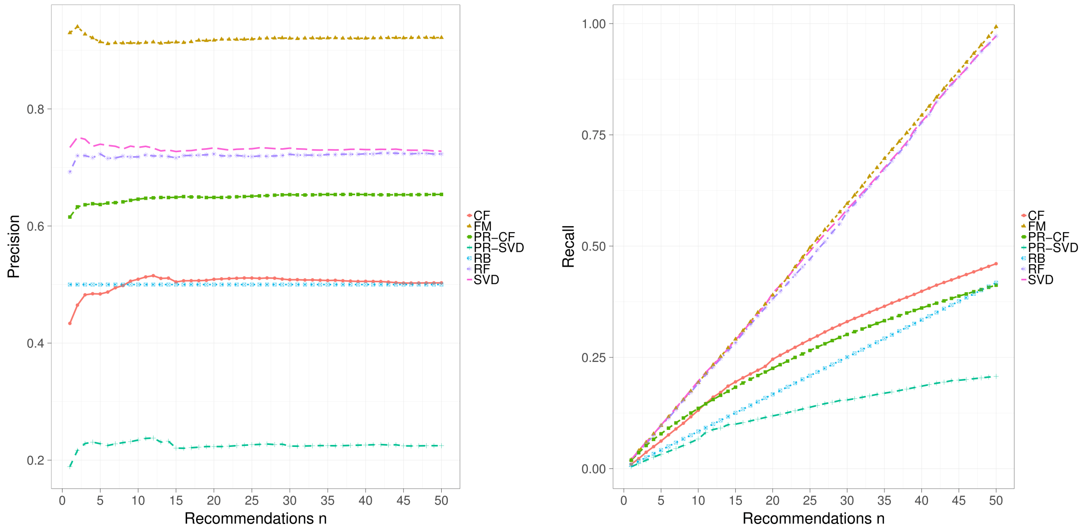
Recommender	RMSE	MAPE
CF	0.389	0.151
Pre-filtering CF	0.143	0.021
SVD	0.366	0.177
Pre-filtering SVD	0.939	0.927
RF	0.415	0.172
FM	0.380	0.221

Table 4.3.: Evaluation of the Rating Prediction Task (relevant Tracks).

When considering only tracks with a predicted rating of $\hat{r} \geq 0.5$, the results show that all algorithms except pre-filtering SVD outperform the baseline. The SVD-based recommender system even performs better than the RF-based one and slightly outperforms our proposed FM-based recommender. Furthermore, in this scenario, applying pre-filtering to CF improves results.

However, we argue that for the use cases we discuss later in this section, the top-n-recommendations evaluation is of higher importance as a user-centric evaluation that measures the utility of the top-n recommendations provided to the user is vital and of higher importance than actual error rates. Particularly, we argue that a top-n performance for low n is vital for users. Hence, we are particularly interested in the performance of the proposed recommendation approaches for lower n. Hence, not the precise rating prediction is crucial but ranking the track, such that the most relevant tracks for a user in a given context are listed within the top-n recommendations. This is, as the recommender system computes the list by sorting all potential track recommendations descending by the predicted rating $\hat{r}$ and returns the top-n tracks based in this list. Amongst others, in the remainder of the this section we empirically show the discrepancy between rating prediction accuracy and top-n prediction accuracy: although the RMSE and MAPE of CF is low, even lower than using RF, the performance evaluated measuring the accuracy of the top-n recommendations hardly outperforms the baseline.

For the presenting the results of the top-n performance evaluation of the proposed recommendations task, we depict the *precision*- and *recall*-curves in the Figures 4.2a and 4.2b for $n = \{1 \ldots 50\}$. Aiming at making the performance of the recommender systems easily comparable, we integrated both, the *precision*- and the *recall* into the F$_1$ measure and plot the F_1 measure in Figure 4.3. Figure 4.2b shows that the FM, RF and SVD-based approaches perform substantially better in terms of recall than the other baselines across all number of recommendations n. Notably, the pre-filtering SVD approach performs worse than the random baseline across all n. As for precision (shown in Figure 4.2a) we detect a similar behavior. Again, pre-filtering SVD reaches substantially lower precision values than the other approaches. Interestingly, the SVD approach performs better than the pre-filtering SVD approach and reaches values similar to the random baseline. The FM-based approach performs substantially better than SVD and RF, followed by pre-filtering CF.



(a) Precision@n.

(b) Recall@n.

Figure 4.2.: Evaluation: Recall and Precision for $n = \{1 \ldots 50\}$

When examining the F_1 results in Figure 4.3, we consequently observe that all approaches outperform the baseline approach for $n < 25$. For $n >= 25$, only pre-filtering SVD reaches F_1 values lower than the random baseline. Considering the *precision* and *recall* plots of the algorithms in Figures 4.2a and 4.2b respectively, we observe that pre-filtering SVD performs poorly independent of the evaluation measure. However, we also note that a recommender system based on latent features computed via SVD provides accurate results and reaches high *recall* values. From this, we derive that the implicitly computed latent features represent track-context associations. Hence, pre-filtering limits the amount of input data available for computing latent features. Hence, we argue that pre-filtering SVD is not an effective approach for our recommendation task.

Moreover, we observe that all approaches besides pre-filtering SVD outperform classical CF. However, we have to note that CF hardly beats the random baseline, for which we assume that the chances to guess whether a track was listened by a user ($r = 1$) or not ($r = -1$) is 50%. We lead this back to a lack of non-boolean ratings as explicit ratings would allow a more precise computation of the user similarity and hence, more precise recommendations. We argue that this would improve the ordering of the tracks, which is especially crucial for the top-n recommendation task.

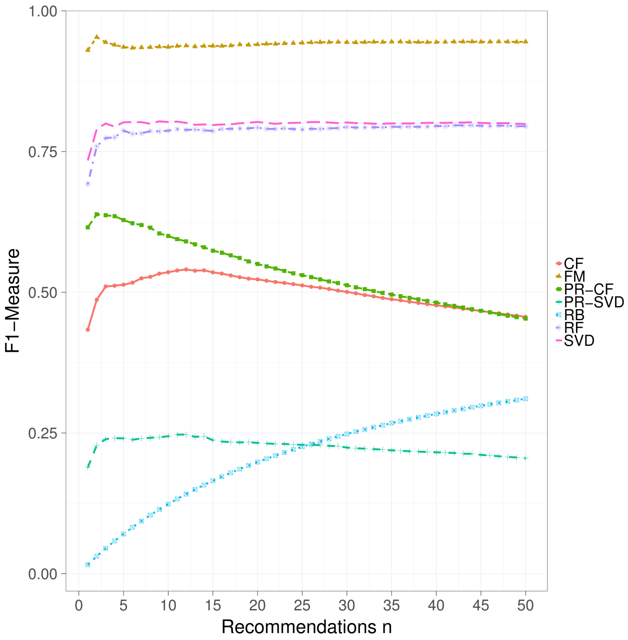


Figure 4.3.: F1@n.

Additionally our experiments show that contextual pre-filtering is beneficial for CF. Prefiltering CF beats the random baseline by a 46% higher F_1-score, which confirms the results of Pichl et al. [361]. However, as we observe in Figures 4.2a and 4.2b, prefiltering is only highly beneficial for *precision*. The obtained *recall* value is slightly lower for the pre-filtering CF approach than for standard CF approach (-3,08%). 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 limits the *recall*. Secondly, as we compute user similarities on a restricted amount of data, not all similarities are captured which also possibly limits the set of possible recommendations.

Finally, we note that our proposed FM-based recommender system clearly outperforms all other approaches including SVD and RF in terms of *precision*, whereas the *recall* behaves similar for the three best approaches (FM, SVD and RF). We lead this behavior back to the way recall is computed. 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 ($\frac{n}{RIT}$). As we sort recommendations by the predicted rating $\hat{r}$ 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 contained in the top-n recommendations. This ultimately results in a higher number of RIT, as we compare the top-n recommendations to the actual rating value r. This is why the top-algorithms approach a recall of $n/50$.

Bollen et al. [57] addressed the problem of choice overload and state that user satisfaction is highest when presenting the user with top-5 to top-20 items—naturally assuming that the recommendation list contains a sufficient number of relevant items for the user. This is why we state the results for a small number of recommendations n in Table 4.4. Please note that we only list the top-3 algorithms here (FM, SVD and RF).

Recommender	F_1@1	F_1@5	F_1@10	F_1@20	F_1@50
FM	0.93	0.94	0.94	0.94	0.95
SVD	0.73	0.80	0.80	0.80	0.80
RF	0.69	0.79	0.79	0.79	0.80

Table 4.4.: F_1-Measure for different n.

The results in Table 4.4 show that for maximizing the user satisfaction according to Bollen et al. [57], our proposed FM-based approach clearly outperforms RF-based approaches (where we model the context explicitly) as well as SVD-based approaches (where we model the context-track associations implicitly via latent features). The FM model in Equation 4.2 depicts that the FM models the context explicitly as part of the variable's main effects: $\sum_{i=1}^{n} w_i x_i$ and additionally similar to the SVD approach implicitly in the pair-wise interactions: $\sum_{i=1}^{m} \sum_{j=i+1}^{m} \vec{v_i}, \vec{v_j} x_i x_j$. Underpinned by an empirical evaluation we argue 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 context-aware music recommendation in a setting similar to the one presented in this work.

Summing up, in this work we show how contextual clusters can be leveraged for context-aware music recommendations. We find that contextual clusters can be leveraged for music recommendations without the drawbacks of the pre-filtering approach either by using a classifier approach or by incorporating latent features. Particularly, we find that by using Factorization Machines, the best results regarding the accuracy of recommendations can be obtained. Possible use cases for such recommender systems are (i) the generation of track suggestions during the playlist generation phase of a user and (ii) "contextual browsing" which helps users discovering music they like. For the first use case, the recommender system can recommend tracks that are likely to be interesting to the user that can be added to the currently curated playlist. Thus, the recommender system presents tracks to the user, which similar users added to contextually similar playlists. The second use case, the "contextual browsing", is based on the finding of Cunningham et al. [109] that people browse music collections to discover tracks they like to listen to during different activities or situations. After a user selects a certain context (or the context is automatically inferred), our recommender system can provide lists of interesting tracks for this specified context. This use case is similar to the classical top-n recommendation task we evaluated in Section 4.4.

4.6. Conclusion

In this work, we propose a novel approach for incorporating contextual clusters extracted from the names of user playlists for the computation of context-aware track recommendations. Particularly, we present a recommendation approach based on Factorization Machines. We evaluate the prediction accuracy of different recommendation approaches based on a dataset of 15,000 users. Our k-fold cross-validations show that contextual clusters can indeed contribute substantially to recommendation accuracy by relying on either a classifier-based approach or approaches facilitating latent features. Particularly, the obtained results show that our proposed factorization machined-based recommender system is able to outperform the baseline approaches substantially. We consider these findings highly promising. Hence, in future work, we aim to evaluate different FM-models and configurations. Particularly, we are also interested in the use of higher order factorization machines [52].

5. User Models for Multi-Context-Aware Music Recommendation

Publication

M. Pichl and E. Zangerle. User models for multi-context-aware music recommendation. *Multimedia Tools and Applications*, 80(15):22509–22531, 2021. DOI: 10.1007/s11042-020-09890-7

This manuscript is an extended version of the following conference paper:
M. Pichl and E. Zangerle. Latent Feature Combination for Multi-Context Music Recommendation. In *International Conference on Content-Based Multimedia Indexing*, CBMI '18, pages 1–6. IEEE, 2018. DOI: 10.1109/cbmi.2018.8516495

Abstract

In the last decade, music consumption has changed dramatically as humans have increasingly started to use music streaming platforms. While such platforms provide access to millions of songs, the sheer volume of choices available renders it hard for users to find songs they like. Consequently, the task of finding music the user likes is often mitigated by music recommender systems, which aim to provide recommendations that match the user's current context. Particularly in the field of music recommendation, adapting recommendations to the user's current context is critical as, throughout the day, users listen to different music in numerous different contexts and situations. Therefore, we propose a multi-context-aware user model and track recommender system that *jointly* exploit information about the current situation and musical preferences of users. Our proposed system clusters users based on their situational context features and similarly, clusters music tracks based on their content features. By conducting a series of offline experiments, we show that by relying on Factorization Machines for the computation of recommendations, the proposed multi-context-aware user model successfully leverages interaction effects between user listening histories, situational, and track content information, substantially outperforming a set of baseline recommender systems.

5.1. Introduction

Over the last decade, people have increasingly started to use music streaming platforms providing millions of tracks [276]. Streaming platforms heavily rely on recommender systems to help users navigate through the provided collections and discover music they like. However, the extent to which a user enjoys and likes a recommended song heavily depends on the user's current context. Previous research has shown that information about the context of a user (e.g., time, location, occasion, or emotional state) is vital for providing suitable personalized music recommendations [241, 275] as people listen to different music during different activities [228]. Also, Cunningham et al. [109] have shown that users create playlists that are specifically intended for certain contexts or activities.

Extracting contextual information for a music recommendation scenario, however, is a complex task. To this end, in previous work we proposed an approach for clustering contextually similar playlists by extracting contextual information from the names of playlists, ultimately allowing to find playlists that users created for similar purposes and situations [359, 361]. We proposed to leverage these *situational clusters* as an additional feature for a Factorization Machine-based recommender system. Furthermore, we performed an analysis of the acoustic features (e.g., tempo or danceability) of the tracks contained in individual playlists and found that there are five different groups, so-called *archetypes*, of playlists, described by their audio characteristics [362]. However, what is still missing, is linking information about the situational context of a user with acoustic feature-based playlist archetypes that represent different types of music that users listen to. In this work, we are particularly interested in how contextual and audio characteristics may *jointly* be leveraged for track recommendations[1]. Hence, we present a novel user model combining situational and acoustic context information and refer to this model as *multi-context user model*. We propose to make use of Factorization Machines (FM) [371] as these allow for exploiting latent features and interactions between input variables. This allows us to exploit interaction effects between contextual clusters extracted from the names of playlists and acoustic clusters based on audio characteristics. In several experiments, we show that a recommender system leveraging this proposed model substantially outperforms context-agnostic baselines and, more importantly, a context-aware recommender system that relies on either context- or acoustic feature-based clusters individually.

The main contribution of this work is threefold: firstly, we leverage two types of contextual information for the computation of a multi-context-aware user model that allows capturing a user's preference towards certain archetypes of music (acoustic context) as well as the contexts in which users listen to certain tracks (situational context). Secondly, by utilizing Factorization Machines, we exploit interaction effects between the input variables (user listening history, acoustic feature-based playlist archetypes, and situational

[1] Please note that this manuscript is an extended version of [356], which was presented at the 2018 International Conference on Content-Based Multimedia Indexing (CBMI2018).

context). FMs hence allow us to model and exploit the influence of a certain context on the choice of tracks for a given user. Thirdly, we also investigate higher-order Factorization Machines that aim to leverage higher-order interactions of the input of the Factorization Machine.

The remainder of this paper is structured as follows. In Section 2, we discuss related work. In Section 5.3, we formulate the problem underlying our work. Section 5.4 presents the dataset utilized and in Section 5.5, we present the proposed multi-context user model and recommendation approach. Subsequently, we describe the experimental setup underlying our evaluation in Section 5.6 and present and discuss the obtained results in Section 5.7. Finally, we wrap up our work in Section 5.8.

5.2. Related Work

Related literature can be categorized into recommendation approaches based on matrix-factorization, context-aware recommender systems, and Factorization Machines. In the following, we elaborate on these categories.

User-based collaborative filtering has been shown to work well in the field of music recommender systems [361, 422, 507]. User-based CF relies on the user-item matrix, which holds ratings of users for items (so-called interactions). This matrix is used to group users based on their rating behavior and hence, to find similar users. Based on such nearest neighbors, items for a given user are recommended by choosing the items these nearest neighbors rated favorably and that are new to the user, assuming that similar users will rate items similarly. CF-based approaches utilizing matrix factorization (MF) techniques have been shown to yield better recommendation accuracy than traditional neighborhood-based CF approaches (e.g., [260]). MF approaches are also known as latent factor models, as factorizing the user-item matrix yields a latent representation of user-item interactions on a more abstract level (e.g., by applying Singular Value Decomposition (SVD) [260]). Several extensions to MF have been shown to work well (e.g., for implicit feedback data [199, 372] or for context-aware recommendations [34, 258]).

However, many of the current collaborative filtering-based track recommendation or continuation approaches are not able to cope with so-called “out-of-set” tracks (i.e., tracks that do not appear in the training data) [469]. As a solution, hybrid systems combining collaborative filtering and content-based approaches have been proposed. Vall et al. [469] proposed to combine collaborative filtering and rich content descriptors for music tracks into a feature-combination hybrid in a playlist continuation scenario. Furthermore, McFee and Lanckriet [307] proposed to combine collaborative filtering and content information such as e.g., low-level acoustic features, lyrics, or social tags in a hypergraph, modeling users by random walks on this graph. More recently, van den Oord [470] proposed a Deep Learning-based model for this task, utilizing Convolutional Neural Networks to integrate matrix factorization and latent factors extracted from the audio signal of songs. Furthermore, hybrid systems in this regard have also been realized by traditional hybridization strategies where the results of a CF-based and a content-based recommender system are combined by weighting the results [210] or by re-ranking strategies [180].

Generally, context can be considered as any additional information improving recommendation accuracy and it is widely agreed upon the fact that the user's context improves personalized recommendations [10]. In the field of music recommender systems, users often seek music that suits their current context (i.e., occasion, event, or emotional states) [241, 275]. Kaminskas and Ricci [232] distinguish different kinds of contexts: environment-related context (location, time, weather), user-related context (activity, demographic information, emotional state of the user), and multimedia context (text or pictures the user is currently reading or looking at). Examples for contextual information that is leveraged for music recommendations are emotion and mood (e.g., [33, 178, 505]), the user's location (e.g., [96, 234]), or recommending music matching documents on the web a user reads at the moment [71]. Adomavicius and Tuzhilin [10] classify approaches modeling the user's context into contextual pre-filtering, contextual post-filtering, and contextual modeling approaches. The former two approaches apply non-contextual models to recommendation problems (with an additional initial or final filtering step), whereas contextual modeling leverages contextual information directly in the model, as the approach presented in this work does. In previous work [359], we showed that FM-based contextual modeling is able to outperform pre-filtering approaches.

Factorization Machines can be seen as an enhancement of CF [371]. FMs combine the advantages of support vector machines (SVM) and factorization models. Factorization enables the FM to model all interactions between variables in linear time [371], where the model variables can be metric, nominal or ordinal. Hence, different types of context can be integrated as nominal variables (e.g., weekdays or user groups). Recently, training algorithms for higher-order Factorization Machines (HOFM) have been proposed [52, 317] and shown to be useful for link prediction [52] or recommendations based on implicit feedback [486]. Inspired by the work of Rendle and Schmidt-Thieme [374], Field-aware FMs (FFM) perform a pairwise factorizing of the features, and thus, the factorization step is performed in separate latent spaces (fields). These have been applied for e.g., click-through rate (CTR) predictions [224]. More relevant for this work, FFMs have also been applied for music recommendation [87], where audio descriptors and mood information serve as input for the task of recommending music for a given text that the user currently writes. However, FFMs suffer from a quadratic complexity with the number of fields.

In this work, we present a multi-context-aware user model and recommendation approach. We utilize SVD to represent the user's situational context in a latent feature space and also model the user's general preference towards types of music. We rely on FMs to exploit interaction effects of different types of user context in a rating prediction and top-n recommendation scenario. To the best of our knowledge, this is the first music recommender system leveraging pre-computed nominal contextual variables in an FM-based recommender system, where interaction effects allow us to model which user listens to which type of music in which situation.

5.3. Problem Formulation

In the following, we formally define the *context-aware track recommendation problem* addressed in this paper. The basic input for such a context-aware track recommender system is a user-item matrix R, which holds prior user ratings for items (so-called interactions). It consists of m rows (corresponding to the number of users) and n columns (corresponding to the number of tracks). The elements r_{ij} of the matrix correspond to the rating a user i has assigned to track j. Based on this matrix, the track recommendation problem can be formulated as a rating prediction task as stated in Equation 5.1. The utility function f_R computes predicted ratings $\hat{r}_{ij}$ for <user,track>-pairs that do not feature a rating (yet). In classical CF models, f_R is learned from prior user-track interactions.

$$f_R = User \times Track \rightarrow Rating \tag{5.1}$$

f_R can be learned by matrix factorization techniques such as SVD [262] as depicted in Equation 5.2, where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal factor matrices that embed users and tracks onto a lower-dimensional space of latent features. Σ is a $m \times n$ diagonal matrix of singular values, estimating the impacts of the latent features on a rating r.

$$R = U \Sigma V^T \tag{5.2}$$

Using this representation, a single rating $\hat{r}_{ui}$ can be estimated using the dot product of the feature vectors of the user $\vec{u_i}$ and the item $\vec{v_j}$: $\hat{r}_{ui} = \vec{u_i} \cdot \vec{v_j}$.

Prior research has shown that people listen to different music during different activities [228] and people create playlists that are intended for certain activities [109]. Hence, depending on different user contexts, different tracks need to be recommended. This problem can be formulated as depicted in Equation 5.3, where f_{CR} is a utility function assigning predicted ratings $\hat{r}_{ij}$ to user u for track i given user contexts c [10].

$$f_{CR} = User \times Track \times Contexts \rightarrow Rating \tag{5.3}$$

Hence, the problem we study is the computation of track recommendations that match the current context of a user given his/her listening history including the contexts in which those tracks have been listed to.

5.4. Dataset

For our approach and the experiments conducted (cf. Sections 5.5 and 5.6), we require a dataset holding (i) listening histories of users, (ii) information about the situation in which those songs were listened to, and (iii) acoustic characteristics of these songs. Hence, we propose to leverage a publicly available dataset containing Spotify playlists [362]. We enrich this dataset with situational context information and audio characteristics of the tracks. The dataset contains the names of playlists which we will utilize to extract situational context information from (cf. Section 5.5). As for the audio characteristics, we gather and add content-based audio features for each track by querying the Spotify API[2]. These high-level features are well established in the MIR community and are widely used as a compact form for describing songs for modeling audio characteristics of tracks in an abundance of previous works in the field of music information retrieval (e.g., [21, 209, 314, 318, 359, 362, 508]). The employed content features are extracted and aggregated from the audio signal and comprise:

1. *Danceability* describes how suitable a track is for dancing and is based "on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity."

2. *Energy* measures the perceived intensity and activity of a track. This feature is based on the dynamic range, perceived loudness, timbre, onset rate, and general entropy of a track.

3. *Speechiness* detects the presence of spoken words in a track. High speechiness values indicate a high degree of spoken words (e.g., talk shows or audiobooks), whereas medium to high values indicate e.g., rap music.

4. *Acousticness* measures the probability that the given track is acoustic.

5. *Instrumentalness* measures the probability that a track is not vocal (i.e., instrumental).

6. *Tempo* quantifies the pace of the track in beats per minute.

7. *Valence* measures the "musical positiveness" conveyed by a track (i.e., cheerful and euphoric tracks reach high valence values).

8. *Liveness* captures the probability that the track was performed live (i.e., whether an audience is present in the recording).

[2] A detailed description of these features and the API can be found at `https://developer.spotify.com/web-api/get-several-audio-features/`.

For more detailed analyses on the acoustic features of user playlists, genre distributions among clusters or playlists, we refer the interested reader to the original papers describing the dataset [360, 362]. Furthermore, we provide an interactive playlist explorer tool[3] that allows exploring the dataset and its acoustic characteristics in detail.

5.5. Multi-Context-Aware User Model and Recommender System

The main idea of our approach is to compute recommendations based on the listening histories of users and contextual information regarding audio content and situational features. Particularly, we model and exploit pairwise interaction effects between these different contexts, between users and contexts and between tracks and contexts.

An overview of the proposed framework is given in Figure 5.1, where the steps taken to extract contextual information that is leveraged in the recommendation computation are outlined. As input for the proposed approach, we require a dataset of playlists (i.e., sets of tracks[4]) assembled by users as presented in Section 5.4. Based on this dataset (shown in Figure 5.1 as "Spotify Playlists Dataset"), we compute two types of contextual information for the computation of multi-context-aware track recommendations: (i) *playlist archetypes (clusters)* and (ii) *situational clusters.* For playlist archetypes ("Acoustic Cluster Component" in Figure 5.1), the input comprises the track id and the acoustic features for each track as provided by the Spotify API (cf. Section 5.4). This component computes the assignment of each track to an acoustic cluster. We describe this procedure in detail in Section 5.5.1. For computing situational clusters, the input comprises the track id and the names of the playlists the track is contained in. This component ("Situational Cluster Component" in Figure 5.1) computes the assignment of each track to a situational cluster. We detail this procedure in Section 5.5.2.

The extracted context information allows modeling *user preferences for tracks contained in certain playlist archetypes in a given situation.* We refer to the clusters mined from acoustic features as *acoustic feature clusters (AC)* and to the clusters mined from playlist names as *situational clusters (SC).* To finally incorporate this information (user, track, AC, and SC assignments) as input into a context-aware recommender system tackling the problem as stated in Section 5.3, we propose to utilize Factorization Machines (FM) [371] in a recommendation component ("Recommendation Component" in Figure 5.1). This allows capturing user preference towards a certain archetype of music in a certain situational context and to exploit the interaction effects between these two notions of context. this procedure results in a list of tracks sorted by the predicted relevance score for the given user in a given situation. We describe the recommendation computation in more detail in Section 5.5.3.

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

[4] In contrast to e.g., [150], we consider a playlist as an unordered set of tracks.

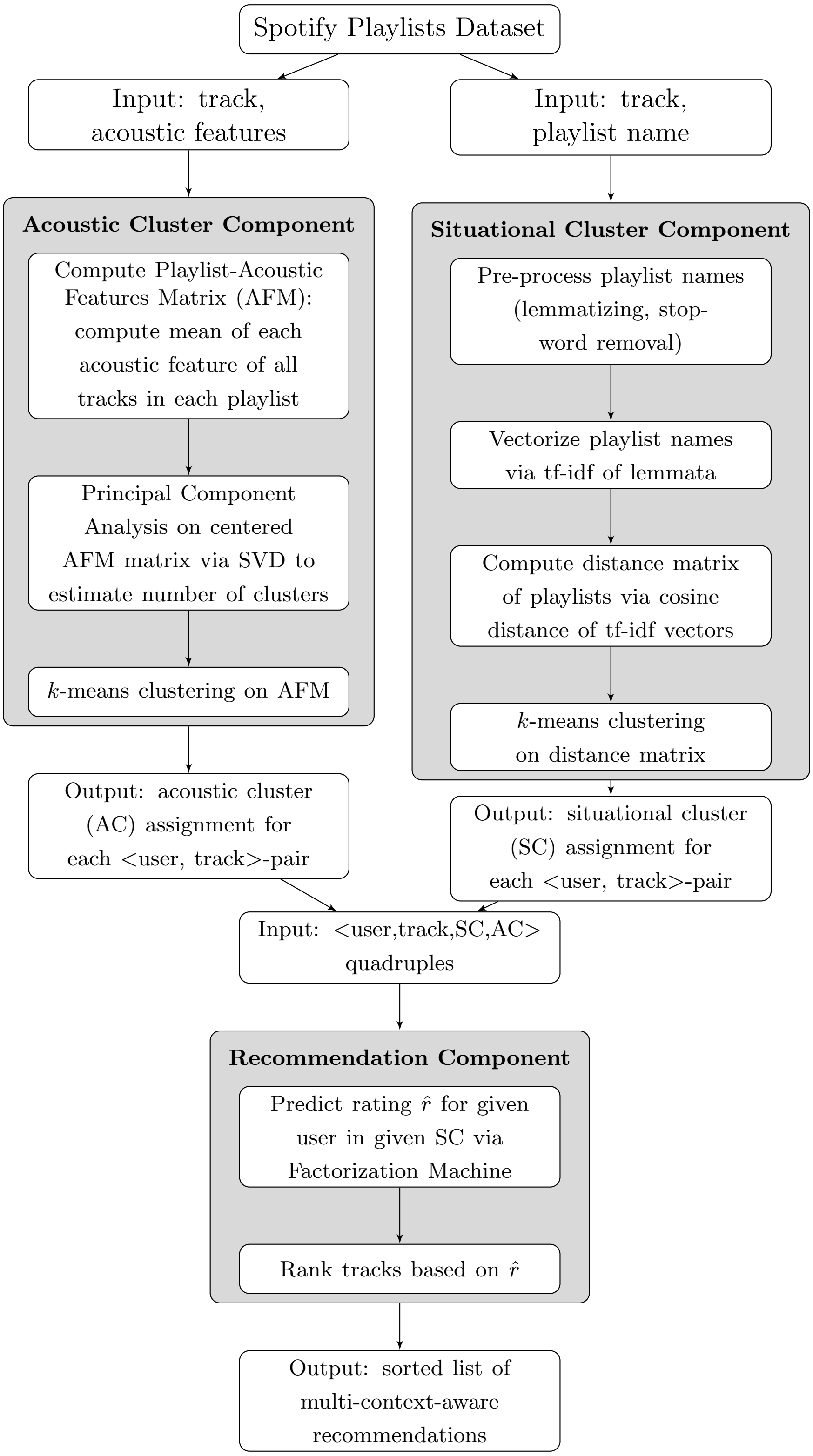

Spotify Playlists Dataset
Input: track, acoustic features
Input: track, playlist name
Acoustic Cluster Component
Compute Playlist-Acoustic Features Matrix (AFM): compute mean of each acoustic feature of all tracks in each playlist
Principal Component Analysis on centered AFM matrix via SVD to estimate number of clusters
k-means clustering on AFM
Situational Cluster Component
Pre-process playlist names (lemmatizing, stop-word removal)
Vectorize playlist names via tf-idf of lemmata
Compute distance matrix of playlists via cosine distance of tf-idf vectors
k-means clustering on distance matrix
Output: acoustic cluster (AC) assignment for each <user, track>-pair
Output: situational cluster (SC) assignment for each <user, track>-pair
Input: <user,track,SC,AC> quadruples
Recommendation Component
Predict rating $\hat{r}$ for given user in given SC via Factorization Machine
Rank tracks based on $\hat{r}$
Output: sorted list of multi-context-aware recommendations

Figure 5.1.: Proposed framework for computing multi-context-aware recommendations.

5.5.1. Playlist Archetypes

The proposed approach relies on clusters of playlists (archetypes) that share similar acoustic features (e.g., the tempo of the tracks contained). The major steps of this computation are also depicted in Figure 5.1. In a first step, we aggregate the eight acoustic features obtained via the Spotify API (cf. Section 5.4) of each playlist using the arithmetic mean. To ensure that the arithmetic mean is indeed representative, we analyze the dispersion of the tracks forming a playlist by comparing the mean and mean absolute deviation (MAD) [282] for each feature for each playlist. Here, we argue that the MAD is a robust measure with respect to outliers. With this analysis, we find that except for loudness, the variance of each of the acoustic characteristics of the tracks inside a playlist is low and the MAD is rarely higher than the mean. This allows us to conclude that aggregating the characteristics of the individual tracks to playlist characteristics using the mean is representative. For loudness, the variance among 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 analyses and refer to [362] for further analyses of the clusters. This aggregation step provides us with a lower-dimensional $m \times n$ matrix AFM (acoustic feature matrix), where each row represents a playlist and each column represents one of the proposed acoustic features. To find *archetypes* of music a user listens to, we apply factorization to the centered matrix AFM (all columns have a mean value of 0 and a standard deviation of 1) as this allows us to conduct a Principal Component Analysis (PCA) [349] via SVD [223].

The principal components (PCs) obtained by the conducted PCA allow explaining differences in playlists and, more importantly, estimate the number of acoustic clusters (ACs) to be obtained by the explained variance of each PC (squared singular values s_i^2 (diagonal of Σ)). For $k = 5$ clusters, the accumulated variance of the principal components is 85.64 and hence exceeds the 80% threshold. Thus, we set the number of acoustic feature clusters to be computed to $k = 5$. We compute the 5 clusters by applying k-means on the dimension-reduced matrix AFM. The clustering assigns each playlist and hence, implicitly each track, to one of five playlist archetypes that allow capturing a user's preferences towards certain types of music. We depict the result of this approach in Figure 5.2, where each playlist is represented by an integer that represents the cluster assignment. The clusters are marked by individual colors and are annotated with the respective acoustic features. From the conducted PCA, we observe that playlists that are highly influenced by instrumental and acoustic features are separated from the remaining playlists by the first PC (PC1). Furthermore, PC1 and PC2 separate energetic playlists with 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 5.2, separates playlists with high speechiness values from other playlists. The clusters (archetypes) obtained serve as one notion of context to be used for the computation of multi-context-aware track recommendations. We refer to our previous work in [362] for further details on this approach and analyses of the resulting clusters.

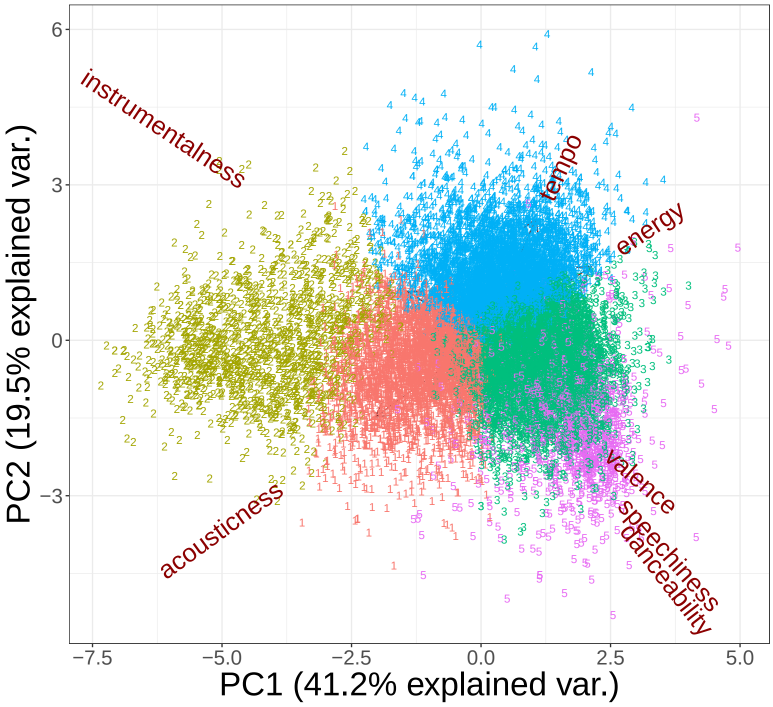


Figure 5.2.: Latent representation of playlist clusters.

5.5.2. Situational Clusters

Besides capturing musical preferences, we also aim to contextualize playlists by extracting situational context from the names of playlists. The underlying assumption here is that the names of playlists provide information about the situational context in which the playlist's tracks are listened to (e.g., “Summer Fun”, “Workout Mix”, or “Christmas”). Along the lines of [359, 361], we mine for activities and other descriptors (seasons, events, etc.) in the names of playlists.

As depicted in Figure 5.1, we firstly lemmatize all terms contained in playlist names using WordNet [319]. Next, we remove stop words and non-contextual terms (e.g., genre, artist, and track names) as these do not provide any contextual information. Furthermore, we utilize AlchemyAPI's entity recognition services[5] to remove playlist names that do not provide any contextual information. These are mostly playlist names that consist of artist names, track names, or genre descriptions. This results in a set of cleaned lemmata per playlist. However, those playlist names are rather short and heterogeneous. To create a meaningful distance matrix suitable for clustering playlists based on their names

[5] Please note that AlchemyAPI is now part of IBM Watson's Natural Language Understanding API: https://cloud.ibm.com/apidocs/natural-language-understanding.

is challenging. Therefore, we again use WordNet to enrich the lemmata of each playlist with semantically matching synonyms and hypernyms to create a more expressive term frequency-inverse document frequency (tf-idf) matrix. We derive this matrix by using a bag-of-words describing each playlist based on the derived lemmas, synonyms, and hypernyms. For the resulting bags of lemmata describing each playlist, we compute the term frequency-inverse document frequency (tf-idf) for each bag-of-lemmata representing a playlist name. Playlist similarities can now be computed by the pairwise cosine similarity of the resulting vectors. Based on these similarities, we span a distance matrix and find contextually similar playlists by applying k-means clustering. Along the lines of [361] (cf. Section 5.6), we empirically determine the number of clusters and set these to $k = 23$. This provides us with a set of 23 situational clusters capturing in which context a user listened to certain tracks. For instance, one of the clusters comprises Christmas songs, whereas another cluster comprises playlists and tracks related to a "summer" theme (e.g., containing playlist names such as "my summer playlist", "summer 2015 tracks", "finally summer" and "hot outside"). We refer to our previous work [359, 361] for further details on the computation of situational clusters and their usage in recommendation scenarios. In the next section, we present how we incorporate the gained contextual information in the computation of recommendations.

5.5.3. Recommendation Computation

The context extraction steps described in Sections 5.5.1 and 5.5.2 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. We now combine these clusters and the listening history of users in a joint user model that informs the track recommender system.

In this work, we propose to use FMs [371] for the computation of recommendations, i.e., to compute a predicted rating $\hat{r}$ for a given user i and a given track j, incorporating situational clusters (SCs) and acoustic feature-based clusters (ACs). We process the input for the rating prediction task as follows: first, <user,track>-pairs are enriched by the corresponding contextual cluster assignments, now forming <user,track,AC,SC>-tuples (as can also be seen in Figure 5.1). By adding a fifth column—rating r—to each entry in the dataset, we derive the input matrix R for our rating prediction problem to be solved (holding user, track, AC, SC, and rating columns).

Our dataset does not contain any implicit feedback by users (i.e., play counts, skipping behavior, or session duration). Therefore, we cannot estimate any preference towards an item as e.g., proposed by [199]. However, we assume that adding a track to a playlist signals a user's preference for the track. As the recommendation task is transformed into a rating prediction task, we require the dataset to also include negative examples. Therefore, for each user, we randomly add tracks the user did not interact with in a given situation (i.e., tracks t_j with $r_{i,j} = 0$ for the given user u_i) to the dataset until the listening history of each user in both the training and test sets are filled with 50% relevant and 50% non-relevant items for the user. We chose to oversample the positive class to avoid class imbalance and hence, a bias towards the negative class (the number of tracks not listened to is much larger than the number of tracks listened to for all users as naturally, users only listen to a small fraction of the songs available). Hence, for each unique <user,track,AC,SC>-tuple, the *rating* r_{ijsc} is defined as stated in Equation 5.4.

$$r_{ijsc} = \begin{cases} 1 & if\ u_i\ listened\ to\ t_j\ in\ SC_s\ and\ AC_c \\ 0 & otherwise \end{cases} \tag{5.4}$$

Based on this dataset, for computing the predicted rating $\hat{r}$, we model the influence of a user i, a track j, the situational cluster s, and the content-based cluster c on $\hat{r}$ in a FM. Relying on FMs, we are able to model all pairwise interactions, allowing to model the influence of the simultaneous occurrence of two variable values, i.e., of a track j and the contexts s and c or a user i and the contexts s and c. Furthermore, we model the interaction of the contexts c and s which can be interpreted as the influence of the current activity of a user (SC) on the playlist archetype (AC) and vice versa. This is shown in Equation 5.5: the FM computes $\hat{r}$ by estimating a global bias (w_0), estimating the influence of the user, track as well as the contexts ($\sum_{i=1}^{n} w_i x_i$) along with estimating the quadratic interaction effects of those ($\sum_{j=i+1}^{n} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j$). However, instead of learning all weights $w_{i,j}$ for the interaction effects, as traditional approaches such as logistic regression with quadratic interaction effects do, FMs rely on factorization to model the interaction as the inner product $\langle \vec{v_i}, \vec{v_j} \rangle$ of low-dimensional vectors [371].

$$\hat{r}_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j \tag{5.5}$$

The weights of the latter interaction effects are computed by applying matrix factorization during the FM optimization using a Markov Chain Monte Carlo (MCMC) solver as proposed by [152, 371].

Recently, higher-order Factorization Machines (HOFM) have been introduced, that allow for incorporating higher-order interaction effects [52, 317]. Aiming at further advancing the presented approach, we propose to also exploit 3-way interaction effects. A HOFM model is depicted in Equation 5.6, where a further factor capturing 3-way interactions is added (in comparison to 2-way Factorization Machines as depicted in Equation 5.5). Again, we rely on the Markov Chain Monte Carlo (MCMC) learning method.

$$\hat{r}_{HOFM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j + \\ + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{l=j+1}^{m} \langle \vec{v_i}, \vec{v_j}, \vec{v_l} \rangle x_i x_j x_l \tag{5.6}$$

5.6. Experimental Setup

In the following section, we present the experimental setup used to assess the performance of the proposed user model and recommendation approach. The proposed approach and the respective baselines were implemented in R, utilizing the libFMexe[6] wrapper for the original libFM implementation for FMs and the FactoRizationMachines[7] package for Higher-Order Factorization Machines. All experiments are based on the same input data as described in the following, before we present the methodology applied for the evaluation and the approaches evaluated.

5.6.1. Input Data for FM

We take the following steps to create the input data for the FM. In a first step, we apply the proposed dimension reduction and clustering methods on the dataset described in Section 5.4 to obtain the proposed acoustic feature (AC) and situational clusters (SC). To also allow looking into the impact of the clustering step for acoustic features, we evaluate a model that uses the individual acoustic features of tracks (AF). Therefore, we also add these features to the dataset. This results in a dataset containing <user,track,SC,AC,AF,rating>-tuples.

In the next step, we assign each track a rating value r. The rating indicates whether a certain user listened to a certain track in a certain situational cluster ($r = 1$) or not ($r = 0$) based on the underlying dataset as described in the previous Section. Please note that a user might listen to the same song in different situations (clusters), whereas a track always belongs to the same acoustic feature-based cluster. The final dataset used for the presented evaluation contains 956 unique users who listened to 485,304 unique tracks (we removed tracks we could not obtain acoustic features for and playlists for which we could not extract situational information from the playlist name). On average, a user in the dataset listens to 770.19 tracks (SD=2,168.62, Median=264.50).

A fragment of the resulting dataset is shown in Table 5.1. This excerpt shows that user 872 has listened to track 250246 (belonging to acoustic feature-based cluster 4) in

[6] https://github.com/andland/libFMexe

[7] https://github.com/cran/FactoRizationMachines

User	Track	SC	AC	AF_1	...	AF_7	Rating
872	309275	0	3	0.24		0.16	1
872	309275	1	3	0.24		0.16	0
872	250246	0	4	0.10		0.12	1
911	250246	2	4	0.10		0.12	1

Table 5.1.: Dataset fragment.

situational context cluster 0, whereas user 911 has listened to track 250246 in SC 2. This dataset forms the foundation for our experiments, which are presented in the next section.

5.6.2. Evaluation Methodology

To assess the performance of the proposed user models in a FM-based recommendation scenario, we employ the following evaluation method. For each user in the dataset, we perform a 5-fold cross-evaluation with random sampling on the user's tracks (i.e., for each fold, we utilize 80% of the user's tracks for training and the remaining 20% as test set such that each track is used once in the test set and four times as training data). We compute a predicted rating $\hat{r}$ for each track in the user's test set and hence, compute the probability whether a certain user listened to a certain track in a certain situational cluster. The evaluation metrics are computed for each fold separately and subsequently, averaged over all folds and in a final step, those metrics are averaged over all users. Due to the random selection of data for the folds, we allow the folds to contain an arbitrary number of relevant ($r = 1$) and irrelevant items ($r = 0$). However, as the distribution of the dataset we sample from has a 1:1 ratio between relevant and irrelevant tracks, the distribution within folds yields a similar distribution.

We aim to assess the performance of different recommendation models in both a top-n recommendation task as well as a rating prediction task. For the top-n recommendation task, we rank the items based on the predicted rating $\hat{r}$. We consider all tracks with a predicted rating below 0.5 ($\hat{r} < 0.5$) as irrelevant and do not consider these for recommendation. This proxy for the perceived usefulness of a user towards an item is finally used to rank the remaining tracks and cut off @n to retrieve a list of top-n track recommendations. Subsequently, we compute the *precision*, *recall*, and F_1 measures. We also evaluate the performance of a rating prediction task for the different user models. Therefore, we compute the root mean squared error (RMSE) for the predicted ratings $\hat{r}$ and the actual ratings r in the test set.

5.6.3. Evaluated User Models and Recommendation Approaches

To assess the effects of incorporating different contextual information encoded as clusters into a recommender system, we propose to evaluate and compare a theoretical random baseline, three baseline approaches, and a number of variations of the proposed multi-context-aware user model, which we detail in the following.

The theoretic random baseline (TR) guesses whether a track is relevant or irrelevant for a user. To outperform this random baseline, the values for RMSE have to be lower than 0.5. The probability of correctly guessing the correct rating in the sample space $\Omega = \{0, 1\}$ is $P(0) = P(1) = 0.5$ for each track. For 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 items and can be stated as $rec = \frac{n}{2 \cdot |relevant\ items|}$ assuming that every other guess $(\frac{n}{2})$ is a hit.

Furthermore, we employ the following three baseline methods: (i) a user- and content-agnostic approach that recommends the most popular tracks (MP) of each situational cluster; (ii) a collaborative-filtering baseline that incorporates the users' listening histories as input to the FM (CF); (iii) a CF model extended with the acoustic features of the tracks (AF), as this is known to work well [306] (again computed via the FM). Here, we use the individual acoustic features of each track and do not rely on acoustic feature clusters in this model. We consider this model a more advanced but nevertheless context-agnostic baseline. Please note that the goal of the work at hand is in investigating user models for multi-context-aware music recommendation scenarios and therefore, we aim to compare the different proposed user models and do not focus on the recommendation part. We argue that in previous work, we have already shown that utilizing Factorization Machines for context-aware recommendations contributes to recommendation performance [359], and hence, we rely on Factorization Machines and do not experiment with further recommendation approaches. However, the proposed CF baseline is a matrix factorization approach and hence, employs a different approach for the computation of recommendations.

Table 5.2 gives an overview of the evaluated models (combining a user model and recommendation approach) and the respective input data. We derive a set of extended models utilizing the situational clusters mined from the playlist names and playlist context derived from acoustic feature clusters as follows. Firstly, we evaluate a context-aware model extending the CF baseline by incorporating the situational clusters mined from playlist names (SC). Analogously, we extend the CF baseline by incorporating the playlist context (AC), the acoustic features (AF), and a combination of both (AF+AC). Finally, we evaluate a multi-context-aware model that combines both clusters (AC+SC) and a model incorporating the situational clusters mined from the playlist names combined with the eight individual acoustic features, the AF+SC model.

Model	CF	AF	AC	SC
TR (theoretic random baseline)				
MP (most popular baseline)				
CF (collaborative filtering baseline)	✓			
AF (CF + acoustic features baseline)	✓	✓		
SC (CF + situational clusters)	✓			✓
AC (CF + acoustic clusters)	✓		✓	
AF+AC (CF + acoustic features + acoustic clusters)	✓	✓	✓	
AF+SC (CF + acoustic features + situational clusters)	✓	✓		✓
AC+SC (CF + acoustic clusters + situational clusters)	✓		✓	✓

Table 5.2.: Overview of evaluated models (top: baseline approaches, bottom: variations of the proposed multi-context approach).

To also analyze the impact of interaction effects on the recommendation performance, we perform a final experiment based on the best performing user model detected in the previous experiments. We aim to assess the impact of different orders of interaction effects (no interaction effects, 2-way, and 3-way interactions) and also analyze the role of the number of latent features used in the factorization step.

5.7. Results and Discussion

In the following we first discuss the results of the top-n recommendation task evaluation (Section 5.7.1), followed by the results of the rating prediction task (Section 5.7.2). Subsequently, we present the results of the evaluation of the impact of interaction effects (Section 5.7.3).

5.7.1. Top-n Recommendation Task

In this evaluation, we aim to analyze the recommendation and ranking performance of the proposed models.

In a first step, we evaluate a recommendation list containing all recommendations (i.e., $n = R$, the number of tracks in the test set for a given user). The results of this analysis are depicted in Table 5.3. We observe a superior precision and F_1 performance of our proposed multi-context-aware AC+SC model jointly incorporating acoustic clusters (AC) and situational clusters (SC). The highest precision is reached by the AC+SC model (0.96), which outperforms the AF model (0.86) by 11.63%. Similarly, the AF+AC model reaches a precision of 0.85. We observe that both models jointly incorporating situational contexts and acoustic information (AC+SC and AF+SC) outperform all baselines, with the MP baseline reaching a precision of 0.73. In terms of the F_1-measure, our proposed

Approach	**Precision**	**Recall**	$\mathbf{F_1}$
AC+SC (CF + acoustic clusters + situational clusters)	**0.96**	0.81	**0.88**
AF+SC (CF + acoustic features + situational clusters)	0.80	0.78	0.79
AF (CF + acoustic features baseline)	0.86	0.68	0.76
AF+AC (CF + acoustic features + acoustic clusters)	0.85	0.65	0.74
AC (CF + acoustic clusters)	0.47	**0.85**	0.60
SC (CF + situational clusters)	0.46	0.83	0.59
CF (collaborative filtering baseline)	0.43	0.70	0.53
TR (theoretic random baseline)	0.50	0.50	0.50
MP (most popular baseline)	0.73	0.01	0.01

Table 5.3.: Top-R evaluation results (sorted by F_1, best results in bold).

AC+SC approach is 11.39% more accurate than a model exploiting acoustic features (no clustering) along with situational clusters (AF+SC) and 18.92% more accurate than a model relying solely on the individual acoustic features (AF). Furthermore, the AC+SC model is 49.15% more accurate than a model solely exploiting situational clusters (SC).

The context-agnostic CF baseline is outperformed if contextual clusters are incorporated into the model in isolation: in terms of F_1, a model solely exploiting acoustical clusters (AC) is 13.21% more accurate than the CF baseline and a model solely leveraging situational clusters (SC) outperforms the CF baseline by 11.32%. However, clusters in isolation cannot outperform a context-agnostic model incorporating acoustical features. Models that incorporate acoustical features constantly perform better than models without. 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 more efficiently in our scenario. In a later analysis (top-10 recommendations) in this section, we find that acoustical features are especially suitable for recommending tracks from the long tail. We suspect that a similar behavior causes the good recall performance of the AC model. However, we argue that in a music recommendation scenario, precision is more important to users than recall [46]. In comparison to the context-agnostic CF baseline, which only considers each user's listening history as input, our results show that AC+SC and AF+SC constantly outperform the CF baseline substantially.

For the top-n recommendations evaluated in this experiment, in terms of the F_1-measure, the AC+SC model on average performs 15.14% better across all n than the AF approach, which is the best performing approach that does not leverage situational clusters.

Inspecting the baselines, we observe that all proposed models that combine different contexts, as well as the AF model, outperform the CF and TR baselines in terms of precision and F_1. In terms of recall, the CF baseline is outperformed by the AC+SC, AF+SC, AC, and SC approaches. The MP baseline (recommending the most popular

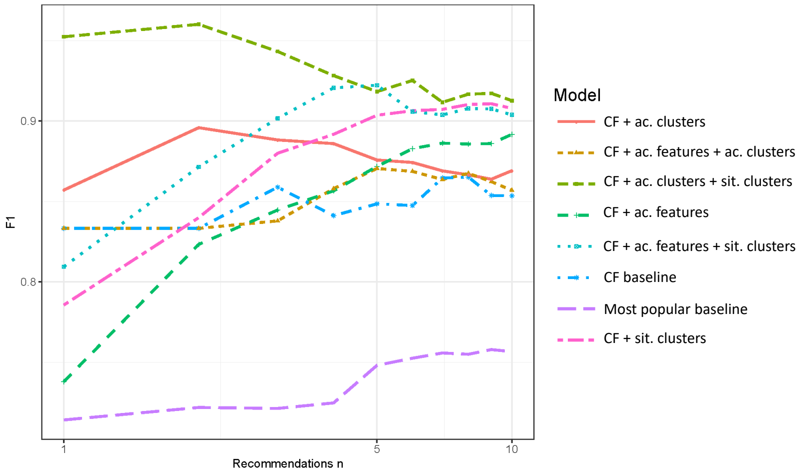


Figure 5.3.: F_1 for $n = 1 \dots 10$ recommendations.

items in the respective situational cluster) reaches reasonably good precision values. We explain this behavior of the MP baseline by the fact that the natural cap of the recall measure is rooted in the long-tailed distribution of the play counts, where popular tracks with high play counts among several users are rare [23]. Hence, the set of "good" recommendations of the MP approach is limited to this small amount of popular tracks, naturally limiting its recall performance.

Generally, user satisfaction has been shown to be highest when presenting the user with a short top-list of items naturally assuming that this recommendation list contains a sufficient number of relevant items [57]. Therefore, we evaluate the top-n performance of the proposed recommender system for a small number of n. Figure 5.3 depicts F_1 for $n = 1 \dots 10$, where we observe that the AC+SC model with an average F_1@10-score of 0.93 outperforms all other approaches. Notably, it outperforms the AF+SC model with an average F_1@10-score of 0.89 by 3.70%. Moreover, models leveraging situational clusters outperform all other models: the AC+SC model is the most accurate model, followed by the AF+SC model and the SC model. This is in contrast to the F_1@R-score results presented previously and a deeper analysis showed that situational clusters increase the precision only for a limited number of recommendations n. Incorporating SCs is beneficial for a small number of recommendations n but limits the discovery of new items in the long tail and hence, limits the performance for a large number of n. We believe that this is one of the reasons why the hybrid AC+SC and AF+SC models outperform all other approaches in both evaluations.

We observe that models that incorporate acoustic features along with situational clusters provide the best performance independently of the number of recommendations n. Our experiments also show that for a small number of recommendations n ($n \leq 10$), incorporating situational substantially impacts the recommendation performance. Moreover, the AC model leveraging acoustic clusters performs better than the AF model that leverages all acoustic features for small numbers of n. However, this does not hold for larger n, where it is important to be able to recommend tracks from the long tail. This long tail includes tracks with low play counts (i.e., non-mainstream, niche music). ACs group users who enjoy listening to similar music, which is sufficient for small n and for users with a rather narrow and less diverse music taste. In this context, we suspect that ACs favor mainstream music over less common music. However, to recommend tracks from the long tail, the system needs to accurately model the user's preferences in more detail by incorporating individual audio features (AF) of the tracks in the listening history of the user. Our experiments show that additionally incorporating the situational context (SC) improves the recall and F_1 for both, short and long lists of recommendations and precision can also be improved for short recommendation lists. Hence, we believe that the findings based on the evaluation of the top-n recommendations show that context is vital for improved recommendations, which is also in line with previous findings (e.g., [33, 34]). While the performance SC and AC in isolation indeed shows the importance of situational context, we can also show that incorporating both clusters along with the interaction effects is beneficial for the performance of the system. We analyze the impact of interaction effects further in Section 5.7.3.

5.7.2. Rating Prediction Task

To get a deeper understanding of the recommendations computed by FMs in relation to the individual models evaluated, we also evaluated a rating prediction task. The FM-component in our recommender system computes a predicted rating $\hat{r}$, i.e., the probability of a user listening to a certain track in a certain situational cluster. Hence, $\hat{r}$ can be seen as a proxy for the perceived usefulness of a user towards an item and hence, can be evaluated by measuring the error of this prediction. I.e., we evaluate this task by error metrics computed between $\hat{r}$ and r.

Approach	**RMSE**
AC+SC (CF + acoustic clusters + situational clusters)	**0.40**
AF+SC (CF + acoustic features + situational clusters)	**0.40**
AF (CF + acoustic features baseline)	0.44
AF+AC (CF + acoustic features + acoustic clusters)	0.47
AC (CF + acoustic clusters)	0.57
MP (most poplar baseline)	0.71
SC (CF + situational clusters)	0.72
CF (collaborative filtering baseline)	0.75

Table 5.4.: Rating prediction evaluation results (sorted by RMSE, best results in bold).

Table 5.4 depicts the results of the rating prediction measures computed over the test set in Table 5.4. Our results show that the AC+SC and the AF+SC models achieve the lowest RMSE values, which also is in line with the results of the evaluation of the top-n

recommendation task. Both models incorporating acoustic features and situational clusters (AC+SC, AF+SC) outperform a model solely using the situational clusters (SC) by 44.44.% and a model solely using acoustic-feature clusters (AC) by 29.82%, 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 strongly impact the performance of context-aware track recommendations. To investigate the impact of interaction effects, we compare the proposed FM to a FM that does not incorporate any interaction effects in a further evaluation in Section 5.7.3. Furthermore, in line with our findings of the top-n evaluation, the AF model is also able to capture user preferences well. Analogously to the previous evaluation, we show that this is particularly the case for tracks in the long tail, consequently, the AF model also performs well in the rating prediction evaluation.

Interestingly, the most popular (MP) approach outperforms the CF- as well as the SC-model. However, 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, and thus, we assume a predicted rating of $\hat{r} = 0$. In contrast, the FM approaches estimate $\hat{r}$, the probability of 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 naturally is an error for the other approaches (although the track is correctly classified) as these estimate $\hat{r}$ in [0,1]. 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 in the range between 0 and 0.5.

5.7.3. Impact of Interaction Effects

In a final set of experiments, we are interested in the extent to which the performance of the utilized FM is dependent on the number of latent features used for modeling the interaction effects in the FM and the impact of the order of interaction effects.

To estimate the impact of interaction effects on the recommendation quality, we compare the performance of a FM that does not exploit any interaction effects and a FM that leverages interaction effects based on the best user model detected (AC+SC). The results of these experiments can be seen in Table 5.5. These results show that adding interaction effects allows for a 17.41% higher F_1-score (0.88 vs. 0.75) and an increase in precision of 28.13%, while the recall values are comparable. This is also reflected in the RMSE of 0.41 for a model incorporating interaction effects and an RMSE of 0.67 for a model not incorporating these (improvement of 38.81%). This again strengthens our hypothesis that exploiting interaction effects is highly beneficial in such a scenario.

In a second experiment, we evaluate the performance of our 2-way FM dependent on the number of latent features. A boxplot presenting the results of this evaluation can be seen

Approach	**Precision**	**Recall**	$\mathbf{F_1}$
AC+SC (2-way interactions)	**0.96**	0.81	**0.88**
AC+SC (no interactions)	0.69	**0.82**	0.75

Table 5.5.: Impact of interaction effects: top-R evaluation, where the AC+SC model incorporates CF + acoustic clusters + situational clusters.

in Figure 5.4. We find that the best performance in terms of the F_1-measure is reached with $k = 20$ or $k = 5$. However, the differences among all configurations regarding the number of latent features are subtle. In fact, the difference is smaller than the standard deviation and hence, not significant. Therefore, we argue as there are no differences in performance and training the $k = 20$ model took approximately four times longer than the training of the $k = 5$ model in our experiments, choosing $k = 5$ seems a reasonable choice.

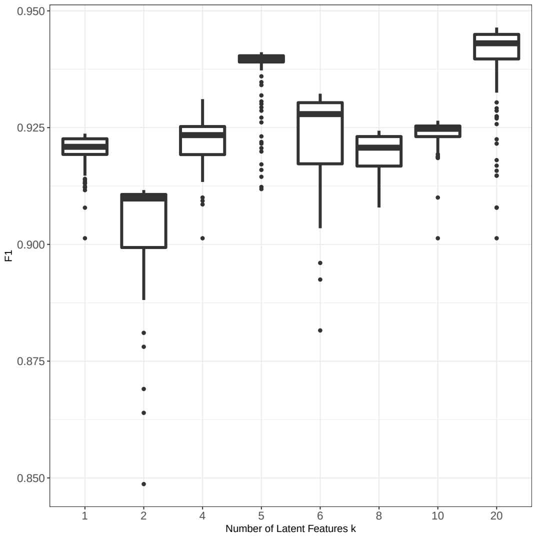


Figure 5.4.: F_1@R for different numbers of latent features k.

In a final evaluation, we are interested in the performance of higher-order Factorization Machines (HOFM) and hence, the impact of 3-way interaction effects in our scenario. Based on the results of our previous experiments regarding the number of latent features (and hence, the dimensionality of the factorization of interactions), we fixed k for the second-order dimensions at $k = 5$. The results of our comparison between a FM without any interaction effects (FM0), traditional FMs (FM), and HOFMs (HOFM) for the AC+SC model are depicted in Figure 5.5. The results show that also for HOFM, AC+SC is the model obtaining the best results. For HOFM, we observe a minor performance improvement of below 1% for both F_1 and RMSE. Please note that these experiments were performed using the HOFM library (cf. Section 5.6) to conduct a fair comparison among the three approaches (FM0, FM, and HOFM), which also explains the slight difference to the results of the previous experiments (which were performed using the original libFM library). However, as the standard deviation is larger than the mean, these differences are not significant. Hence, as a HOFM has no significant advantage regarding its F_1 performance and the fact that HOFMs naturally are a more complex model and thus, require higher computational efforts, we argue that relying on traditional FMs is a feasible and reasonable choice, which also is in line with previous findings [486].

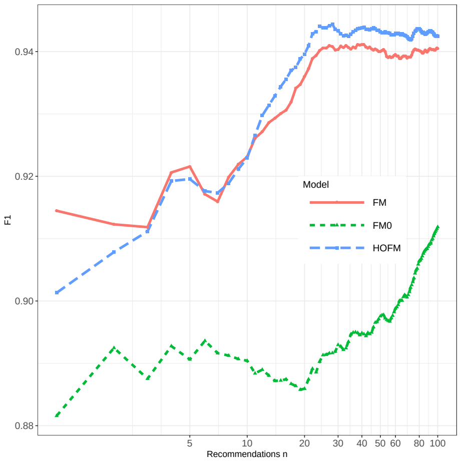


Figure 5.5.: F_1 for FM0 (no interactions), FM (2-way interactions) and HOFM (3-way interactions) for the AC+SC model for $n = 0...100$ (x-axis log-scaled).

5.8. Conclusion and Future Work

In this paper, we presented a multi-context-aware user model that jointly exploits (i) situational context extracted from the names of playlists, and (ii) playlist archetypes that share acoustic characteristics to model which kind of music is listened in certain situational contexts. Both the situational context and musical preferences are represented as cluster assignments. For the computation of recommendations, we use Factorization Machines which use the proposed user model as input to exploit interaction effects among contexts. In extensive offline experiments, we show that (i) the integration of situational context improves the precision of music recommender systems and that (ii) acoustic features and thereby, a user's musical taste, are particularly beneficial to retrieve tracks a user likes from the long tail. Our experiments show that interaction effects between situational context and musical preferences (playlist archetypes, acoustic clusters) provide the most accurate recommendations.

We believe that the use of Factorization Machines allows for easily extending our current approach with further notions of context such as emotion [505] or culture [511]. Also, the extraction of situational information from the names of playlists may also benefit from utilizing factorization models [102]. From an evaluation perspective, we also aim to investigate beyond-accuracy metrics [230] in future work to look into how contextual factors might affect aspects such as diversity of recommendation lists or novelty.

6. Content-based User Models: Modeling the Many Faces of Musical Preference

Publication

E. Zangerle and M. Pichl. Content-based User Models: Modeling the Many Faces of Musical Preference. In *Proceedings of the 19th International Society for Music Information Retrieval Conference 2018 (ISMIR 2018)*, ISMIR '18, pages 709–716. ISMIR, 2018. DOI: 10.5281/zenodo.1492515

Abstract

User models that capture the musical preferences of users are central for many tasks in music information retrieval and music recommendation, yet, it has not been fully explored and exploited. To this end, the musical preferences of users in the context of music recommender systems have mostly been captured in collaborative filtering-based approaches. Alternatively, users can be characterized by their average listening behavior and hence, by the mean values of a set of content descriptors of tracks the users listened to. However, a user may listen to highly different tracks and genres. Thus, computing the average of all tracks does not capture the user's listening behavior well. We argue that each user may have many different preferences that depend on contextual aspects (e.g., listening to classical music when working and hard rock when doing sports) and that user models should account for these different sets of preferences. In this paper, we provide a detailed analysis and evaluation of different user models that describe a user's musical preferences based on acoustic features of tracks the user has listened to. perform an evaluation of the models' capabilities to represent a user's preferences well.

6.1. Introduction

In the last decade, the amount of tracks available on streaming platforms has literally exploded. Users are supported in exploring and wading through these music collections by means of personalization—mostly by recommender systems that provide users with a list of tracks they might like to listen to. Such personalization is central for the success of streaming platforms as it eases the task of discovering new and enjoyable music for users.

For music information retrieval (MIR) and particularly, for personalization tasks in this context, modeling the musical preferences of users is naturally a central aspect. Yet, user modeling for MIR and music recommender systems (MRS) has hardly been investigated [55, 412, 413]. To this end, music recommender systems have mostly been realized by means of collaborative filtering (CF) methods [261] or more advanced factorization approaches [262], where recommendations are based on interactions between users and items. Such systems are agnostic to content features as recommendations are computed based on the similarity of users (or items) based on their co-occurrence in the listening histories of all users. On the other hand, (the less adopted) content-based recommender systems [299] compute recommendations based on the similarity of content descriptors of tracks. Also, hybrid recommender systems combining CF- and content-based approaches have been proposed [67].

In the field of MIR, tracks are traditionally characterized by content descriptors—these range from detailed features such as MFCCs [297] to high-level content descriptors such as acousticness, tempo or danceability (e.g., provided by the Spotify platform[1]). While these features are widely used to characterize single tracks, for a user model that captures the user's preferences well, these features have to be aggregated across all tracks the user has listened to. To this end, Pichl et al. [362] utilized content descriptors of tracks for representing a user's musical preference by computing the average acoustic features across all tracks the user has listened to. They also find that users create different playlists that feature different acoustic characteristics—implying that these playlists correspond to different sets of preferences of a user (which may naturally be context-related) and stress the need for more comprehensive user models to describe users' musical preferences [362]. Similarly, Wang et al. [481] state that people prefer different music for different daily activities. Along these lines, we argue that users may exhibit different preferences depending on the context and e.g., listen to more energetic music when doing sports or calming music when being at home [481]. These different preferences cannot be sufficiently reflected in a model that averages the characteristics of all the tracks a user listened to. In a probabilistic user model, Bogdanov et al. [55] characterize a user in a semantic feature space derived from low-level content features by utilizing Gaussian Mixture Models.

[1] https://developer.spotify.com/web-api/get-several-audio-features/

In this paper, we build upon and extend these previous works by proposing different user models to describe the musical preferences of users based on content descriptors of tracks. We perform a large-scale evaluation of these models in a track recommendation task based on 8 million listening events of 13,000 users. Our experiments show that utilizing a user model based on a user's specific preferences regarding different types of music (modeled probabilistically by GMMs) complemented with a user's general musical preference achieves the best results. Our results show that in terms of recommendation quality, the proposed models contribute to substantially improved recommendation performance. We believe that our findings can contribute to improved user models for music recommender systems and generally, MIR tasks.

The remainder of this paper is organized as follows. Section 2 discusses related work and Section 3 presents the features utilized and the dataset underlying our experiments. Section 4 presents the user models proposed. Section 5 details the experimental setup and Section 6 presents the results of our study, which are discussed in Section 7. Section 8 concludes the paper and discusses future work.

6.2. Related Work

Generally, Schedl et al. [412, 413] note that the user and his/her preferences are often not considered when it comes to MIR and MRS tasks. Particularly, the authors lay out that user modeling for such tasks has hardly been explored and evaluated yet.

To this end, content descriptors have widely been used in MIR and MRS. For similarity search, often a content-based similarity measure is used for matching queries and a music database [97, 294, 429, 516]. In the context of music recommender systems, Yoshii et al. [501] propose a hybrid recommender system that combines collaborative filtering via user ratings and content-based features modeled via Gaussian Mixture Models over MFCCs by utilizing a Bayesian network. Also, Liu [294] investigates different distance metrics for content-based recommender systems. Recently, also deep learning-based hybrid MRS have also been proposed [482]. In regards to user modeling for MRS, Bogdanov et al. compute a user's musical preferences by a set of exemplary tracks that the user enjoyed. They model the user's preference in a latent semantic space based on a set of diverse content features and propose a set of similarity-based recommender systems. One system models a user by a Gaussian Mixture Model based on the proposed semantic audio feature space. The authors evaluated these recommender systems in a user experiment with twelve users. As for musical preferences of users, Pichl et al. found in a large-scale study of Spotify users that music streaming users listen to different types of music. Those types can be observed via k-means clustering of content descriptors of tracks. They also found that users organize their music in playlists based on these types and stress the importance of more comprehensive user models to describe users' musical preferences [362]. Along these lines, we specifically investigate user models that are solely based on content descriptors. We propose six user models and compare these in a large-scale offline study based on a recommendation task comprising 13,000 users and 8 mio. listening events.

6.3. Dataset and Features

The main data source used in our experiments is the publicly available LFM-1b dataset [408], which provides the full listening histories of 120,322 Last.fm users. For each listening event (i.e., a certain user listening to a certain track), information about the track, artist, album and user is available. Besides the information contained within the LFM-1b dataset, we also require content features to describe tracks. Following the lines of, e.g., [21, 314, 362], we propose to rely on the Spotify API[2] to gather the following content descriptors for each track:

1. *Danceability* describes how suitable a track is for dancing and is based "on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity."

2. *Energy* measures the perceived intensity and activity of a track. This feature is based on the dynamic range, perceived loudness, timbre, onset rate and general entropy of a track.

3. *Speechiness* detects presence of spoken words. High speechiness values indicate a high degree of spoken words (talk shows, audio book, etc.), whereas medium to high values indicate e.g., rap music.

4. *Acousticness* measures the probability that the given track is acoustic.

5. *Instrumentalness* measures the probability that a track is not vocal (i.e., instrumental).

6. *Tempo* quantifies the pace of a track in beats per minute.

7. *Valence* measures the "musical positiveness" conveyed by a track (i.e., cheerful and euphoric tracks reach high valence values).

8. *Liveness* captures the probability that the track was performed live (i.e., whether an audience is present in the recording).

[2] A detailed description of these features and the API can be found at https://developer.spotify.com/web-api/get-several-audio-features/.

These features are high-level descriptors of the acoustic content of tracks. We argue that they are nevertheless representative and hence, the obtained results should give a good impression on the differences of the user models. We expect our findings to also hold for more complex and lower-level content descriptors such as e.g., Mel-Frequency Cepstral Coefficients (MFCC) [297].

To obtain these features for all tracks of the dataset, we apply the following steps: we perform a conjunctive query for the <track, artist, album>-triples extracted from the LFM-1b dataset using the Spotify search API[3] to gather the Spotify URI of each track. This URI is subsequently used to query the acoustic features API[4]. Finally, we add tracks for which can obtain all required features to the dataset[5]

Since the set of tracks a user listened to may also contain outlier tracks that may distort the user profile, we propose to remove outlier tracks from this set by applying the median absolute deviation (MAD) outlier detection method [282]. We consider a feature value an outlier if it is not within $M \pm a \cdot MAD$, where M is the median of this particular feature across all tracks of a user and MAD is the median absolute deviation of these values. We consider a value an outlier if it is not within within three MADs around the median, setting a rather conservative threshold $a = 3$ as proposed by [282]. Lastly, a track is considered as an outlier in the list of tracks of a particular user if one of its features is considered an outlier and consequently removed from the user listening history.

Applying this procedure results in a dataset of 55,149 users, 394,944,868 listening events and 3,478,399 distinct tracks. We randomly sample users from this dataset for our experiments, where we require each user to have more than 100 listening events to ensure that our user models are representative. We present basic statistics about the resulting dataset in Table 6.1. As can be seen, on average, each user has listened to 651 tracks.

6.4. User Models

In the following, we present the proposed user models to capture user's listening preferences. We specifically focus on modeling users solely by acoustic features of tracks they listened to and deliberately neglect other information that could contribute to a user model (e.g., demographic user aspects, cultural information or further contextual features that might improve MRS and MIR performance).

[3] https://developer.spotify.com/web-api/search-item/

[4] https://developer.spotify.com/web-api/get-several-audio-features/

[5] Except for tempo, all of these features are given in the range of $[0, 1]$ and for tempo, we apply a linear min-max scaling.

Item	Value
Listening Events (LE)	8,457,205
Users	12,995
Tracks distinct	965,293
Min. LE per User	1
Q_1 LE per User	252
Median LE per User	478
Q_3 LE per User	826
Max. LE per User	21,660
Avg. LE per User	650.80 (± 713.99)

Table 6.1.: Dataset statistics.

6.4.1. Feature Space

Based on the users, tracks and their acoustic features within the dataset, we perform the following steps prior to the computation of the user models. Most of the proposed models require clustering tracks based on their acoustic features to find groups of tracks that exhibit similar features. Given that we aim to perform a large-scale analysis of the proposed user models (we perform the analysis on 8 million tracks and 13,000 users), these clustering computations are computationally intensive. Hence, we firstly perform a proximity-preserving dimension reduction on the input data by applying UMAP (Uniform Manifold Approximation and Projection) [311]. Also, the use of latent representations of elements in the musical ecosystem (users, tracks, etc.) has been to be effective in MIR and MRS tasks [92, 280, 324]. In our experiments, we compute a 2-d latent representation of tracks for the computation of user models. This allows us to inspect the resulting clusters visually during the development of the user models and, more importantly, reduces cluster computation time substantially, which naturally permits better scalability for larger datasets.

6.4.2. User Models

For modeling user preferences for musical tracks and their characteristics, we naturally require models for both tracks and users as we utilize a user's model and compare it with track models to find suitable similar tracks that may be recommended to the user.

As for modeling tracks and their characteristics, we rely on their acoustic features (AF; e.g., danceability or tempo). However, for users we require more sophisticated user models, as these have to represent a possibly extensive and diverse set of tracks and their characteristics to eventually represent a user's musical preferences. We propose user models that are based on clusters of similar tracks and utilize a user's membership in these clusters (i.e., the fact that user has listened to tracks that belong to a given cluster) to get a fine-grained representation of the many faces of the listening preferences of a given user. For determining such clusters and computing the membership of tracks in these clusters, we experiment with two approaches: (i) we utilize k-means clustering to find tracks that exhibit similar acoustic features and use the characteristics of these clusters to characterize users; and (ii) we apply Gaussian Mixture Models (GMM) [312] as these allow to model a track by the computed probability density function regarding the GMM's components. Based on a track's density functions, we derive a set of GMM-based user models. Generally, the idea is that based on these clusters or components, we aim to model a user based on the characteristics of one or multiple of these track clusters.

In the following, we describe the proposed user models to capture the musical preferences of users. An overview of the user models and the features used to characterize users and tracks is shown in Table 6.2.

Content avg: In a baseline model, we utilize the eight acoustic features of all tracks a user has listened to and compute the average across all tracks of a user for each of the features presented in Section 6.3. This allows us to describe a user with his/her average listening behavior, breaking a user's preferences down into eight acoustic features. Please note that in the remainder of this paper, we refer to models as *Content*-models if the representation of the user or a track relies on acoustic features.

Content avg, sd: This model is built upon the Content avg model, which we extend by adding the standard deviation of each of the acoustic features across all tracks of a user. We expect the added SD to mitigate the effects of averaging a large number of features that potentially differ substantially as users may listen to music with highly diverse acoustic characteristics. We again consider this model a baseline that additionally quantifies to which extent the user's musical preferences vary regarding the acoustic features of his/her listening history.

Content binary k-means: In this model, we rely on the clusters computed by a k-means clustering of all tracks within the dataset in the computed 2-d latent space. In a next step, we attribute each of the tracks a user has listened to a cluster and do a majority vote on the clusters to obtain the cluster that holds most of the user's tracks. We subsequently model a user using the characteristics of the cluster that contains the majority of the user's track. To represent this cluster, we compute the average of the eight acoustic features of all tracks contained in the cluster and add the according standard deviations. Single tracks are represented by its acoustic features. We consider this a rather simple model as we assign the user to a single cluster and hence, limit the model to a single preference scope.

Content weighted k-means: The previous model is limited as it is restricted to a single preference scope. To tackle this problem, we propose the Content weighted k-means model in which we now aim to address multiple sets of preferences of a user.

Model	User Features	Track Feat.
Content avg	user AF avg	AF
Content avg, sd	user AF avg and SD	AF
Content binary k-means	avg. AF of single cluster	AF
Content weighted k-means	weighted avg. AF of clusters	AF
GMM	avg. densities of user's tracks	GMM densities
Content binary GMM	avg. AF of single GMM comp.	AF
Content weighted GMM	weighted avg. AF of GMM comp.	AF
GMM + Content avg, sd	GMM and user AF avg and SD	GMM, AF

Table 6.2.: Overview of evaluated models (AF stands for acoustic features, GMM for Gaussian Mixture Model and SD for the standard deviation).

Therefore, we again rely on the k-means clusters, however, we compute a weight for each cluster based on the number of tracks a user has listened to in each cluster. Based on the user's weights for each cluster, we compute a weighted average for each acoustic feature to represent the user, where each cluster is again characterized by its average acoustic features and its standard deviation. Again, in this model each track is represented by its acoustic features.

GMM: In this model, we utilize a Gaussian Mixture Model [312] for representing both the track and the user. Therefore, we compute Gaussian components and represent a track by its probability densities regarding the GMM components. For users, we compute the average probabilities for each component across all of the user's tracks to model a user's musical preferences by using the GMM components. We consider this model a proxy, as it does not directly utilize acoustic features to represent a track, but the probabilistic assignments of a track to a set of groups of tracks (components).

Content binary GMM: In contrast to the pure GMM model, this model relies on content features instead of probability densities to represent a user. Analogously to the Content binary k-means model, we rely on GMM to assign the user's tracks to components. In particular, we assign the tracks found in the user's listening history to GMM components. In a next step, we select the component with the highest number of user tracks assigned to, where we assign a track to the component with the highest probability density for the track. The user is then modeled by the characteristics of the selected component (again using the average and standard deviation across all acoustic features of the tracks assigned to the component), whereas each track is again represented by its acoustic features.

Content weighted GMM: This model is again analogous to the content weighted k-means model. However, we rely on a GMM to assign a user's tracks to certain a component as described in the previous model. Based on these assignments, we analogously compute the weighted mean and standard deviation for each acoustic feature for each GMM cluster to represent a user and the characteristics of tracks are captured by their acoustic features.

GMM + content avg, sd: In this model, we combine the GMM components baseline model with the content avg, sd baseline model and hence, represent a user by his/her component weights regarding the Gaussian Mixture Model and further add the average and standard deviation across all acoustic features of the user's tracks. Similarly, a track is represented by its GMM densities and its acoustic features.

We also performed experiments on representing users and tracks with cluster or component assignments only and did an analysis of further combinations of the proposed models. However, the results were below the evaluated baselines and hence, we do not list these models here.

6.5. Experimental Setup

We model the evaluation of the proposed user models as a recommendation task, where we aim to obtain a ranked list of tracks that are of interest to the user. For this task, we rely on Gradient Boosting Decision Trees. Particularly, we utilize the popular XGBoost system [93], a scalable end-to-end tree boosting approach that has been shown to be highly suited for recommendation tasks [31, 340]. For the training phase of the tree, we set the training objective to be the binary classification error rate (i.e., the number of wrongly classified tracks in relation to all tracks classified, where tracks with a predicted probability of relevance larger than 0.5 are classified as relevant for the given user, and all other tracks are considered irrelevant for the user). Please note that we deliberately chose a classification-based recommendation approach and refrained from utilizing more elaborate recommender approaches such as context-aware matrix factorization [34] or tensor-based factorization approaches [235] as we aim to focus on user modeling aspects in this paper.

For the recommendation task carried out, we require a rating for each track in the dataset to define whether a given track was listened to and thus, considered relevant for a given user. Hence, we add a binary factor *rating* to the processed dataset: for each unique $<user, track>$-combination, the *rating* $r_{i,j}$ is 1 if the user u_i has listened to track t_j. Due to a lack of publicly available data, our dataset does not contain any implicit feedback of users (i.e., skipping behavior, session durations or dwell times during browsing the catalog). This is why we cannot estimate any preference towards a track a user not listened to as proposed by [199]. Thus, we assume tracks the user has not listened to as negative examples [199] and hence, assign a rating of 0 to these tracks. Even though there is a certain bias towards negative values as some missing values might be positive, Pan et al. [341] found that this method for rating estimation works well. To perform the proposed recommendation task via classification, we require the dataset to also include negative examples. Therefore, for each user, we add random tracks the user did not interact with (i.e., tracks t_j with $r_{i,j} = 0$ for the given user u_i) to the dataset until both the training and test sets are filled with 50% relevant and 50% non-relevant items. We chose to oversample the positive class to avoid class imbalance and hence, a bias towards the negative class.

Using the resulting data set, we train a XGBoost model that performs a binary classification on the relevance of tracks for a given users. We extract the probabilities underlying the classification decision to rank tracks by their probability of relevance in the recommendation task.

To evaluate the performance of the proposed user models in regards to recommendation quality, we perform a per-user evaluation. Therefore, we use each user's listening history and perform a *leave-k-out* evaluation (also known as hold-out evaluation) [64, 106] per user. Based on the dataset that now contains both positive and negative samples for each user, we compute a hold-out set of size k: along the lines of previous research [139, 185], we randomly select 10 positive samples (tracks that the user has listened to) and 100 negative samples (tracks the user has not listened to). These 110 tracks form the test set for each user, whereas the recommender system is trained on the remainder of the dataset. We compute the predicted ratings for the tracks in the test set and rank the track recommendation candidates w.r.t. the probability that the current track belongs to the positive class in descending order. For our experiments, we consider all predicted probabilities > 0.5 as a predicted interaction and thus, we consider these items as relevant, all others as irrelevant and hence, not added to the list of recommendations.[6]

Based on the predicted ratings, we compute *precision*, *recall*, and the F_1-measure to assess the top-10 accuracy [105]. We evaluate the 10 top ranked tracks as too many track recommendations might provoke choice overload and hence, is not feasible. The problem of choice overload has been addressed by Bollen et al. [57] who state that user satisfaction is highest when presenting the user with Top-5 to Top-20 items—naturally assuming that the recommendation list contains a sufficient number of relevant items for the user. For assessing the overall *precision*, *recall*, and F_1-measure of the evaluated recommender systems, we compute the measures for each individual user and compute the average among all users. For computing the *recall* measure, all relevant items in the test set are considered, independent of the number of recommendations. Thus, there is a natural cap for *recall*, namely the number of recommendations divided by the number of relevant items in the test set.

For the tuning of XGBoost parameters, we did a preliminary cross-evaluation aiming to optimize precision values for the proposed models and hence, set the number of maximum

[6] This distinction between the two classes is also utilized by XGBoost for binary classification tasks based on logistic regression.

trees to learn the models to 2,000. For all other parameters, we relied on the default settings. For the training and tuning of k-means and GMM for the creation of the user models, we performed the following steps. For k-means, estimated the number of clusters by utilizing the elbow method based on the within-cluster sum of squares. For the given dataset, we estimated the number of clusters to be 5. For the GMM, we performed a training phase based on expectation maximization and determined the number of components using the Bayesian Information Criterion (BIC), which resulted in a total of 9 components for the GMM.

6.6. Results

We present the results of our evaluation for a recommendation list of size ten in Table 6.3 and in a precision-recall plot depicted in Figure 6.1.

The best results are obtained by the GMM + Content avg, sd model, reaching a precision@10 of 0.771 and a recall@10 of 0.427 and hence, achieving substantially higher precision and recall scores than any other model. Comparing the results of this model to the GMM model (relying on solely the assignments to GMM components) and the Content avg, sd baseline model shows that those two models individually perform substantially worse than when combined. When inspecting the results of the GMM model, we find that solely relying on the GMM density functions does not suffice to represent a user's musical taste. Particularly, all content-based GMM or k-means models achieve higher performance when applied in isolation. However, combining a simple content-based approach that provides acoustic features regarding the user's general preferences, with GMM, provides us with a representative user model. This suggests that the GMM model captures a user's diverse preferences regarding the detected components and hence, his/her distribution in preference towards specific types of music, while his/her general preferences are captured by the average acoustic features and the according standard deviation.

Model	**Prec**	**Rec**	**F$_1$**
GMM + Content avg, sd	**0.771**	**0.427**	**0.632**
Content k-means weighted	0.606	0.316	0.400
Content k-means binary	0.573	0.300	0.383
Content binary GMM	0.569	0.298	0.381
Content weighted GMM	0.569	0.298	0.381
GMM	0.231	0.122	0.226
Content avg, sd	0.161	0.089	0.241
Content avg	0.159	0.087	0.241

Table 6.3.: Precision, Recall and F$_1$@10, ordered by F$_1$.

Our results also show that the user models based on k-means clusters slightly outperform the methods based on GMM components (1.8% in recall, 3.7% in precision). Please note that for k-means we determined the number of clusters to be five, whereas we created nine GMM components (as described in Section 6.5). Our findings regarding the number of clusters are also in line with previous analyses on playlists [362], where the authors found that clustering the tracks within playlists into five clusters allows for cohesive and homogeneous clusters.

The weighted k-means approach achieves better results than the binary k-means approach. This seems natural as the former incorporates the user's membership in all clusters, whereas the latter does a majority vote and utilizes the resulting (single) cluster to characterize the user. However, this does not hold for the GMM-based approaches. While the differences between the weighted and binary k-means approaches are marginal, for GMM there is no difference between weighted and binary Content GMM.

The proposed baseline model Content avg achieves the lowest values regarding recall, precision and F_1. Adding the standard deviation to this model hardly impacts the results. We initially suspected that adding the SD to the model may allow mitigating the effects of aggregating possibly highly different tracks as we aggregate across all tracks of a user (regarding their acoustic features), however, this is not confirmed by our experiments. In preliminary experiments, we also used different representations of clusters: while we now utilize the mean acoustic features and the according SDs, we also used only the mean features. We found that the SD contributes only marginally as the dispersion of tracks in regards to acoustic features is already captured by the individual clusters/components and hence, the tracks contained in a single cluster/component are more homogeneous. We also experimented with models that utilize user-cluster assignments for k-means, however, those models achieved inferior results. In contrast, representing those clusters by the average acoustic features across all contained tracks seems to be representative. Combining k-means cluster assignments with content-based models also lead to inferior results, which we lead back to the fact that the GMM probability densities provide more information than sheer cluster-assignments.

Generally, we conclude that content features strongly contribute to user models and that grouping tracks into clusters (k-means) or components (GMM) and solely relying on the assignment to those clusters or components is not sufficient for a representative user model. Finding groups of similar tracks to represent users by user-group assignments via the tracks a user listened to is not expressive enough. Naturally, utilizing content features allows to compute higher-dimensional similarities between users and their tracks (in our experiments, 8 dimensions) and hence, a more fine-grained notion of similarity.

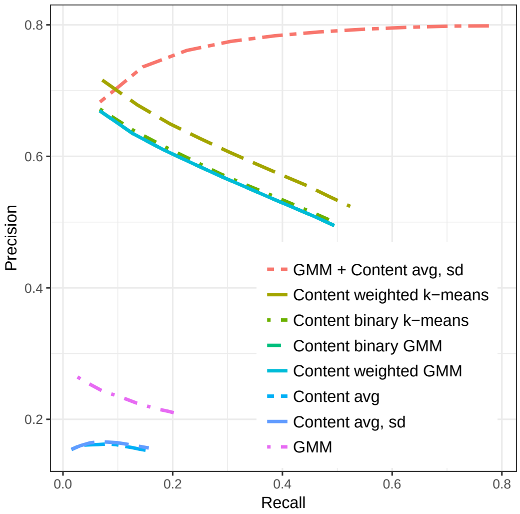


Figure 6.1.: Precision-Recall curves for all models.

6.7. Discussion

We find that a GMM that captures the specific preferences of a user towards a set of nine types of music (captured by nine GMM components) complemented by the general musical preference of a user (captured by the avg. acoustic features of his/her tracks) provides the best results.

Regarding the limitations of this study, we note that the content descriptors utilized are aggregated high-level features. This allowed us to keep the feature space smaller and to specifically focus on the user modeling aspects. Furthermore, this evaluation is solely based on aspects related to the content of tracks and no further user-related aspects as e.g., proposed by Schedl et al. [417]. Lastly, while the proposed models characterize users based on their interest in different clusters/components and hence, are able to build more specific user models, we still represent each cluster/component by the mean acoustic features of the tracks contained, which naturally limits the user model's specificity. However, we believe that our findings are a valuable contribution to advance user modeling for MIR and MRS and to foster further research in this direction.

6.8. Conclusion and Future Work

We proposed and evaluated a set of user models for describing the musical preference of users by leveraging content descriptors of tracks the user has listened to. We find that a GMM complemented by the user's general musical preferences describes a user's different musical preferences best. We believe that our findings can contribute to improved user models for music recommender systems and generally, MIR tasks. In future work, we aim to investigate methods to combine the models evaluated by e.g., ensemble methods. Furthermore, we aim to tackle the problem that our current model still computes average acoustic features across a large number of tracks.

7. Leveraging Affective Hashtags for Ranking Music Recommendations

Publication

E. Zangerle, C.-M. Chen, M.-F. Tsai, and Y.-H. Yang. Leveraging Affective Hashtags for Ranking Music Recommendations. *IEEE Transactions on Affective Computing*, 12(1):78–91, 2021. DOI: 10.1109/taffc.2018.2846596

Abstract

Mood and emotion play an important role when it comes to choosing musical tracks to listen to. In the field of music information retrieval and recommendation, emotion is considered contextual information that is hard to capture, albeit highly influential. In this study, we analyze the connection between users' emotional states and their musical choices. Particularly, we perform a large-scale study based on two data sets containing 560,000 and 90,000 #nowplaying tweets, respectively. We extract affective contextual information from hashtags contained in these tweets by applying an unsupervised sentiment dictionary approach. Subsequently, we utilize a state-of-the-art network embedding method to learn latent feature representations of users, tracks and hashtags. Based on both the affective information and the latent features, a set of eight ranking methods is proposed. We find that relying on a ranking approach that incorporates the latent representations of users and tracks allows for capturing a user's general musical preferences well (regardless of used hashtags or affective information). However, for capturing context-specific preferences (a more complex and personal ranking task), we find that ranking strategies that rely on affective information and that leverage hashtags as context information outperform the other ranking strategies.

7.1. Introduction

People listen to music for different reasons: to relieve from boredom, fill uncomfortable silences, social cohesion and communication, emotion regulation, etc. [298, 404]. From an affective computing point of view, it is interesting to investigate the relationship between a user's musical preference and the user's emotional state. There have been many psychological studies on the role of music in emotion regulation [298, 394, 472]. The emotional state of a listener has also been considered as important contextual information in building recommender systems [33, 146, 387]. A possible application is to build a system that monitors people's emotion and predicts how to subliminally impact them by recommending different music pieces. However, as the emotional state of a user is hard to capture in a large-scale study, most existing studies are conducted in a laboratory setting. It remains unclear to which extent such findings can be generalized to the real-life usage of music [484].

Seeing the popularity of social microblogging websites such as Twitter[1], we have new opportunities to study real-world music listening behavior at scale [183, 358, 408, 509]. Most interestingly for our study, Twitter allows for gathering so-called #nowplaying tweets [509], which are tweets describing the track a user is currently listening to. One such example tweet is "#nowplaying Crazy For You by Adele #Happy". In this example, the user not only publishes the music track and artist he/she is listening to, but also adds a hashtag (i.e., keywords or phrases starting with the symbol '#') describing his/her concurrent emotional state. Users add these hashtags spontaneously in real life, and there is an abundant number of such #nowplaying tweets with affect-related hashtags. We are therefore particularly interested in how the affective hashtags within a tweet are related to the user's musical preferences. For this purpose, we consider only #nowplaying tweets containing hashtags that represent some notion of emotion (i.e., contextual information), and aim to study their role in providing contextual affection-aware music recommendations tailored to the user's current emotional state and musical preferences. We have the following two research questions (RQs) to be answered:

- **RQ1:** How can affective contextual information contribute to improving personalized ranking of track recommendation candidates?

- **RQ2:** How can we computationally represent the affective contextual information in a #nowplaying tweet?

There has been excellent work on context-aware recommendation and representation learning [10, 235, 333, 371, 433], sentiment analysis from text [265, 329, 331, 342, 428], as well as emotion-based music recommendation [37, 88, 117, 178]. The main novelty of this study lies in the way we study the aforementioned two RQs by adapting existing techniques. Specifically, our study differentiates itself from the prior arts in the following aspects:

[1] http://twitter.com

First, we propose to employ, and compare the result of, two evaluation tasks to highlight the importance of contextual information (Section 7.3). For a given user and a context, the first task requires ranking the relevance of a set of tracks that are picked at random, whereas the second task requires ranking a set of tracks that are known (from the training set) to be associated with the user. While the first task is mainly about the *general preference* of a user (i.e., which tracks a user would like), the second task requires modeling the *context-specific preference* of a user for we already know that all the candidate tracks are liked by the user but only one of them can be ranked at top given that specific listening context. An algorithm cannot perform well if it does not know how a user's emotional context affects his or her musical preference. In comparison, existing work on context-aware recommendation usually focuses on the algorithms and simply takes the full catalog of data in their evaluation [235, 333, 371, 433]. Such an evaluation method does not distinguish between tracks that have been known to or not by users, making it hard to assess whether an algorithm learns the general preference or context-specific preference. This is less a concern for a general recommendation algorithm but is critical in addressing our RQs.

Second, to investigate the affective contextual information embedded in the #nowplaying tweets, we propose to treat the user-track-hashtag association as a graph and use state-of-the-art *network embedding* methods [172, 355, 455] to learn latent feature representations of users, tracks and hashtags (Section 7.4.2). By experimenting with different combinations of the representations (Section 7.4.3), we can test different assumptions about the underlying association between users and tracks. For example, a user can be represented by the user's own latent representation (denoted as "user"), but can also be represented by the average latent representation of the hashtags the user has used before in his or her tweets ("usertag"). Similarly, a track can be represented by its own latent representation ("track") or by the average representation of the hashtags the track has been associated with by different users ("tracktag"). As the hashtags are restricted to be affect-related ones, "usertag" and "tracktag" may respectively capture the *general emotional tendency* of a user and a track. A possible consequence is that, if a track is typically listened to in a specific emotional context across users, "tracktag" may outperform "track" in the above-mentioned second task, for "tracktag" encodes affective information in a more explicit way. In total, six ranking methods are considered. To our best knowledge, testing the representations in such an emotion-centered way has not been attempted before.

Third, in addition to the latent representations, we employ different sentiment dictionaries proposed in the literature of sentiment analysis [166, 196, 203, 334, 382, 459] to implement two ranking methods that solely rely on the sentiment scores (Section 7.4.1). In this way, we can study RQ2 using two approaches: based on the latent representations and based on the sentiment scores. Our experiments (Section 7.5) show that for the first task (capturing a user's general preferences), utilizing latent representations for users, tracks and hashtags contributes to better and more personalized ranking results. However, for the second, more complex and personal context-specific task, the

Characteristics	**Original [509]**	**#NP560k**	**#NP90k**
Listening events	21,501,261	564,301	85,528
Tracks distinct	654,012	51,045	31,454
Artists distinct	79,011	8,210	8,020
Users distinct	176,909	9,431	9,336

Table 7.1.: Data set statistics.

sentiment-aware ranking methods outperform the other ranking methods. This finding implies that the more personal and complex a ranking task gets, the higher the influence and significance of affective information gets.

Finally, although emotion-based music recommendation is not new, existing work mostly relies on user data collected in a controlled environment and the scale is usually small [37, 117, 178]. In contrast, our study is based on a large collection of Twitter data (around 560K) that contain real-world music listening information (Section 7.2). We will share the data with the research community for reproducibility and for promoting research in this direction.

7.2. Data Sets

Generally, we require a data set that provides information about the listening behavior and emotional states of users for conducting the proposed experiments. Therefore, we employ the #nowplaying data set compiled by Zangerle *et al.* [510] for the study, as this data set provides the required information. The data set is composed of #nowplaying tweets crawled via the Twitter API [468] and provides the timestamp when the tweet was sent, an anonymized user id, the tweet's source (how it was sent), the contained artist name and track title. An example listening event is: <`2016-05-12 16:26:42, '7bd5237385a73c54265cd02aa136dbecdb88a0b8', 'Twitter Web Client', 'Hello, Goodbye', 'The Beatles'`>.

To gather a data set that allows for representative user profiles, we chose to extract all listening events of users who have sent a minimum of ten listening events in the years 2014 and 2015 from the #nowplaying data set. The characteristics of the resulting data set are shown in Table 7.1 (column "Original"). For our study, we focus on tweets for which we can detect a sentiment value by using the methods described in Section 7.4.1, as only this data allows to evaluate the influence of affective contextual information on the quality of track recommendation rankings. Therefore, we remove the listening events that do not contain any hashtag that we can obtain a sentiment score for, leading to a subset containing 564,301 listening events. Statistics of this *#NP560k* data set are listed in Table 7.1.

	#NP560k					**#NP90k**				
Characteristics	Median	Q3	Max	Mean	SD	Median	Q3	Max	Mean	SD
Listening events per user	2.0	4.0	69,197.0	59.83	1,125.48	2.0	4.0	463.0	9.16	31.70
Listening events per track	2.0	8.0	1,821.0	11.05	33.65	1.0	2.0	1,031.0	2.72	10.42
Tracks per user	2.0	4.0	69,197.0	59.83	1,125.48	2.0	4.0	463.0	9.16	31.70
Distinct tracks per user	2.0	4.0	3,500.0	10.95	78.73	2.0	4.0	319.0	6.26	19.03
Hashtags per user	2.0	5.0	86,855.0	74.10	1,446.10	2.0	5.0	1,025.0	10.84	39.27
Distinct hashtags per user	1.0	3.0	207.0	2.65	5.41	1.0	3.0	108.0	2.57	4.43
Hashtags per track	1.0	1.0	6.0	1.24	0.47	1.0	1.0	6.0	1.17	0.44
Distinct hashtags per track	1.0	1.0	6.0	1.23	0.46	1.0	1.0	6.0	1.16	0.44

Table 7.2.: Five-number-summaries of (left) the #NP560k data set and (right) the #NP90k data set. We show (from left to right) the median, 3rd quantile (Q3), maximum, mean, standard deviation (SD) of the individual characteristics for both data sets. The minimum and 1st quantile for both data sets are all ones for all the characteristics.

Table 7.2 presents the five-number-summaries describing the tagging and listening behavior of users within the #NP560k data set. We also list the tracks and listening events per user (overall and distinct) as well as the number of tags per user and per track (overall and distinct). We observe that while the maximum number of listening events per user, track and hashtag are very high, the mean, median and the 1st and 3rd quartile of these characteristics are substantially lower implying that these distributions are skewed and do not follow a normal distribution. Also, we observe a small number of users and tracks that feature profoundly higher numbers for the analyzed characteristics in comparison to the majority of users and tracks. Such heavy-tailed distributions have been shown to be prevalent in social networks [270, 278].

To mitigate the effects of this distribution on the performance of our approach, we additionally apply an outlier removal method to the #NP560k data set. Particularly, we keep all users within the 99th percentile of the distribution and remove the others, as this outlier removal method has been shown to be suited for highly skewed distributions [386]. This presents us with a smaller data set, referred to as the *#NP90k* data set in this paper. Table 7.1 depicts the basic characteristics and Table 7.2 presents the five-number-summaries for the #NP90k data set. While the #NP560k data set features a number of heavy users (and hence, heavy-tailed distributions), these are removed in the #NP90k data set, making it less skewed.

Please note that we deliberately removed the hashtags #nowplaying, #listeningto and #listento from the data sets as at least one of those hashtags is contained in every listening event and hence do not add any further information.

In these data sets, not only listening events are tagged with hashtags, also tracks can transitively (via the listening event the track is mentioned in) be tagged with the respective hashtags. Similarly, we tag users with hashtags if a given hashtag is used within one of the listening events sent by the user. We reason that hashtags have been shown to serve two roles [495]: i) users wanting to express his/her thoughts, feelings and opinions, ii) using hashtags to tag the content of the tweet. For our study, both factors are important as we aim to evaluate the potential of affective hashtags for ranking music recommendations.

7.3. Evaluation Methods

In the following we present the methods deployed for the evaluation of the ranking methods presented in Section 7.4.

All the experiments are conducted based on the #NP560k and #NP90k data sets presented in Section 7.2. For conducting the evaluation, we need to split the data sets into training and test sets and apply different splitting methods for the two data sets. For #NP560k, we perform the following *per-user* split: for each user in the data set, we randomly choose 70% of his/her listening events as training data and the remaining 30% as test data. We believe that this splitting method allows for mitigating the skewness of the data set as the split is performed on a per-user bases and hence, is robust against dominating users in data set (i.e., users with a high number of listening events). In contrast, for the #NP90k data set that has already been cleaned from outliers (and is therefore less skewed), we employ a *global split* that randomly picks 70% out of all listening events of all users for the training set and uses the remainder of listening events as test data. These contrasting splitting approaches permit to investigate the connection between a user's emotional context and the user's concurrent musical preference independent of the size of user profiles.

For both of these splitting approaches and the underlying data sets, the latent features of nodes are computed for the items within the training set only and do not incorporate any information from the test set.

The basic input items for our evaluation are listening events, which are tweets containing information about a track a user listened to. The workflow of the evaluation is as follows. Based on a listening event randomly chosen of the test set (hereafter referred to as "input listening event") including its affective hashtags, we aim to evaluate the ranking methods proposed in Section 7.4.3. We consider the track contained in this input listening event as our ground truth data and our goal is to find ranking methods that rank this ground truth track first in the recommendation list.

From a recommender system point of view, our data sets represent *implicit feedback* data [199, 371]— the data sets represent traces of user behavior and they only provide us with the tracks a user has listened to. Our data set does not contain any implicit feedback by users (i.e., play counts, skipping behavior, session durations or dwell times during browsing the catalog). As most papers dealing with implicit feedback [199], what we can do is to assume that the user likes these tracks. We are not aware of the tracks that the user dislikes. In other words, all the listening events contained in our data sets are *positive* data, and there is no *negative* data at all. This has been referred to as the *one-class* problem [341]. To learn discriminative latent feature representations, we need to perform so-called *negative sampling* [172, 355, 455] to include user-track-hashtag associations that are not present in our data sets as negative data (cf. Section 7.4.2). Likewise, for evaluation, we need to sample negative data to test how our ranking methods can identify the positive track and rank it on top of the list.

Different ways to perform negative sampling for the test set represents different evaluation tasks. As described in Section 7.1, it is possible to use the full data catalog as negative data, as many prior work on context-aware recommendation do [235, 333, 371, 433], but in this way we are not able to properly study the two RQs. Alternatively, we consider the following two evaluation tasks.

Firstly, we aim to evaluate whether our proposed approach is able to capture the general listening preferences of users. Therefore, we propose the **POP_RND** task, where we add nine randomly chosen tracks to the list containing the input listening event to populate the list. This task allows us to evaluate whether our approach is able to capture the general listening preferences of users.

Secondly, we aim to evaluate a context-specific scenario where we model the sentiment of a user as the context in which tracks are listened to by users. We consider this scenario as more complex than solely capturing the general listening preferences of users. Therefore, we propose the **POP_USER** task, where we randomly pick nine tracks the user has previously listened to and add these to the set of tracks to be ranked. This requires the user to have a listening history comprising at least ten tracks.

As this task selects tracks that are associated with the user, we are able to evaluate the performance of incorporating contextual sentiment and hashtag information in the ranking computation as we have to employ context information to be able to rank those tracks effectively. Therefore, we argue that this task allows us to directly evaluate the usefulness of hashtags and sentiment scores.

We propose to evaluate the ranking performance of our approach for sets of ten tracks. In the field of recommender systems, a set of 5–10 recommendations is most appropriate which also corresponds to the capacity of short-term memory [320]. Furthermore, the work by Bollen *et al.* [57] underlines this choice as the authors conducted an experiments showing that presenting users with a large number of good and valuable items is counterproductive as the choice of an item becomes inherently difficult for the user.

The (unordered) set of ten tracks resulting from the proposed data generation is subsequently used as input for the recommendation ranking evaluation. In the next step, we apply the proposed ranking methods to this set of track recommendation candidates.

As for the evaluation metric, we rely on the mean reciprocal rank (MRR) metric [477] as defined in Equation 7.1 to evaluate the rank of the single correct item. We choose MRR as we are only interested in how the ranking methods perform in regards to ranking the ground truth track as high as possible in the ordered list of recommendation candidates. Ranking the ground truth one as the first item yields a RR of 1, ranking it second yields a RR of 0.5, etc. As the lists to be ranked in our experiments only contain a single correct item, the maximum RR obtainable is 1.

$$RR(item) = \frac{1}{rank(item)} \tag{7.1}$$

In total, we repeat this evaluation procedure for a set of 20,000 listening events randomly extracted from the test set for all the proposed ranking methods and consequently, determine the mean RR (MRR) for the set of all ranked recommendation lists contained in the evaluated set of listening events. We use these to compare the performance of the ranking methods and the underlying latent features.

7.4. Computational Methods

In the following section, we present the methods utilized for leveraging affective hashtags for music recommendations.

7.4.1. Sentiment Detection for Hashtags

The extraction of sentiment polarity from a given word, sentence or text has been studied widely [331, 342]. Also, sentiment detection in the context of Twitter has been addressed by research [265, 329, 428]. In this study, we focus on hashtags that express emotion. Therefore, we aim to detect the sentiment of hashtags in a first step. For this task, we rely on a widely used unsupervised sentiment detection method: so-called sentiment lexica [331]. In principle, sentiment lexica are dictionaries of words, where each word is annotated with its polarity (and possibly, also the strength of this polarity). For detecting the sentiment of a term, it is simply matched against a given lexicon. In the following, we describe the specific steps taken for assigning sentiment values to the hashtags within our data set.

Sentiment Dictionaries

We rely on well-established dictionaries which have been widely used and evaluated [166, 382]. In particular, we use the dictionaries that provide both the best coverage and performance in terms of accuracy according to the study of Ribeiro *et al.* [382]. Table 7.3 contains an overview of the adopted lexica.

The AFINN dictionary [334] was assembled from a set of different word lists (e.g., obscene words and internet slang words) and manually annotated by a single annotator. Opinion

Name	**#Terms**	**Coverage**		
		Hashtags	LEs	Tracks
AFINN [334]	2,477	57.64%	46.87%	54.67%
Opinion Lexicon [196]	6,789	44.86%	44.35%	47.48%
SentiStrength [459]	2,546	71.23%	73.97%	71.50%
Vader [203]	7,517	57.63%	57.80%	61.54%

Table 7.3.: Sentiment dictionaries and their coverage of the #NP560k data set; 'LE' is a shorthand for listening event.

Lexicon [196] is computed by using antonym and synonym relationships among words and using this information to deduce scores for adjectives. The SentiStrength lexicon [459] is based on a manually annotated dictionary, which is subsequently improved by adjusting the scores by machine learning techniques. The Vader dictionary [203] is also created by human annotation and is particularly geared towards sentiment analysis of social media texts.

Affection Computation

Based on the set of hashtags contained in the data set, we employ the following strategy to resolve hashtags against a given sentiment dictionary. Firstly, we aim to match full hashtags against the dictionary, both lowercased. However, this does only match hashtags which represent full proper English words (e.g., `#happy`). For all other hashtags, we apply lemmatization, as provided by the Python NLTK Wordnet package[2]. Consequently, we match these lemmata against the lemmata of the given lexicon. For hashtags that cannot be resolved directly or after lemmatization, we assume that these are either compound words or can simply not be found in the given dictionary. As for compound hashtags, these can either be written as camel case as e.g., `#IAmHappy` or a concatenation of multiple lowercased terms as e.g., `#feelinggood`. We aim to split these compound hashtags to match the single terms contained in the hashtag against the sentiment dictionaries. Therefore, we use the split words (i.e., {I, am, happy} for the above example) to represent the hashtag. As for camel case-hashtags, we split the hashtag using upper-case characters as delimiters. The problem of segmenting all-lowercase compound hashtags has already been addressed in literature [441]. Therefore, we follow previous work [466] to split these up. As the sentiment lexica are limited to English words, we base our approach on a dictionary of 109,582 English words[3]. We split the original hashtag at each position and look into whether the prefix is contained in the dictionary. If it is contained, we recursively repeat the procedure until we find an optimal result. Once we found a representation of

[2] http://www.nltk.org/howto/wordnet.html

[3] http://www-01.sil.org/linguistics/wordlists/english/

the hashtag that consists of a set of individual terms using the methods described, we match these terms against the sentiment lexicon individually. We assign the hashtag the mean of the sentiment scores of all terms contained in the original hashtag.

Table 7.3 features an overview of the coverage of the different sentiment lexica. Here, we list the percentage of hashtags that can be resolved against the various dictionaries for the #NP560k data set. Similarly, we also list the fraction of listening events and tracks that can be assigned a sentiment value using the respective dictionary. Please note that despite the difference in size between the #NP560k and #NP90k data sets, the coverage of the individual sentiment lexica is comparable for both data sets and hence, we only list the coverage numbers for the #NP560k data set here. Besides using these single sentiment dictionaries, we also propose to exploit the variety and extended coverage of the combination of multiple dictionaries by using the mean value of all sentiment values across all available sentiment dictionaries gathered for a given hashtag.

The lexica have different ranges of polarity scores (e.g., AFINN from –5 to 5, and Opinion Lexicon from -1 to 1). Therefore, before computing the mean values, we normalize them by using linear min-max feature scaling.

7.4.2. Computation of Latent Features

While there are many methods for learning feature representations of users, tracks and hashtags from listening data, we employ the so-called network embedding technique [172, 355, 455] to learn such representations. The task of network embedding is to learn the low-dimensional representations of vertices in an information network that can capture and preserve the network structure in the representations. Such network embedding methods are useful for modeling data containing heterogeneous types, which is exactly the case here as we have users, tracks and hashtags to be modeled. In particular, we build a graph containing these three object types from the data sets and then use a network embedding algorithm to learn their representations. Although several network embedding models have been proposed, for this work we use the well-known DeepWalk approach [355]. DeepWalk is one of the most popular network embedding algorithms owing to its effectiveness in modeling the global structure of the input graph [355]. The algorithm learns low-dimensional latent feature descriptions for all the vertices (including users, tracks, and hashtags) within the graph, allowing us to compute their similarity in a joint feature space.

Given a graph G and its vertices V and edges E, the objective is to model the following conditional probabilities:

$$p(v_j|v_i) = \frac{sim(v_i, v_j)}{\sum_{v_k} sim(v_i, v_k)}, \tag{7.2}$$

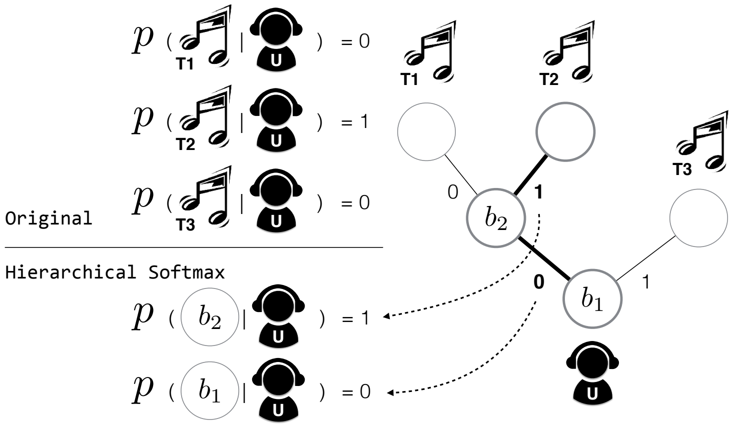


Figure 7.1.: Suppose the task is modeling the user-track pair (u, t_2), the original modeling function requires to compute all pair-wise estimations (i.e., $(u, t_1), (u, t_2), (u, t_3)$) while the transformed hierarchical softmax computes the estimations with only the passing nodes (i.e., $(u, b_1), (u, b_2)$).

where *sim* is a function that measures the similarity between two vertices v_i and v_j based on their representations. Therefore, the vertices sharing similar neighbors receive similar conditional probability distribution.

To obtain the low-dimensional representations of each vertex, we further conduct a mapping function $\Phi : v \in V \mapsto \mathcal{R}^{|V| \times d}$ in Equation (7.2) to map the node v into a low-dimensional vector $\Phi(v)$, which also satisfies the above objective function:

$$p(v_j|\Phi(v_i)) = \frac{sim(\Phi(v_i), v_j)}{\sum_{v_k} sim(\Phi(v_i), v_k)}. \tag{7.3}$$

Instead of computing all vertex pairs, which is quite expensive owing to the number of given vertices, DeepWalk factorizes the conditional probability using the hierarchical softmax [322] to assign each vertex a series of binary codes by Huffman tree construction. For a pair (i, j), suppose the path to vertex v_j is identified by a sequence of tree nodes $[b_0, b_1, \cdots]$, then the final objective is converted to multiple binary classification predictions:

$$p(v_j|\Phi(v_i)) = \prod_l p(b_l|\Phi(v_i)). \tag{7.4}$$

Thereby, the computational complexity is reduced by the transformation from $\mathcal{O}(|V|)$ to $\mathcal{O}(\log |V|)$. Figure 7.1 shows the idea of the hierarchical softmax transformation.

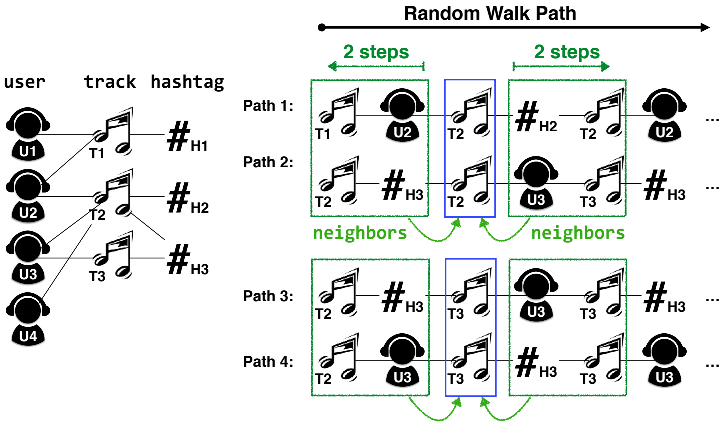


Figure 7.2.: The paths (right) are generated by random walks according to the given graph (left). When the window size is set to two, the connected vertices within two steps are treated as the context information of the centered vertex. In this way, vertices with similar neighbor connections will receive similar connection status and thus, receive similar probability distributions.

In order to further efficiently learn the low-dimensional representations, DeepWalk also uses sampling techniques for conducting the concept of random walk, which is a common technique when dealing with a huge graph. Figure 7.2 plots the stochastic random walk of DeepWalk. It uses a random walk strategy to generate a path, and then adopts a certain window size to dynamically sample the observed pairs (v_i, v_j) for modeling Equation (7.4). The appearing probability also implies the reachability between two vertices, and can serve as *sim* in Equation (7.2). Finally, the vertices which share similar neighbors will pass similar tree node paths and thus, receive similar representations. For optimizing the representations, stochastic gradient descent [61] is utilized.

Differently designed graphs underlying the DeepWalk computation can lead to different assumptions on the relationships among the vertices in the graph.

In a conventional recommendation task, the connections between users and tracks (i.e., the listening events) provide the most useful information about users' taste on music. Hence, we build a user-to-track graph (**u2t**) as the baseline network.

In our study, to analyze the impact of hashtags, we further add the connections between tracks and hashtags to the baseline network. Although there are several other ways to construct the graph, such as 'u2t2h' (i.e., no direct connection between users and hashtags), 'u2h2t' (i.e., no direct connection between users and tracks), 't2u2h' (i.e., no direct connection between tracks and hashtags), and 'uth' (i.e., allowing connections among users, tracks and hashtags), we select **u2t2h** out of the other four because in this way, the sampled random walks will always visit a track every two steps, as demonstrated in Figure 7.2. According to our observations, placing the tracks at the center of the modeling process in this way obtains better representations. Consequently, we employ the following two input graphs for computing the DeepWalk latent features:

- **u2t**: This represents the user-to-track bipartite graph, the relations of which are determined by whether a user has listened to a track in previous listening events.
- **u2t2h**: The user-to-track-to-hashtag graph that further considers the links between a track and its hashtags.

7.4.3. Ranking

The goal of ranking is to list the most suitable items (tracks in this study) on top. It is therefore a crucial task not only in recommender systems [8], but also more generally in the area of information retrieval [343] as it directly influences precision of recommendations or search results.

The main building blocks for computing a ranking for a set of recommendation candidates are users, tracks and hashtags that are extracted from the graph. The employed network embedding technique allows us to represent users by the latent features computed for users. We refer to this representation as "*user*". To also explicitly incorporate the hashtags that a user has previously adopted into the user's representation, we propose to use the latent representations of the hashtags the user made use of, leading to the user representation "*usertags*". A user may also be represented by the average sentiment value assigned to these hashtags as a measure of the user's general sentiment, which is a scalar. We refer to this user representation as "*usersent*". Similarly, we may model a track by its latent representation in the graph ("*track*"), the latent representations of all the hashtags which have been used to tag the track ("*tracktags*"), or the average sentiment value assigned to these hashtags as the track's general sentiment ("*tracksent*"). Furthermore, we aim to exploit information about the hashtags which are used for the given input tweet by using the average latent representation of these hashtags, leading to the representation of a tweet ("*tweettags*"). Besides solely relying on latent features, we also propose to represent the input tweet as the sentiment value associated with the hashtags mentioned in the input tweet ("*tweetsent*").

Based on these building blocks, we propose the following methods for ranking a given set of tracks. In principle, these methods differ in the way users, tracks and hashtags are characterized.

- **user_track:** rank according to the similarity of the latent representations of a given user and track.

- **user_tracktags:** rank according to the average pairwise similarity of the latent representation of the user and the individual latent representations of hashtags annotating the given track.

- **usertags_track:** rank according to the average pairwise similarity of the latent representations of hashtags a user has made use of and the latent representation of the track.

- **usertags_tracktags:** rank according to the average pairwise similarity of the latent representations of the hashtags of a user and the latent representations of hashtags used for annotating the given track.

- **tweettags_track:** ranking computed based on the average pairwise similarity of the latent representations of the hashtags used in the given input tweet and the latent representation of the track to be ranked.

- **tweettags_tracktags:** rank according to the average pairwise similarity of latent representations of hashtags of the input tweet and hashtags annotating the track to be ranked.

- **tweetsent_tracksent:** rank according to sentiment score similarity by using the difference of the sentiment scores assigned to the input tweet and the sentiment scores assigned to the tracks to be sorted. If a tweet or track features more than a hashtag, we compute the average sentiment score assigned to the track and compute the difference between these as

  $$sim = abs(avg(sent(tweet)) - avg(sent(track))) \tag{7.5}$$

  where *sent* determines the set of sentiments assigned to a given tweet or track.

- **usersent_tracksent:** rank according to the sentiment score similarity between the average sentiment of the user's previously used hashtags and the sentiment values annotating the track.

We can use either the cosine similarity or the Euclidean distance to compute the similarity between two latent representations. This similarity score is subsequently used to actually rank the tracks in order of descending similarity.

7.5. Results

We conduct three experiments in our study. The first and fundamental experiment aims to verify that utilizing an embedding approach is beneficial in our setting and that embedding approaches allow to capture a user's general listening preferences. The second experiment (and for us, the central experiment) aims to extensively evaluate the performance of different ranking methods and hence, the impact of affective contextual information extracted from hashtags. The third experiment is targeted at complementing our view on sentiment-based ranking methods and investigates the performance of individual sentiment lexicon. We present the results below.

7.5.1. Experiment 1: Effectiveness of Latent Features

In the first experiment, we aim to show that incorporating latent features contributes to a better ranking, capturing the general listening preferences of users. We therefore base this evaluation on the POP_RND task and evaluate the performance of the user_track ranking method (similarity of latent features of users and tracks), where latent features are computed based on the user-to-track graph (u2t). Hence, we do not consider any hashtag or affective information in this first experiment. We compare this approach with the following baseline methods:

- A random ranking approach that randomly shuffles the items within the recommendation list;
- Ranking according to the tracks' popularity within our data set (i.e., the number of distinct users having listened to the track) [157, 370]. Picking random items or the most popular items are basic and simple baselines often used for dealing with the cold-start problem [370];
- An item-item-based collaborative filtering approach based on the k-nearest neighbors (kNN) [402]. We set the size of the neighborhood k to 30 and use cosine similarity to measure the similarity between items, following the suggestion of Sarwar *et al* [402]. Herlocker *et al.* have also found that generally, a neighborhood size of 20 to 50 seems reasonable for real-world settings [188].

There are a few parameters to be empirically decided for the DeepWalk algorithm for learning the latent features. In a preliminary study we found that the following setting works reasonably well: dimension of the latent representation, which controls the model complexity—64, number of walks and the walk length, which control the number of sampling pairs for the modeling stage—16 and 64 respectively, the window size, which determines the reachable vertices—4. We use this parameter setting throughout the following experiments, for both u2t and u2t2h.

The results of the conducted analysis are listed in Table 7.4. As can be seen, incorporating latent features increases the quality of the ranking compared to the baseline methods. The random baseline reaches an average MRR of 0.29 for both data sets, while ranking according to the popularity of tracks reaches a MRR of 0.73 (#NP560k data set) and

Ranking Method	**#NP560k**	**#NP90k**
Random	0.29 (0.26)	0.29 (0.26)
Most popular tracks	0.73 (0.32)	0.76 (0.30)
kNN	0.81 (0.33)	0.79 (0.35)
user_track (u2t embedding; cos.)	**0.92** (0.21)	**0.81** (0.34)
user_track (u2t embedding; eucl.)	0.83 (0.31)	0.68 (0.41)

Table 7.4.: The mean reciprocal rank (MRR) achieved by different ranking methods for POP_RND for both the #NP90k and #NP560k data sets (standard deviation in parenthesis).

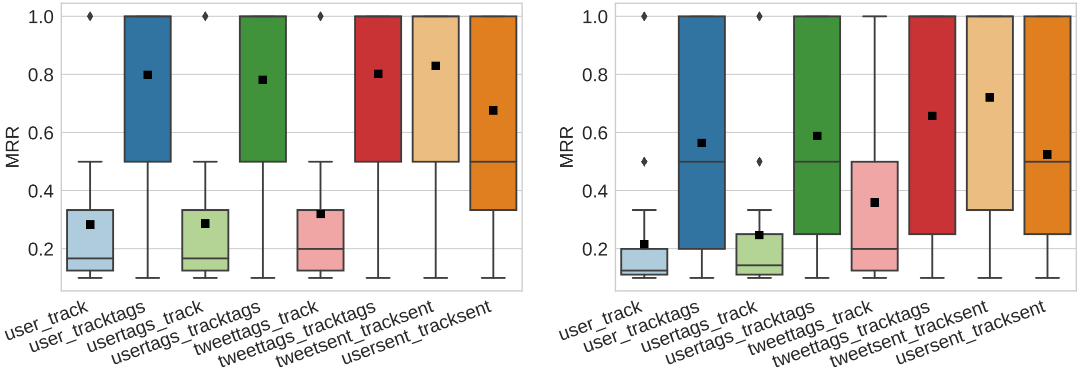
0.76 (#NP90k data set). Among the baselines, the item-item collaborative filtering baseline (kNN) reaches a MRR 0.81 (#NP560k data set) and 0.79 (#NP90k data set), respectively.

The use of latent representations of tracks and users increases the MRR to 0.92 for #NP560k and 0.81 for #NP90k. Also, cosine similarly outperforms euclidean similarity. Generally, from this first experiment we conclude that incorporating latent features in the ranking process yields improved results compared to the evaluated baseline approaches. Hence, this validates the effectiveness of the latent features for capturing a user's general musical preferences.

7.5.2. Experiment 2: Effectiveness of Affection and Hashtag Information

The goal of this experiment is to examine the benefit of incorporating hashtag and affective information into the ranking process. Ultimately, we aim to evaluate the performance of the individual proposed ranking strategies in a context-aware ranking task. Therefore, we consider both the POP_RND and POP_USER task in this experiment.

Table 7.5 depicts the results of this evaluation for the #NP560k data set and Table 7.6 presents the results for the #NP90k data set. As our experiments showed that cosine similarity consistently outperforms Euclidean similarity by a small margin, we only list the results of cosine similarity. For POP_RND, we see that the best results are obtained by the user_track ranking method, achieving a MRR of 0.92 (u2t embedding; #NP560k data set) and 0.83 (u2t2h embedding; #NP90k data set). For the #NP90k data set, usertags_track also reaches a MRR of 0.83. As for the user_track ranking method, we do not observe substantial differences between ranking approaches incorporating hashtags (i.e., u2t2h) and those not incorporating hashtags (i.e., u2t) in the latent features representation. As for the other ranking approaches, we observe that usersent_tracksent, user_tracktags, tweettags_tracktags and usertags_tracktags reach lower MRR values.



(a) #NP560k Data set

(b) #NP90k Data set

Figure 7.3.: Boxplot of MRR achieved by different ranking methods using POP_USER (u2t2h embedding; black square marks mean value across all evaluations).

Notably, the tweetsent_tracksent ranking method, which solely relies on the sentiment scores associated with the tracks to be ranked and hashtags the user made use of in the current input tweet, achieves 0.81 (#NP560k data set) and 0.70 (#NP90k data set).

In contrast, for POP_USER, we can see from Tables 7.5 and 7.6 that the sentiment ranking method tweetsent_tracksent outperforms all other methods, achieving the highest MRRs of 0.82 (#NP560k data set) and 0.71 (#NP90k data set), respectively. The results support our hypothesis that sentiment hashtags and embeddings incorporating hashtags allow for better capturing a user's context and hence, exploiting this information for ranking track candidates. For a better comparison, we also provide a boxplot of the MRR results for both data sets in Figure 7.3. The other sentiment-based ranking method, usersent_tracksent, achieves a MRR of 0.68 and 0.53, respectively. Notably, these methods do not use latent features. Methods utilizing "tracktags" for representing tracks, including user_tracktags, tweettags_tracktags, and usertags_tracktags, also perform well and reach a MRR around 0.80 for #NP560k and 0.60 for #NP90k. In contrast, the user_track method performs poorly here, with a MRR below 0.30 across all settings. In general, methods using "track" for representing tracks do not perform well. These findings suggest that contextual affective information and in general, information about the tags used to describe tweets or tracks is indeed exploited in this task. This is also signaled by the fact that methods that incorporate latent features of hashtags and sentiment information perform substantially better than the approach not incorporating any affective or hashtag information.

In sum, we argue that ranking tracks the user has already listened to is more challenging than ranking a set of randomly chosen tracks as these traditionally differ more. Therefore,

Ranking method	Graph	POP_RND	POP_USER
user_track	u2t	**0.92** (0.21)	0.28 (0.26)
user_track	u2t2h	0.91 (0.22)	0.29 (0.26)
user_tracktags	u2t2h	0.88 (0.26)	0.80 (0.32)
usertags_track	u2t2h	0.89 (0.24)	0.29 (0.26)
usertags_tracktags	u2t2h	0.84 (0.29)	0.78 (0.32)
tweettags_track	u2t2h	0.89 (0.23)	0.32 (0.29)
tweettags_tracktags	u2t2h	0.86 (0.27)	0.80 (0.31)
tweetsent_tracksent	—	0.81 (0.34)	**0.82** (0.30)
usersent_tracksent	—	0.39 (0.30)	0.68 (0.32)

Table 7.5.: The MRR achieved by different ranking methods, using cosine similarity for the #NP560k data set (standard deviation in parenthesis).

Ranking method	Graph	POP_RND	POP_USER
user_track	u2t	0.81 (0.34)	0.22 (0.22)
user_track	u2t2h	**0.83** (0.32)	0.22 (0.21)
user_tracktags	u2t2h	0.79 (0.33)	0.56 (0.37)
usertags_track	u2t2h	**0.83** (0.31)	0.25 (0.23)
usertags_tracktags	u2t2h	0.74 (0.34)	0.59 (0.37)
tweettags_track	u2t2h	0.84 (0.30)	0.36 (0.33)
tweettags_tracktags	u2t2h	0.78 (0.34)	0.65 (0.37)
tweetsent_tracksent	—	0.70 (0.37)	**0.71** (0.35)
usersent_tracksent	—	0.42 (0.32)	0.53 (0.32)

Table 7.6.: The MRR achieved by different ranking methods, using cosine similarity for the #NP90k data set (standard deviation in parenthesis).

we consider this result as promising. Our experiments also show that both data sets (and hence, splitting methods regarding training and test data) deliver robust and consistent results.

7.5.3. Experiment 3: Effectiveness of Individual Sentiment Lexica

In this experiment we aim to get a deeper understanding for the performance of different sentiment detection approaches or rather, lexica. Therefore, we now focus on the performance of the sentiment-aware ranking methods and firstly evaluate the performance of single sentiment lexica.

Dictionary	**Fallback**	**POP_RND**	**POP_USER**
AFINN	None	0.79 (0.34)	0.79 (0.34)
Opinion Lexicon	None	0.81 (0.32)	0.80 (0.33)
SentiStrength	None	0.85 (0.29)	0.85 (0.27)
Vader	None	0.87 (0.25)	0.85 (0.29)
AFINN	user_track	0.85 (0.29)	0.81 (0.31)
Opinion Lexicon	user_track	0.86 (0.28)	0.82 (0.31)
SentiStrength	user_track	0.86 (0.28)	0.84 (0.29)
Vader	user_track	0.89 (0.24)	0.85 (0.28)

Table 7.7.: Performance (in MRR) of different sentiment dictionaries for tweetsent_tracksent in the #NP560 data set (standard deviation in parenthesis).

The usage of sentiment dictionaries for the detection of sentiment in a text is naturally limited by the coverage of the given sentiment dictionary (cf. Section 7.4.1 regarding the coverage of the sentiment lexica used). This limited coverage consequently constrains the number of affective hashtags detectable using any single dictionary which further limits the number of tracks which can be actually assigned with a sentiment score. Thus, only a limited number of tracks can be compared in this regard.

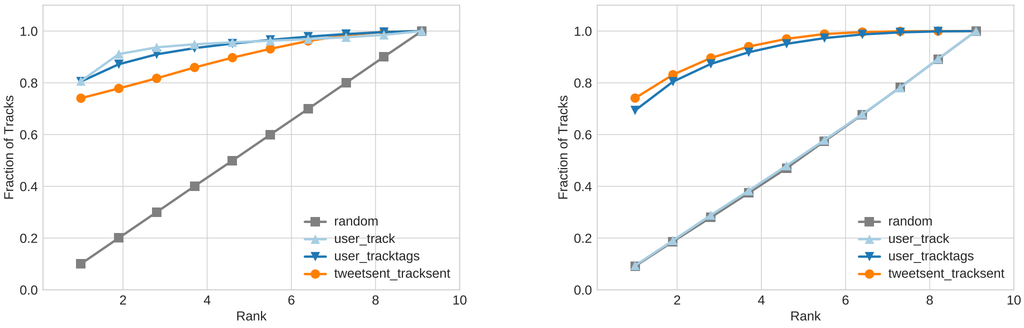
To compare the different lexica nonetheless, we propose the following method. For those tracks, users and tweets for which we can compute a sentiment score using the given dictionary, we rely on the best performing ranking method tweetsent_tracksent as evaluated in the previous experiments. However, for the remaining tracks, users and tweets with no sentiment scores, we employ a *fallback* method. Here we distinguish two cases: i) if we cannot detect a sentiment score for either the user or the tweet, we use the fallback method for all the tracks to be ranked; ii) if we cannot detect a sentiment for a track (or a set thereof), we compute the similarity of user (or tweet; depending on the ranking method) and the track using the fallback method. As for the fallback methods, we chose to use and evaluate the best-performing ranking methods not relying on affective information for each task. Hence, we evaluate user_track for POP_RND and tweettags_tracktags for POP_USER as fallback methods, respectively and utilize the average sentiment score detected for a given tweet or track for the comparison.

Tables 7.7 and 7.8 show the results for the #NP560 and #NP90k data set, respectively. Here, we consider POP_RND as a special case as the best performing method is not sentiment-based and the user_track fallback method performs better than the sentiment-aware ranking methods. Hence, the usage of such a fallback method naturally increases the performance of the evaluation where the degree of improvement depends on the coverage of the dictionary used. However, the goal of this evaluation is to evaluate the individual dictionaries and therefore, we still list the results. To provide a complete pic-

Dictionary	Fallback	POP_RND	POP_USER
AFINN	None	0.68 (0.39)	0.68 (0.37)
Opinion Lex.	None	0.71 (0.38)	0.69 (0.37)
SentiStrength	None	0.72 (0.36)	0.73 (0.35)
Vader	None	0.77 (0.33)	0.77(0.34)
AFINN	tweettags_tracktags	0.77 (0.35)	0.68 (0.36)
Opinion Lex.	tweettags_tracktags	0.79 (0.33)	0.68 (0.36)
SentiStrength	tweettags_tracktags	0.76 (0.35)	0.73 (0.34)
Vader	tweettags_tracktags	0.80 (0.22)	0.74 (0.34)

Table 7.8.: Performance (in MRR) of different sentiment dictionaries for tweetsent_tracksent in the #NP90 data set (standard deviation in parenthesis).

ture of the results, we also list the performance of the individual sentiment dictionaries when no fallback method is used (i.e., 'Fallback None'). As the table shows, the best results (by a slight margin) are obtained using the user_track fallback method. Examining the dictionaries used, we do observe slight differences but note that Vader performs the best. As for POP_USER, we observe that in this case, using no fallback method performs slightly better than using the fallback method as our experiments in Section 7.5.2 already showed that tweetsent_tracksent is the best performing ranking strategy (again, by a moderate margin). As for the individual dictionaries, we find that the differences in regards to the MRR are rather moderate with Vader again performing the best for both the user- and the tweet-based sentiment ranking.



(a) POP_RND

(b) POP_USER

Figure 7.4.: Cumulative ranking distribution of different methods for the #NP560k data set.

7.6. Discussion

We further discuss the evaluation results in this section.

In the first experiment we showed that representing tweets, tracks and users by latent features computed by the DeepWalk algorithm and using similarities between these for ranking tracks achieves comparable results as traditional ranking methods. Therefore, we conclude that these latent representations are able to capture users' general listening preferences and that those can be used for ranking tracks in a recommendation or retrieval scenario. Our experiment comparing the performance of user_track learned from u2t and u2t2h sees marginal differences in terms of MRR, for either POP_RND or POP_USER. In this scenario, these findings signal that hashtag information integrated in the computation of latent features does hardly influence the resulting latent feature representations for users and tracks.

However, using u2t2h as the underlying graph permits learning the latent feature representations for hashtags, which is useful for the POP_USER task. That is, for the context-aware ranking task, hashtags (providing contextual information) naturally contribute to an improved ranking. Analyzing the results of the different proposed ranking methods in our second experiment, we find that using the latent representations of hashtags that are used to tag tracks (i.e. "tracktags") seem to be more representative of a track than using solely the latent representation of the track itself (i.e. "track") for POP_USER. We can observe that tracktags performs substantially better over all configurations. However, this does not hold for POP_RND. These findings suggest that for POP_RND, the latent representation of a track seems more suitable than using the hashtags annotating a track. This shows that for capturing a user's general listening preferences, utilizing the latent representations of users and tracks are sufficient for computing a suitable ranking.

Similarly, users can either be represented by the user's latent feature representation ("user") or by the latent representations of the hashtags the user made use of ("usertags"). However, we encounter mild differences between the performance of these two representations for either POP_RND or POP_USER. Hence, we conclude that the differences of different representations for users are hardly distinctive.

Among the two sentiment-based ranking methods, we find that using the sentiment of the input tweet ("tweetsent") performs better for both POP_RND and POP_USER. These results suggest that using the sentiment expressed by the user in the current tweet captures the current affective context better than using the average sentiment a user has previously expressed through hashtags. This can also be seen in Figure 7.4, which plots the cumulative ranking function for the random baseline and the ranking methods user_track, user_tracktags, tweetsent_tracksent (utilizing the average score across all sentiment lexica). For POP_RND we observe that user_track provides superior results across all ranks incorporated. In contrast, for POP_USER we observe that user_track shows behavior highly similar to the random ranking approach (those two lines actually overlap heavily), whereas tweetsent_tracksent and user_tracktags perform substantially better across all ranks.

From these experiments we observe that choosing a suitable representation for tracks, users and tweets is crucial for the quality of the ranking. We find that for POP_RND, comparing the latent representations of users and tracks is sufficient to provide high-quality ranking of tracks. However, the POP_USER experiment showed that this does not suffice when the ranking task gets more personal and complex. This experiment showed that ranking based on contextual affective information performs best. Particularly, the tweetsent_tracksent ranking method outperformed the other methods. From these findings we conclude that while for the POP_RND task the latent representations did capture the user's preference well, for the POP_USER task the sentiment did capture the user's musical interest better.

The third experiment aimed to evaluate the performance of the individual sentiment lexica and hence, their suitability for this task. We observed that Vader performed best across all evaluations. However, we have to note that the differences are rather moderate. Given that Vader performs similar to the other dictionaries in terms of coverage, we lead this back to the fact that Vader is a particularly geared towards social media texts.

Our evaluation design proposes a fallback method to compensate for those tweets, users and tracks which could not be assigned with a sentiment score using the given dictionary. This naturally implies that the choice of the fallback ranking method is vital. We propose to evaluate the best performing algorithm not considering sentiment data. For POP_RND, the fallback methods individually perform better than the sentiment-based ranking methods. Thus, an improvement of the results when introducing user_track is an obvious result. The tweettags_tracktags ranking method is also able to improve the results, though to a lower degree. As already laid out, we consider the POP_USER task as the more difficult and personal task. For this evaluation, results worsened by the fallback methods as expected, since these methods did not perform as well as the sentiment-based methods in the previous experiments. From these results we reason that implementing a fallback method is a good choice as it provides means for compensating the lack of coverage. Also, we conclude that choosing the fallback method according to the complexity and degree of how personal the ranking task is, seems plausible. As for the choice of sentiment dictionaries, we propose to employ the union of multiple dictionaries to increase coverage. While our experiments show that minor improvements for single dictionaries, we argue that in this case, coverage should be prioritized as it allows for a higher applicability of the sentiment-based ranking, which has shown to perform better.

7.7. Background and Related Work

Before concluding the paper, we give a brief review of related work in psychology and recommender systems, to put this work in the context of the literature.

7.7.1. Psychological Studies on Emotion Regulation

Emotion regulation is important for the performance and well-being of mankind [171]. It is widely accepted that emotions play a major role in driving our decisions. Beneficial emotion regulation strategies help people to stay calm under stress, handle failures in a mindful and positive way, etc. Due to its importance, emotion regulation has been recognized as one of the fastest growing areas within the field of psychology [257, 453].

Emotion regulation has also been identified as an essential reason for musical engagement [179, 227, 472]. Boer and Fischer [54] found that emotion regulation represents the most important personal use of music across human subjects from four cultural backgrounds. Goethem and Sloboda [472] found that music listening is the second-most used tactic for emotion regulation, just behind "talking with friends".[4]

Saarikallio *et al.* [392] found that a person's general tendency to emotionally appreciate, enjoy and react to music (i.e., *emotional reactivity to music*) is positively correlated with the tendency to use music for emotion regulation (i.e., *emotional use of music*). Being familiar with a music piece increases a person's emotional use of that piece in daily life [392]. Moreover, informal engagement through listening, but not formal musical training, correlate with heightened emotional use of music. Saarikallio also argued that music should not simply be considered as one emotion regulation mechanism, but rather as a tool for realizing several different emotion regulation strategies, including positive mood maintenance, relaxation and revival, induction of strong emotions, diverting away from worries, discharging negative emotion, mentally working through emotion preoccupations, and finding solace and understanding [393]. Individual differences in the use of these strategies have also been noted: e.g., some prefer emotional reinforcement of current experiences, while others prefer to distract themselves and change emotions [391, 452].

While many psychological studies were conducted in the lab with small to medium sample size, what we investigate here is the relationship between a user's self-report emotional state and the self-report musical preference through Twitter at scale and "in the field." Moreover, a computational approach that investigates how to represent the affective information of users and music using machine learning and sentiment detection techniques is taken. Although our study may also lead to psychological insights, the focus is more on the engineering side, targeting at applications such as affective music recommendation.

[4] The other tactics considered in their study include "exercising", "reading a book/magazine", "watching TV/movie", among others [472].

There have been psychological evidences showing that the emotional state of users affects musical preference. For example, depressed patients expressed an intensified response to sad-sounding music when compared to healthy controls [53]. Moreover, such patients evaluated negative-valence music as significantly more sad and angry than healthy controls did [366]. However, according to Gross [171], people do not always attempt to stay away from negative emotions. Reasons for up-regulating negative emotions include promoting a focused, analytic mindset; fostering an emphatic stance; and influencing others' actions.

An interesting research direction is therefore to use Twitter data to computationally study the effect of music in emotion regulation at scale using a longitudinal approach. This requires tracking specific users' #nowplaying tweets and emotional states over time, which to our knowledge has not been attempted before. We leave this as a future work.

Finally, we remark that Hargreaves and North [179] proposed that music has three types of psychological functions: cognitive, emotional, and social functions. The focus of this paper is on the emotional functions of music, neglecting the possible social functions manifested in the Twitter data.

7.7.2. Affective Multimedia Recommendation

Contextual factors relevant to music recommendation may include the time, location and device of music listening, user's present emotional state and activity, etc. [243]. While it is relatively easier to infer some of these factors from sensors such as clocks, GPS and accelerometers [411, 481, 494], accessing the emotional state of a user is more difficult. As users may not always be willing to report their emotions, computational methods for user emotion prediction from facial expressions, prosody cues, text, and physiological signals have been widely studied [72, 288, 514].

With 40K blog posts collected from the social blogging website LiveJournal[5], Yang and Liu [493] investigated the relationship between the emotional state of a user and the emotion of preferred music pieces. Similar to the Twitter data set we use in this paper, the LiveJournal data set they employed also contains the self-report emotional states and self-report preferred music pieces [290]. Yang and Liu [493] used audio signal processing and machine learning techniques to recognize the emotion of the music [491] and then correlated the emotion in music with the emotional state of the users, finding that users do prefer music of different emotions in different emotional states. Following this work, Chen *et al.* [88] showed that considering the emotional state of a user indeed improves the quality of music recommendation, comparing to conventional collaborative filtering approaches that do not use affective information. This article extends from these two prior articles in that we use a larger data set and more sentiment detection methods.

[5] http://www.livejournal.com

Ferwerda and Schedl [146] proposed the idea of exploiting both the personality and emotional state of a user for music recommendation, but did not actually implement such a system. Rosa *et al.* [387] built a system that recommends music according to the emotional states inferred from user-generated text by sentiment detection, but the system was evaluated using a small-scale data set collected from a crowdsourcing platform, not from social media websites. Deng *et al.* [117] assumed that the emotional state of a user can be determined by the emotions of the music pieces the user just listened to. There are some other affective music recommender systems proposed in the literature, but many of them require users to indicate their present emotional states or the desired emotions of the music [24, 37, 178].

Affective movie recommendation has also been studied in recent years, using mainly the users' self-report emotional states [434, 464, 522]. Although it might be possible to crawl movie preference data from social platforms such as Twitter, few attempts have been made thus far.

7.8. Conclusion and Future Work

In this paper, we have proposed a set of novel methods for ranking music recommendation candidates. In particular, we proposed to represent the building blocks (users, tracks, affective hashtags) by their latent features computed by a network embedding algorithm called DeepWalk. Based on these latent feature representations, we proposed a number of ranking methods. Furthermore, we proposed two ranking methods that are solely based on sentiment scores. Our evaluation using #nowplaying tweets showed that the use of latent features to represent users, tracks and hashtags contributes to better ranking. The evaluation procedure distinguished two tasks of increasing complexity: i) ranking a set of randomly picked tracks and ii) ranking a set of tracks the target user has already listened to. We find that for the first task, comparing the latent representations of users and tracks (regardless of used hashtags or affective information) performs best and this confirms our hypothesis that applying an embedding technique effectively captures the general listening preferences of users. However, for the second, context-aware ranking task, using solely affective information extracted from hashtags leads to the best result. This shows that the more personal and complex a ranking task gets, the higher the influence and significance of affective and hence, context information is. Finally, an evaluation of the different sentiment lexica showed that the differences in performance of the individual lexica is rather moderate. While Vader achieves the best results, we argue that combining several dictionaries or implementing fallback methods results in a more robust approach.

Future work includes incorporating more sophisticated sentiment detection approaches both regarding the underlying dictionaries as well as the computation of the sentiment scores. In a first step, we aim to further evaluate different aggregation methods for tracks that are tagged with multiple tags with divergent sentiment scores (e.g., #happysad). Also, we aim to experiment with probabilistic models for representing a user's or track's sentiment values (e.g., using Gaussian Mixture Models [479]).

Modeling affective information in a multidimensional model such as the valence-arousal space [480, 491] is worth exploring. Also, we aim to extend the unsupervised sentiment-detection approach (based on sentiment dictionaries) to a supervised learning approach that permits cross-lingual sentiment detection [29]. Lastly, the computation of latent features based on the proposed graph needs to be investigated in more detail. Particularly, we aim to investigate the influence and performance of different embedding strategies for the computation of latent representations. Ultimately, we intend the development and evaluation of real-world applications for music recommendation and music-based emotion regulation based on our findings.

Acknowledgements

The computational results presented have been achieved (in part) using the HPC infrastructure LEO of the University of Innsbruck. The research of Chen, Tsai and Yang is supported by a grant from the Ministry of Science and Technology, Taiwan, under project MOST 106-3114-E-002-007.

8. User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues

Publication

E. Zangerle, M. Pichl, and M. Schedl. User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues. *Transactions of the International Society for Music Information Retrieval*, 3(1):1–16, 2020. DOI: 10.5334/tismir.37

Abstract

Integrating information about the listener's cultural background when building music recommender systems has recently been identified as a means to improve recommendation quality. In this article, we, therefore, propose a novel approach to jointly model users by their *musical preferences* and *cultural backgrounds*. We describe the musical preferences of users by the acoustic features of the songs the users have listened to and characterize the cultural background of users by culture-related socio-economic features that we infer from the user's country. To evaluate the impact of the proposed user model on recommendation quality, we integrate the model into a *culture-aware recommender system*. By analyzing a dataset comprising approximately 400 million listening events of about 55,000 users from 36 countries, we show that incorporating both acoustic information of the tracks a user has listened to as well as the cultural background of users in the form of a *music-cultural user model* contributes to improved recommendation performance. Furthermore, we provide a systematic analysis of the influence of different features on the quality of the provided culture-aware track recommendations. We find that considering acoustic features that model the characteristics of tracks and a user's musical preferences have the highest impact on recommendation performance. However, adding socio-economic features allows further improving the recommendation quality. In addition, we identify interesting correlations between acoustic characteristics of music preferences and cultural features of populations at the country level.

8.1. Introduction

Recent advances in recommender systems and music information retrieval have shown that contextual information is vital for highly personalized results (e.g., [62, 356, 481]). In this scope, context can be defined as "conditions or circumstances which affect some thing" [11, 232], where, e.g., environment-related contextual information may include location, time or weather [231]. Consequently, the user's listening context can be defined as the user's context during listening to music. To this end, the geographic location of a user is often exploited as one basic notion of context. Leveraging GPS coordinates to model *similarity* between listeners, which is key to build recommender systems, results in location-aware systems, which are however agnostic to cultural characteristics and the cultural background of users. In the scope of this article, we define the *cultural background* of users as a set of attributes that allow for describing the culture the user is embedded in, including social or economic aspects, as well as, e.g., cultural practices, values, and behavior. However, location alone does not necessarily serve as a good indicator for the cultural background of a user as geographically close users might have a very different cultural background. A user's cultural background may also not coincide with political borders [363]. Notably, the cultural background of a user was identified already in [421] as a possibly relevant aspect to improve recommender systems. We hence argue that modeling users based on musical properties of the songs they listen to (approximating their musical preference) on the one hand and the user's cultural background on the other contributes to capturing *music-cultural listening patterns.* These patterns particularly describe the complex interrelation between users, their cultural background, and the characteristics of the music they listen to. In this article, we propose a novel *music-cultural user modeling* approach to exploit such listening patterns for recommender systems by integrating information about (i) the acoustic qualities of the music users have listened to and (ii) culture-specific information derived from the users' location/country to describe the user's likely cultural background.

Leveraging a standardized collection of almost one billion user-generated listening events, we evaluate the proposed user model.[1] By exploiting music-cultural listening patterns captured by the proposed user model in a recommender system, we show that the resulting culture-aware music recommendations are more accurate than those provided by a recommender agnostic to cultural information. Particularly, we find that capturing a user's individual music taste by the high-level audio features of the tracks the user has listened to and adding Hofstede's cultural dimensions [192] as well as data from the World Happiness Report (WHR) [187] as a description of the cultural (and socio-economic) background of the user provides the best recommendation results, in terms of accuarcy and error measures.

The remainder of the article is organized as follows. Section 8.2 briefly reviews related work on context- and culture-aware music recommendation. The dataset we use, a processed version of the LFM-1b dataset [408], is presented in Section 8.3. Section 8.4

[1] A listening event is defined as a quintuple <user, artist, album, track, timestamp>.

provides details on (i) our methods for user modeling according to musical preferences and cultural aspects, and (ii) our proposed culture-aware recommender system. The experiments we conducted to evaluate the user models and recommender system approaches are explained in Section 8.5. We present and discuss the results obtained in Section 8.6. To gain more insights into the overall and country-specific patterns of acoustic music preferences, Section 8.7 presents results of an additional study on differences in acoustic preferences between countries and on correlations between cultural and musical features. The paper is rounded off by a summary and outlook to follow-up research in Section 8.8.

8.2. Related Work

In music recommender systems, unlike for instance in movie recommendation, content-based approaches have been the dominant focus of research for a long time [244]. Music content is, in this case, either incorporated into the recommendation algorithm in the form of hand-crafted acoustic features or—more recently—by automatic feature extraction from the raw audio signal using deep neural networks. Examples of the former include a rich set of features that have been proposed in the past two decades of music information retrieval research, and ranges from Mel frequency cepstral coefficients (MFCCs), e.g., [296], to semantic descriptors of acoustic properties, e.g., [321, 467]. For an overview, consider, for instance, [75, 242]. Deep learning-based approaches to automatic feature learning for content-based music recommendation include convolutional neural networks (CNN) and recurrent neural networks (RNN), in particular its variants long short-time memory (LSTM) and gated recurrent units (GRU). For a more detailed review of deep learning approaches in music recommendation, please consider Schedl [405].

Nowadays, it has become widely accepted that incorporating contextual information into recommender systems contributes to improved recommendations [11]. Particularly for music recommender systems, studies showed that users often seek for music that matches their current situation, and hence context (i.e., occasion, event or emotional states) [241, 275]. In the scope of music recommender systems, [232] distinguish environment-related context (location, time or weather), user-related context (activity, demographic information or emotional state of the user), and multimedia context (text or pictures the user is currently reading or looking at). For our study, the environment-related context of a user is of particular relevance as we aim to leverage both the musical preferences and cultural background of users for improving track recommendations.

[420] performed a study on the contribution of geospatial information to the performance of artist recommender systems. They conclude that if users listen to various different artists, the integration of geospatial information is beneficial. In [422], the authors approximate the cultural distance of users by the country or continents a user is located in and show that this is beneficial for users particularly in the U.S. and Russia.

Furthermore, there are several approaches that exploit places of interest as contextual information, where the idea is to recommend music that suits the environment—in an emotional or cultural sense [63, 234]. Rich sensory devices such as smart phones allow mapping a certain location to a certain activity that can be exploited for personalized location-based music recommendations, depending on the user's inferred activity [481]. [33] propose a context-aware music recommender system for car drivers, where a set of diverse contextual factors are incorporated (e.g., driving style, traffic conditions, weather or road type). [27] propose the Foxtrot system, which allows users to tag music with geolocations. Based on this information, users can be provided with location-specific music recommendations. [96] model the listener's short-term music needs, location, and overall popularity to create personalized music recommendations. [198] propose a music recommender system that integrates track genre, release year, freshness, and temporal aspects.

As for cultural aspects in the broader field of music information retrieval, [147] found that a user's cultural background (modeled by Hofstede's cultural dimensions [192]) influences how diverse the musical preferences of users are. Particularly, they found that highly individualist countries and countries that are flexible, pragmatic, and eager to adapt to changes listen to more diverse genres. [419] also performed a study on whether cultural similarity between countries (described by Hofstede's cultural dimensions and the Quality of Government (QoG) dataset) is reflected in music taste (described by tags annotating music tracks). They found medium correlations of music taste and several cultural and socio-economic factors. Notably, this evaluation is based on the LFM-1b dataset, which is also utilized in the experiments conducted in this study. Furthermore, Liu et al. have uncovered similarities between countries based on cultural and socio-economic aspects on the artist level and on the album level [291, 292].

[363] clustered users based on their individual musical preferences and their cultural characteristics. Relying on density-based spatial clustering, they find nine clusters that describe similar users regarding both their musical preference and cultural background. The cultural background of users was described by the World Happiness Report [187] and the authors found that incorporating cultural information allows for more precise user descriptions compared to relying on geographic information only. However, this evaluation did not target recommender systems and was done on a substantially smaller dataset.

We are not aware of any work exploiting the cultural background of users for the computation of context-aware music recommendations and hence locate a research gap here. In this paper, we show that utilizing the cultural background of users together with their general musical preference contributes to improved recommendation quality.

8.3. Data

In this section, we present the data utilized for performing our analyses and experiments.

Item	Value
Listening events	394,944,868
Users	55,149
Tracks distinct	3,478,399
Min. LE per user	1
Q_1 LE per user	1,442
Median LE per user	5,667
Q_3 LE per user	9,738
Max. LE per user	399,210
Avg. LE per User	7,161.41 (± 10,326.91)
Avg. Users per Country	1,155.93 (± 1,894.96)

Table 8.1.: Statistics of the dataset utilized (LE=listening event).

For our analyses, we require a dataset that contains a substantial number of listening histories of users as well as country information about these users. There are indeed a number of datasets containing listening histories: The Million Musical Tweets Dataset [183] and the MusicMicro dataset [407] come with contextual information related to time and location. The musical listening histories dataset [475], the Yahoo! Music ratings dataset [127] and the #nowplaying dataset [509] contain a substantial number of users, items also including timestamps of LEs; however, no contextual information regarding the user's country is given. Hence, we base our investigations on the LFM-1b dataset [408], which contains more than one billion listening events created by users of the online music platform Last.fm,[2] where music listeners can share information about their listening behavior. The LFM-1b dataset has been created in the following way using various endpoints of the Last.fm API [406]: First, the top artists labeled by any of the 250 top user-generated tags used on Last.fm were retrieved. Then, the top fans of these artists were fetched, resulting in about 465,000 users. Listening histories (i.e., each user's set of listening events) of a randomly chosen subset of 120,322 users were subsequently downloaded. The creation time of the listening events cover the time span between January 2005 and August 2014.

Since we aim to model music-cultural preferences jointly by individual musical preference and the cultural background of users, we require the data to contain information about the location of the user. For 45.87% of all users within the LFM-1b dataset, country information about the user is available. Therefore, we restrain the dataset to those users

[2] https://www.last.fm

(and their tracks) for whom we are able to obtain country information. This provides us with a dataset comprising 55,191 users, who have listened to a total of 26,022,625 distinct tracks, which are captured by a total of 807,890,921 listening events.

Besides the information contained in the LFM-1b dataset, we also require information about the tracks the users listened to (cf. Section 8.4.1). Particularly, we are interested in content features that are able to describe a given track. Therefore, we rely on the Spotify API to gather content-based audio features, as described in Section 8.4.1, for each track. For all listening events of users for whom we can obtain country information, we search for the <track, artist, album> triples extracted from the LFM-1b dataset using the Spotify search API[3] to gather the Spotify URI of each track (i.e., we provide all three parts in a conjunctive query). This URI is subsequently used to query the audio features API,[4] which returns the set of audio features describing the contents of a given track (cf. Section 8.4.1), which allowed gathering 4,326,809 Spotify URIs. For the remainder of the tracks, the Spotify API is not able to correctly resolve the triples to a track. We attribute this to two factors: either the searched track is not provided by Spotify or the track, artist, and album information cannot be matched to a Spotify track URI unambiguously. Also, the Spotify API does not provide all features for all tracks and hence, we remove those tracks for which the API does not provide a full set of audio features from the dataset. Employing this procedure, we are able to acquire the full set of audio features for a total of 3,478,399 tracks. Notably, these 13.36% of the distinct tracks for which we can obtain audio features are able to capture 48.89% of all listening events (i.e., the tracks listened to by users).

The remaining tracks and respective listening events are excluded from the dataset. This eventually results in a dataset of 55,149 users, 394,944,868 listening events and 3,478,399 distinct tracks. Table 8.1 depicts the main characteristics of the dataset underlying our analyses.[5] As can be seen, the average number of listening events per user is 7,161, which we consider a substantial number that is able to capture a user's individual musical preferences well. Furthermore, the average number of users per country is 1,156. Along the lines of [147], we restrain the dataset to countries with more than 200 users to ensure that countries are well-characterized and results are valid and representative (at least of a typical music streaming community such as the one at Last.fm). Table 8.2 depicts the number of users per country for all countries with more than 200 users within our dataset. In total, the cleaned dataset features users from 36 different countries. Note that countries in this article are abbreviated using their ISO 3166 2-digit country code.[6]

[3] https://developer.spotify.com/web-api/search-item/

[4] https://developer.spotify.com/web-api/get-several-audio-features/

[5] To foster further research, we provide the dataset at https://doi.org/10.5281/zenodo.3477842.

[6] https://www.iso.org/iso-3166-country-codes.html

Abbrv.	Country	Users
US	United States	10,251
RU	Russian Federation	5,021
DE	Germany	4,576
UK	United Kingdom	4,533
PL	Poland	4,403
BR	Brazil	3,882
FI	Finland	1,409
NL	Netherlands	1,375
ES	Spain	1,242
SE	Sweden	1,230
UA	Ukraine	1,140
CA	Canada	1,077
FR	France	1,055
AU	Australia	976
IT	Italy	973
JP	Japan	798
NO	Norway	750
MX	Mexico	705
CZ	Czechia	632
BY	Belarus	558
BE	Belgium	513
ID	Indonesia	484
TR	Turkey	478
CL	Chile	425
HR	Croatia	372
PT	Portugal	291
AR	Argentina	282
CH	Switzerland	277
AT	Austria	276
HU	Hungary	272
DK	Denmark	271
RS	Serbia	253
RO	Romania	237
BG	Bulgaria	236
IE	Ireland	219
LT	Lithuania	202

Table 8.2.: Number of users per country for countries with more than 200 users. We use ISO 3166 2-digit country codes to abbreviate country names.

8.4. Methods

In the following, we detail the proposed approach for leveraging individual and cultural listening patterns for the computation of track recommendations based on the underlying dataset (as described in Section 8.3). We first present our user modeling approach (for individual and cultural listening patterns) and secondly present the proposed music-cultural user model. Subsequently, we show how we leverage this model for the computation of track recommendations.

8.4.1. User Modeling: Musical Preferences

As for modeling individual musical preferences, we gather content-based audio features for each of the tracks in the dataset by querying the Spotify API[7]—following the lines of, e.g., [21, 314, 362]. We make use of these Spotify high-level features for a number of reasons: First, the LFM-1b dataset does not contain audio data that we could use to extract audio features from. Second, our analyses aim at investigating the general suitability of merging acoustic and cultural cues for music recommendation rather than low-level feature engineering and hence, we rely on Spotify's audio features as a compact characterization of tracks. These content features are extracted from the audio signal of a track and comprise:

1. *Danceability* describes how suitable a track is for dancing and is based "on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity."

2. *Energy* measures the perceived intensity and activity of a track. This feature is based on the dynamic range, perceived loudness, timbre, onset rate and general entropy of a track.

3. *Speechiness* detects presence of spoken words in a track. High speechiness values indicate a high degree of spoken words (talk shows, audio book, etc.), whereas medium to high values indicate e.g., rap music.

4. *Acousticness* measures the probability that the given track only contains acoustic instruments.

5. *Instrumentalness* measures the probability that a track contains no vocals (i.e., instrumental).

6. *Tempo* quantifies the rate of the beat in beats per minute.

[7] A description of these features and the API can be found at https://developer.spotify.com/web-api/get-several-audio-features/.

7. *Valence* measures the "emotional positiveness" conveyed by a track (i.e., cheerful and euphoric tracks reach high valence values).

8. *Liveness* captures the probability that the track was performed live (i.e., whether an audience is present in the recording).

8.4.2. User Modeling: Cultural Aspects

As for the cultural dimension, we propose to model cultural aspects on a country level and make use of two different resources: Hofstede's cultural dimensions [192, 194][8] and the World Happiness Report[9] of 2016 [187], which we describe in the following.

A widely accepted instrument to describe cultures is Hofstede's cultural dimensions (HOF). This framework describes a nation's culture and values by the following six dimensions:

1. *Power distance* (PD) is defined as "the extent to which the less powerful members of organizations and institutions (like the family) accept and expect that power is distributed unequally" [187].

2. *Individualism* (IDV) captures the extent to which people are integrated into groups. Societies with high scores possess only loose ties and the individual is considered more important than the collective group.

3. *Masculinity* (MAS) assesses a preference in society for achievement, heroism, assertiveness and material rewards for success. Low masculinity (femininity) signals a preference for cooperation, modesty, caring for the weak and quality of life.

4. *Uncertainty avoidance* (UA) measures to which degree members of a society tolerate ambiguity. Countries with a high score tend to rely on stiff codes, guidelines, and laws. In contrast, lower scoring countries show more tolerance and acceptance of differing thoughts.

5. *Long-term orientation* (LTO) measures the connection of the past with current and future actions or challenges. Low-scoring societies tend to keep traditions and norms and are suspicious of societal change, while high-scoring societies encourage thrift and adaption.

[8] https://www.hofstede-insights.com/models/national-culture/
[9] http://worldhappiness.report/

6. *Indulgence* (IND) captures the happiness of a country and "relatively free gratification of basic and natural human drives related to enjoying life and having fun". In countries with low indulgence scores, gratification of needs is suppressed and regulated by strict social norms.

In addition to Hofstede's cultural dimensions, we complement our model with socio-economic characteristics of countries. We capture these by figures extracted from the World Happiness Report (WHR) [187]. [424] showed that cultural factors are directly influenced by the subjective well-being of people. Therefore, we rely on the WHR as it captures people's cognitive and affective evaluations of their daily life and thus, their subjective well-being [121] on a country level. The WHR provides the following set of measures capturing the perceived happiness of countries:

1. *Freedom* measures the perceived freedom to make life choices.

2. *Healthy life expectancy* captures the healthy life expectancy at birth in a given country.

3. *Generosity* specifies whether people in a country are willing to spend money on a charity.

4. *Social support* states if people have people helping them if they need support (i.e., relatives or friends).

5. *Trust* measures the publicly perceived absence of corruption in government and business.

6. *Happiness* quantifies the subjective and perceived happiness.

7. *GDP* is the real gross domestic product per capita.

8.4.3. Music-Cultural User Model

Based on the features we leverage to capture a user's musical preferences (Section 8.4.1) and a user's cultural background (Section 8.4.2), we propose the following music-cultural user model for computing culture-aware recommendations.

Generally, we characterize a user's individual musical preferences and cultural background in a single feature vector. As for capturing a user's *individual musical preferences* based on the tracks listened to, we leverage the audio features of tracks as presented in Section 8.4.1. Except for tempo, all of these features are given in the range of $[0, 1]$. For tempo, we apply a linear min-max scaling to also represent it in the range of $[0, 1]$. To exclude tracks with audio features that distort a user's aggregated musical features, we remove outlier tracks from the user's listening history by applying the median absolute deviation (MAD) outlier detection method [282]. We consider a feature value an outlier if it is not within $M \pm a \cdot MAD$, where M is the median of this particular feature across all tracks of a user and MAD is the median absolute deviation of these values. As for the choice of a, we set a strongly conservative threshold $a = 3$ as proposed by [282]. Hence, a value is considered an outlier if it is not within three MADs around the median. Lastly, a track is considered as an outlier in the list of tracks of a particular user if one of its features is considered an outlier and consequently removed from the user listening history. For each of the features, we compute the average feature value and the standard deviation across all tracks in the user's listening history and add these average and standard deviation (SD) values to the user's feature vector. We chose to add the standard deviation of each of these features to mitigate the effects of averaging a large number of features that potentially differ substantially.

For the approximation of the *cultural background* of users (or rather, the country they live in) by socio-economic aspects, we rely on the variables of Hofstede's cultural dimensions and the World Happiness Report and extract these based on the user's country information. We add these variables to the feature vector to find *cultural* listening patterns that reflect cultural similarity better than the geographic distance. For each of these variables, we perform a linear min-max scaling such that all elements of the vectors are within $[0, 1]$ and concatenate it with the user vector.

8.4.4. Recommendation Computation

We model the computation of context-aware music recommendations based on the proposed user model as a learning task for rating prediction, where we aim to learn the probability P that a given user u has listened to a given track t. To learn these probabilities $P(u, t)$ for all users and tracks, we rely on Gradient Boosting Decision Trees. Particularly, we utilize the popular XGBoost system [93], a scalable end-to-end tree boosting approach which has been shown highly suited for recommendation tasks [31, 340, 465]. Using XGBoost, we set the learning objective to logistic regression for binary classification, which provides us with the desired probabilities. For the training phase, we set the training objective to be the binary classification error rate (i.e., the number of wrongly classified tracks in relation to all tracks classified, where tracks with a prediction value larger than 0.5 are classified as relevant for the given user, and all other tracks are considered irrelevant for the user).

Please note that we deliberately chose a classification-based recommendation approach and refrained from utilizing more elaborate recommender approaches such as context-aware matrix factorization [34] or tensor-based factorization approaches [235] as we aim to focus on user modeling aspects in this paper. Hence, we chose to compare different user models based on a simple classification-based recommendation approach which also allows us to get a deeper understanding of the contribution of individual features of the user model (cf. Section 8.6).

For the classification task carried out, we require a rating for each track that allows us to define whether a given track was listened to (and thus, considered relevant) for a given user. Hence, we add a binary factor (rating) to the processed dataset: for each unique <user, track> combination, the rating $r_{i,j}$ is 1 if the user u_i has listened to track t_j at least once. Please note that users and tracks may be represented by different models as described in Section 8.5.1. Due to a lack of publicly available data, our dataset does not contain any implicit feedback of users (i.e., skipping behavior, session durations, or dwell times during browsing the catalog). This is why we cannot estimate any preference towards an item a user has not listened to as proposed by [199]. Thus, we assume tracks the user has not listened to (in the case of implicit data, all non-observed tracks) as negative examples [199]. Even though there is a certain bias towards negative values as some missing values might be positive, Pan, Zhou, Cao, Liu, Lukose, Scholz, and Yang [341] found that this method for rating estimation works well. The rating $r_{i,j}$ for a given user u_i and given track t_j can now be defined as stated in Equation 8.1.

$$r_{i,j} = \begin{cases} 1 & \text{if } u_i \text{ listened to } t_j \\ 0 & \text{otherwise} \end{cases} \tag{8.1}$$

We train an XGBoost model that performs a binary classification on the relevance of tracks for the given users. We extract the probabilities underlying the classification decision, which can be used to (i) perform a ranking of tracks by their probability of relevance in the recommendation task which allows us to conduct a ranking-based evaluation of the proposed models, and (ii) evaluate the predictive performance of the proposed models by computing error metrics.

8.5. Experiment Design

This section reports on the experiments conducted for evaluating the previously described culture-aware recommender system.

8.5.1. Experimental Setup

In the following, we first present the user models evaluated and describe the evaluation method utilized for capturing the recommendation performance of the proposed user model.

Evaluation Strategy

To evaluate the performance of the proposed contextual user modeling in regard to recommendation quality, we perform a per-user evaluation. Therefore, we use each user's listening history and perform a *leave-k-out* evaluation per user (also referred to as hold-out evaluation) [64, 106], where we set k to 50 (as described later in this section).

The underlying dataset only provides items with positive feedback [199] (i.e., items that have been listened to by the user) gathered via users' listening histories. As the recommendation task is transformed into a rating prediction task, we require the dataset to also include negative examples. Therefore (and as described previously in Section 8.4.4), for each user, we randomly add tracks the user did not interact with (i.e., tracks t_j with $r_{i,j} = 0$ for the given user u_i) to the dataset until the listening history of each user in both the training and test sets are filled with 50% relevant and 50% non-relevant items for the user. We chose to oversample the positive class to avoid class imbalance and hence, a bias towards the negative class (the number of tracks not listened to is much larger than the number of tracks listened to, for all users).

As we aim to evaluate the benefit of adding cultural aspects in a track recommendation scenario, we also need to characterize tracks. For our proposed model, we rely on the acoustic features of each track and add these to the track vector. However, we also need to assign cultural features to tracks to be able to match users of a certain culture with tracks that are listened to by users with a similar cultural background. This is particularly relevant for tracks in the negative class. Preliminary experiments showed that we cannot assign randomly computed cultural features or the cultural features of the current user to tracks as this causes the XGBoost model to learn that all tracks with the user's culture assigned belong to the positive class, whereas all tracks from any other culture (i.e., culture information that is consistent across a number of users or purely random culture information) belong to the negative class. Therefore, we propose to assign the cultural features of the country in which the track is most popular to each track. We argue that the track is most characteristic and representative for the country in which the track is most popular. Therefore, we first compute the playcounts of each track in each country within the dataset. Next, we normalize the playcount (PC) of each track $t \in T$ (i.e., the universe of tracks in the dataset) in each country c by the total amount of listening events of the country (i.e., we compute $\frac{PC(c,t)}{\sum_{j \in T} PC(c,j)}$ for each country c and for each track t). This allows us to infer the country in which it accounts for the highest share of listening events and hence, is most popular. We subsequently assign the culture of this country to the track. For obtaining negative samples (tracks), we randomly select a track from the dataset that the current user has not listened to and again assign this track the cultural features of the country where the track is most popular in.

Based on the dataset that now contains an equal amount of positive and negative samples for each user, we perform a leave-k-out evaluation strategy. Therefore, we have to compute a hold-out set of size k for each user: along the lines of previous research [139, 185], we randomly select 50 positive samples (tracks that the user has listened to) and 500 negative samples (tracks the user has not listened to). These 550 tracks form the test set for each user, whereas the recommender system is trained on the remainder of the dataset. Subsequently, we compute the predicted ratings for the tracks in the test set as presented in Section 8.4.4, aiming to rank the 50 positive samples on top, whereas the negative samples should be ranked on the bottom of the ranked list of recommendations.

Evaluated Models and Baselines

To assess the performance of each of the proposed user models, variations thereof and two baseline approaches in terms of recommendation quality, we separately evaluate these different user models and compare their performance. An overview of the evaluated modeling approaches is depicted in Table 8.3. The evaluated models describe a user either by the user's individual music preferences described by the acoustic features of the tracks the user listened to (U_AF), the user's cultural/socio-economic background described by Hofstede's dimensions (U_HOF) and the World Happiness Report (U_WHR), or the user identifier (U_ID). Similarly, we describe tracks by their acoustic features (T_AF), the culture they are embedded in (T_HOF and T_WHR) or by their track identifier (T_ID). Please note that we include the user and track identifiers in the respective models as this allows us to extend and directly compare the approaches to a baseline model (User + Track), that is only based on these two identifiers. As can be seen from Table 8.3, we evaluate the music-cultural model (*Music + Culture*) as proposed in Section 8.4.3. We also individually evaluate the performance of a model solely relying on musical preferences of users and features of tracks (*Music* model), and analogously a model that describes users and tracks by their cultural background (*Culture* model).

Furthermore, we investigate a set of baselines to compare our proposed models to. First, we evaluate an approach that uses each user's listening history and additionally, utilizes the user's country code (e.g., US for users from the United States) as contextual information for both the user and the track (*Country* model). Here, we aim to evaluate whether the country code may act as a proxy for cultural factors of users. Furthermore, we evaluate a context-agnostic baseline relying solely on the users' listening histories and hence, a model that solely relies on the user and track ids for classification (*User + Track*) in a traditional collaborative filtering approach.

Evaluation Metrics

We model the context-aware recommendation of tracks as a rating prediction task, therefore we use the root mean squared error (RMSE) and mean absolute error (MAE) to measure the prediction error. We compute the RMSE and MAE for each individual user and consequently compute the average among all users. Furthermore, we are also

Model	**User Features**	**Track Features**
Music + Culture	U_ID, U_AF, U_WHR, U_HOF	T_ID, T_AF, T_WHR, T_HOF
Music	U_ID, U_AF	T_ID, T_AF
Culture	U_ID, U_WHR, U_HOF	T_ID, T_HOF, T_WHR
Country	U_ID, U_Country_ID	T_ID, T_Country_ID
User + Track	U_ID	T_ID

Table 8.3.: Overview of evaluated models, where features prefixed with U describe a user and features prefixed with T describe a track; the models on two last rows serve as baselines.

interested in a decision-based evaluation [79] of our approach and therefore, compute *precision*, *recall*, and the F_1-measure to assess the top-n accuracy [105], where n is the number of top-ranked track recommendations that is evaluated. Therefore, we require the set of computed recommendations to be ranked. Hence, we rank the track recommendation candidates with respect to the probability that they belong to the positive class in descending order and compute the top-n track recommendations. Next, we have to transform the rating prediction task into a binary classification task [341] for deciding whether a given track is relevant or not for a given user. For our experiments, we consider all predicted probabilities $P(u, i) > 0.5$ as a predicted interaction and thus, we consider these items as relevant, all others as irrelevant.[10] For assessing the overall *precision*, *recall*, and F_1-measure of the evaluated recommender systems, we compute the measures for each individual user and compute the average among all users. For computing the *recall* measure, all relevant items in the test set are considered, independent of the number of recommendations. Thus, there is a natural cap for *recall*, namely the number of recommendations divided by the number of relevant items in the test set.

Regarding the number n of evaluated recommendations, we argue that exposing a user to more than 10–20 tracks at a time might provoke choice overload and hence, is barely meaningful. The problem of choice overload has been addressed by [57] who state that user satisfaction is highest when presenting the user with top-5 to top-20 items—assuming that the recommendation list contains a sufficient number of relevant items for the user. Hence, we are particularly interested in the performance of the proposed recommendation approaches for lower values of n. Furthermore, we argue that in the presented scenario, precision is the more important measure to consider from a user perspective as it able to capture the user's effective utility of the provided recommendations better [46] and hence, the practical value of the recommender system for the user. Thus, we argue that particularly the precision@10 results are relevant for our evaluation. As for the tuning

[10] Please note that this distinction between the positive and negative class is also utilized by XGBoost for binary classification tasks based on logistic regression.

of XGBoost parameters, we performed a preliminary cross-evaluation aiming to optimize precision values for the proposed models and hence, set the maximum number of trees to learn the models to 1,000. For all other parameters, we rely on the default settings.

8.6. Experimental Results and Discussion

In the following, we first present the findings of the top-n recommendation evaluation task (Section 8.6.1), before presenting the evaluation of the underlying rating prediction task in Section 8.6.2. Subsequently, we elaborate on the importance of individual features of the proposed user model (Section 8.6.3) and discuss the limitations of the approach (Section 8.6.4).

8.6.1. Top-n Recommendation Evaluation

Table 8.4 shows the results obtained by the evaluated user models (cf. Table 8.3), where we consider the top-10 ranked recommended tracks for the evaluation. Regarding the precision of the computed recommendations, we observe that the best results are obtained by the proposed Music + Culture model, which incorporates both the user's general musical preferences and the cultural background of the user. This model reaches a precision@10 of 0.98, whereas the Music model reaches a precision of 0.95 and the Culture model a precision of 0.31, respectively. Compared to the baselines, we observe that using only the country of the user as a proxy for cultural aspects (Country model) achieves a precision value of 0.83, whereas the User + Track model performs worse, reaching a precision value of 0.13.

Regarding the recall values obtained, we observe that again, the Music + Culture model performs best (0.63), followed by the Music (0.59) and Country (0.52) models. The User + Track baseline again reaches a lower value (0.08), whereas the Country model again performs well (0.52). For the sake of completeness, we also list the F_1 values obtained by the individual models, which are consistent with the individual findings regarding recall and precision. In preliminary baseline experiments, we have also compared our approach with a traditional context-agnostic matrix factorization approach. Singular value decomposition based on implicit feedback achieved a precision of 0.49, a recall of 0.10, and an F_1-score of 0.17. As already elaborated, we consider the precision metric more relevant in this scenario. Thus, these baseline results show that the proposed models do indeed contribute to recommendation quality.

Figure 8.1 shows a precision/recall plot of the evaluated approaches for $n = 1 \ldots 50$ track recommendations. From this plot, we again observe the superior performance of the music-cultural user model across all evaluated lengths of recommendation lists n. The plot also highlights the difference between the two models that incorporate acoustic features for describing musical preferences (Music + Culture and Music) and

Model	**Prec**	**Rec**	**F_1**
Music + Culture	0.98 (± 0.04)	0.63 (± 0.15)	0.75 (± 0.10)
Music	0.95 (± 0.06)	0.59 (± 0.15)	0.72 (± 0.11)
Country	0.83 (± 0.11)	0.52 (± 0.12)	0.63 (± 0.10)
Culture	0.31 (± 0.15)	0.18 (± 0.08)	0.24 (± 0.09)
User + Track	0.13 (± 0.10)	0.08 (± 0.06)	0.13 (± 0.06)

Table 8.4.: Precision, recall, and F_1-score for all proposed models (sorted by performance; standard deviation in parenthesis).

the remaining user models that do not exploit this information, where precision and recall are both substantially lower. These findings underline that the musical preference of users is paramount for recommendation scenarios. We can also observe that using the user's country as a proxy for their cultural background does indeed contribute. Naturally, including a set of cultural features to describe the user's cultural background also allows to exploit a more comprehensive, multi-dimensional notion of similarity between users [420], which can be exploited by the recommender system. We also have experimented with combining musical features and country code, however, this did not increase performance compared to using only musical features.

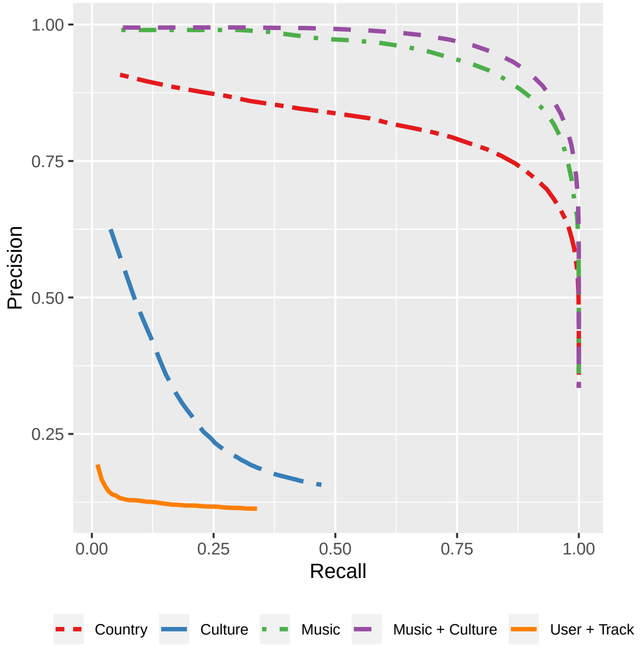


Figure 8.1.: Precision-recall-curves for top-$n = 1 \dots 50$ recommendations for all models.

8.6.2. Rating Prediction Evaluation

Besides the decision-based evaluation regarding recall and precision, we are also interested in the prediction accuracy of the individual user models. Table 8.5 presents the RMSE and MAE per user across all tracks within the user's test set. These findings are in line with the decision-based findings as the lowest RMSE is again achieved by the Music + Culture model (RMSE of 0.15). In comparison, relying solely on acoustic features to describe users and tracks (Music model) achieves a RMSE of 0.17, whereas relying on cultural aspects only results in a RMSE of 0.88. The baseline approaches reach RMSE values of 0.36 (Country model) and 0.93 (User + Track model), respectively. The evaluation of mean absolute errors of the individual models is consistent with the findings of the RMSE findings.

Model	**RMSE**	**MAE**
Music + Culture	0.15	0.02
Music	0.17	0.03
Country	0.36	0.13
Culture	0.88	0.77
User + Track	0.93	0.85

Table 8.5.: RMSE and MAE of all models.

8.6.3. Influence of Features

Apart from the performance of the proposed music-cultural user model in regard to recommendation quality, we are also interested in the contribution of the individual features of the user model to the trained XGBoost classification model. Therefore, we utilize the gain of each feature in the XGBoost model [93], which is a measure for the improvement in accuracy when adding a split on the given feature to the tree. This gain is computed for each feature in every tree of the trained model and is then averaged to a final gain value for each feature. Figure 8.2 shows the contribution of the top-30 individual features to classification performance of the proposed music-cultural user model. Please recall that in the proposed model, both users and tracks are described by musical and cultural features (cf. Table 8.3). Hence, we color the bars of user features in blue and track features in red. In total, acoustic features account for 93% of the gain (76% user features, 17% track features), WHR features account for 4% and Hofstede's dimensions for 3% of the gains.

The results show that the major contributing features are related to the acoustic features that describe the user's musical preference and the tracks. This high importance of acoustic features when it comes to describing users is congruent with the analyses of [363] and in line with the findings of the top-n recommendation evaluation, where the Music model was the second best performing model. The features that contribute most to the classification accuracy (and hence, recommendation performance) are the average acousticness (user_acousticness_avg), instrumentalness (user_instrumentalness_avg) and danceability (user_danceability_avg) of tracks the user has listened to. As for the track features, acousticness and instrumentalness are also the main contributing features. This high contribution of instrumentalness and acousticness is in line with previous findings [362], where these two features have been shown to discriminate tracks well in a principal component analysis. These findings are also congruent with the results of the evaluation conducted, where the user model that solely relies on the user's preferences achieved the second best recall and precision values (performing substantially better than the Culture, Country, and User + Track models). However, while socio-economic factors are not among the top contributing features, socio-economic features nevertheless contribute to the recommendation quality and make a decisive difference regarding recommendation performance. The user features contributing most are healthiness, social support, happiness, gdp and masculinity and for tracks, the happiness and social support features provide the highest gain. While WHR features contribute more in our scenario, features stemming from both sources (WHR and Hofstede's cultural dimensions) are among the top-contributing features; this also supports our choice to include both social and economic features in the user model as both contribute to higher recommendation performance.

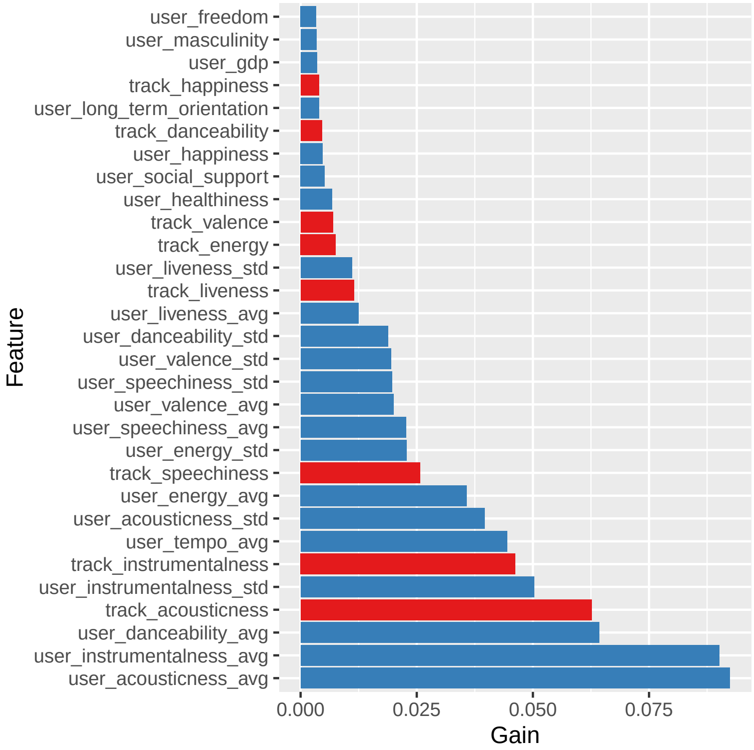


Figure 8.2.: Information gain of the top 30 individual user and track features of the Music + Culture model.

8.6.4. Discussion and Limitations

We believe that the proposed music-cultural user model and the conducted evaluation are an important first step towards culture-aware music recommender systems. The obtained results show that the proposed music-cultural user model outperforms all other evaluated models. However, we still see a few limitations of our approach, which we will elaborate on in the following. First, we currently represent the musical preferences of a user by utilizing the average of the acoustic features of the tracks the user has listened to and the standard deviation thereof. While we believe that this method is sufficiently elaborate for the experiments conducted, this is a rather naive approach towards representation and does not reflect the diverse and often context-related musical preferences of users. Similarly, we currently use a rather simple majority voting approach for assigning cultural features to tracks. However, in the paper at hand, we are particularly interested in the influence of individual features and characteristics of users, their cultural background, and tracks on the recommendation performance and, hence, deliberately refrain from utilizing a more comprehensive user model. Nevertheless, looking into creating more comprehensive and complex user models based on the cultural background of users is part of our future research agenda. For instance, [508] employed Gaussian Mixture Models (GMM) for modeling a user's diverse tastes of music and showed that utilizing such a GMM approach in combination with the acoustic features of the tracks the user listed to is able to capture a user's musical preferences well.

The test set creation procedure applied (random 50 positive and 500 negatives samples per user) allows for evaluating the ability to distinguish positive and negative samples. We have also experimented sampling 10 relevant and 100 irrelevant tracks for each user, however, we argue that given the high number of listening events per user in the dataset, sampling 50 positive and 500 negative tracks reflects a more suitable scenario. The results achieved were high in precision and low on the prediction error metrics, showing that the proposed models were able to detect the 50 positive samples and rank these on top.

As already stated in Section 8.4.4, we consider the classification-based approach for the computation of recommendations as a baseline regarding the actual recommender system. However, we believe that even though the method is rather simple, it provides us with conclusive results regarding the user models evaluated, where we clearly put our focus on.

8.7. Interplay Between Country Characteristics and Music Preferences

In the following, we analyze the cultural/socio-economic and acoustic features on a country level more thoroughly, aiming to uncover country-specific patterns of their inhabitants' music preferences in terms of acoustic features and to identify similarities and differences between countries (Section 8.7.1). We further investigate to which extent cultural/socio-economic and acoustic features correlate with each other, on a per-feature-basis (Section 8.7.2).

8.7.1. Country-specific Differences of Acoustic Feature Preferences

	Danceability	Energy	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo
AR	0.512 (0.091)	0.739 (0.140)	0.048 (0.017)	0.113 (0.163)	0.059 (0.166)	0.145 (0.034)	0.482 (0.122)	123.113 (7.756)
AT	0.476 (0.102)	0.766 (0.172)	0.059 (0.025)	0.106 (0.182)	0.127 (0.227)	0.154 (0.042)	0.405 (0.133)	124.400 (8.483)
AU	0.491 (0.100)	0.746 (0.157)	0.057 (0.028)	0.112 (0.172)	0.119 (0.228)	0.153 (0.043)	0.435 (0.129)	123.562 (9.116)
BE	0.507 (0.106)	0.718 (0.170)	0.056 (0.029)	0.143 (0.198)	0.165 (0.260)	0.148 (0.045)	0.428 (0.129)	122.783 (8.825)
BG	0.491 (0.101)	0.801 (0.135)	0.062 (0.029)	0.063 (0.123)	0.117 (0.215)	0.159 (0.044)	0.418 (0.131)	124.052 (10.034)
BR	0.509 (0.089)	0.758 (0.148)	0.053 (0.024)	0.114 (0.173)	0.029 (0.112)	0.154 (0.054)	0.478 (0.121)	124.566 (10.589)
CA	0.495 (0.098)	0.736 (0.159)	0.056 (0.028)	0.126 (0.180)	0.117 (0.222)	0.153 (0.048)	0.441 (0.128)	123.161 (8.588)
CH	0.518 (0.106)	0.706 (0.169)	0.053 (0.025)	0.161 (0.197)	0.134 (0.251)	0.142 (0.037)	0.442 (0.140)	122.438 (8.510)
CL	0.495 (0.099)	0.769 (0.136)	0.054 (0.022)	0.091 (0.155)	0.072 (0.170)	0.151 (0.041)	0.455 (0.131)	124.367 (7.929)
CN	0.502 (0.118)	0.643 (0.197)	0.051 (0.041)	0.232 (0.249)	0.153 (0.279)	0.145 (0.074)	0.393 (0.153)	121.190 (13.016)
CO	0.532 (0.097)	0.755 (0.129)	0.050 (0.017)	0.099 (0.154)	0.073 (0.169)	0.142 (0.036)	0.486 (0.141)	123.085 (7.644)
CZ	0.487 (0.097)	0.769 (0.154)	0.057 (0.024)	0.094 (0.166)	0.139 (0.235)	0.157 (0.051)	0.418 (0.137)	123.901 (8.317)
DE	0.502 (0.110)	0.776 (0.154)	0.063 (0.039)	0.094 (0.166)	0.114 (0.227)	0.158 (0.048)	0.445 (0.138)	124.570 (9.937)
DK	0.524 (0.099)	0.701 (0.172)	0.052 (0.026)	0.161 (0.203)	0.107 (0.220)	0.147 (0.059)	0.445 (0.125)	121.128 (8.498)
EE	0.504 (0.095)	0.755 (0.144)	0.056 (0.028)	0.091 (0.151)	0.147 (0.246)	0.147 (0.037)	0.428 (0.124)	124.531 (10.383)
ES	0.514 (0.101)	0.733 (0.163)	0.052 (0.023)	0.141 (0.196)	0.085 (0.194)	0.148 (0.038)	0.474 (0.136)	123.432 (8.257)
FI	0.487 (0.103)	0.806 (0.132)	0.062 (0.032)	0.062 (0.131)	0.122 (0.219)	0.166 (0.042)	0.428 (0.136)	123.707 (8.277)
FR	0.533 (0.113)	0.704 (0.159)	0.057 (0.035)	0.152 (0.193)	0.152 (0.249)	0.144 (0.046)	0.452 (0.145)	120.900 (9.452)
GR	0.473 (0.091)	0.709 (0.161)	0.049 (0.020)	0.124 (0.193)	0.198 (0.267)	0.144 (0.033)	0.397 (0.127)	121.519 (8.147)
HR	0.473 (0.101)	0.752 (0.157)	0.056 (0.026)	0.110 (0.165)	0.158 (0.245)	0.151 (0.038)	0.418 (0.132)	122.991 (8.289)
HU	0.494 (0.116)	0.800 (0.144)	0.064 (0.033)	0.066 (0.140)	0.189 (0.283)	0.162 (0.045)	0.408 (0.146)	124.793 (10.081)
ID	0.510 (0.089)	0.716 (0.165)	0.048 (0.023)	0.150 (0.195)	0.040 (0.144)	0.147 (0.048)	0.448 (0.126)	123.762 (12.311)
IE	0.503 (0.092)	0.696 (0.174)	0.051 (0.024)	0.164 (0.211)	0.120 (0.222)	0.146 (0.040)	0.445 (0.125)	122.503 (8.780)
IN	0.487 (0.104)	0.704 (0.186)	0.053 (0.037)	0.158 (0.234)	0.143 (0.266)	0.145 (0.058)	0.398 (0.134)	121.598 (11.939)
IR	0.455 (0.101)	0.599 (0.215)	0.049 (0.031)	0.278 (0.265)	0.181 (0.281)	0.133 (0.038)	0.298 (0.137)	119.224 (12.176)
IT	0.501 (0.090)	0.705 (0.166)	0.051 (0.023)	0.158 (0.199)	0.085 (0.186)	0.144 (0.036)	0.444 (0.130)	122.752 (8.591)
JP	0.512 (0.102)	0.729 (0.189)	0.056 (0.032)	0.153 (0.220)	0.156 (0.268)	0.153 (0.060)	0.474 (0.159)	123.181 (13.594)
LT	0.477 (0.105)	0.750 (0.154)	0.054 (0.020)	0.097 (0.165)	0.182 (0.264)	0.146 (0.037)	0.393 (0.124)	122.687 (8.250)
LV	0.494 (0.099)	0.730 (0.172)	0.056 (0.033)	0.122 (0.192)	0.158 (0.263)	0.149 (0.046)	0.399 (0.125)	121.961 (12.291)
MX	0.529 (0.091)	0.757 (0.124)	0.051 (0.023)	0.091 (0.145)	0.079 (0.191)	0.146 (0.040)	0.485 (0.130)	124.044 (8.197)
NL	0.518 (0.100)	0.705 (0.171)	0.053 (0.029)	0.154 (0.202)	0.115 (0.235)	0.144 (0.040)	0.446 (0.130)	122.553 (9.230)
NO	0.507 (0.101)	0.710 (0.162)	0.052 (0.024)	0.147 (0.193)	0.117 (0.225)	0.145 (0.037)	0.435 (0.130)	122.500 (8.098)
NZ	0.486 (0.100)	0.771 (0.144)	0.059 (0.026)	0.085 (0.154)	0.136 (0.252)	0.158 (0.044)	0.432 (0.134)	124.857 (9.177)
PL	0.504 (0.102)	0.766 (0.145)	0.065 (0.046)	0.093 (0.155)	0.099 (0.208)	0.154 (0.048)	0.436 (0.137)	122.569 (10.738)
PT	0.478 (0.107)	0.736 (0.178)	0.056 (0.028)	0.129 (0.203)	0.145 (0.241)	0.150 (0.041)	0.407 (0.132)	122.887 (9.709)
RO	0.476 (0.113)	0.720 (0.166)	0.053 (0.023)	0.121 (0.184)	0.224 (0.285)	0.142 (0.034)	0.373 (0.139)	121.389 (7.864)
RS	0.499 (0.119)	0.745 (0.154)	0.059 (0.034)	0.102 (0.167)	0.139 (0.240)	0.151 (0.041)	0.424 (0.143)	121.517 (8.257)
RU	0.485 (0.099)	0.790 (0.146)	0.061 (0.032)	0.071 (0.149)	0.141 (0.247)	0.161 (0.049)	0.415 (0.136)	124.464 (10.373)
SE	0.512 (0.096)	0.725 (0.159)	0.053 (0.028)	0.138 (0.185)	0.115 (0.227)	0.147 (0.036)	0.454 (0.123)	123.027 (7.834)
SK	0.479 (0.103)	0.755 (0.172)	0.064 (0.040)	0.109 (0.178)	0.184 (0.263)	0.156 (0.040)	0.381 (0.136)	122.172 (9.100)
TR	0.498 (0.095)	0.669 (0.184)	0.049 (0.023)	0.199 (0.228)	0.128 (0.238)	0.137 (0.040)	0.398 (0.125)	119.935 (9.252)
UK	0.512 (0.096)	0.723 (0.163)	0.054 (0.027)	0.134 (0.192)	0.110 (0.227)	0.148 (0.041)	0.465 (0.128)	123.424 (9.642)
US	0.507 (0.100)	0.721 (0.163)	0.057 (0.044)	0.140 (0.194)	0.108 (0.221)	0.150 (0.049)	0.461 (0.130)	122.624 (9.813)
VE	0.515 (0.101)	0.777 (0.113)	0.054 (0.022)	0.070 (0.120)	0.082 (0.198)	0.151 (0.042)	0.476 (0.152)	124.961 (10.287)

Table 8.6.: Means and standard deviations (in parenthesis) of acoustic preferences of each country's users. Highest value of each acoustic property is printed in blue; lowest in red. Countries are sorted alphabetically according to their country code.

To obtain insights into country-specific particularities of the acoustic properties of music consumption, we provide an overview of the investigated acoustic features (and their standard deviations) per country, computed over all users in each country in Table 8.6. Overall, we observe pronounced differences between countries for most of the properties, but also non-negligible standard deviations within countries, indicating partly substantial variances in music preferences among citizens. Highest danceability in music preferences can be found in France (0.533), Colombia (0.532), and Mexico (0.529); lowest in Iran (0.455). Notably, Iran is also the country with lowest music energy (0.599) in its population's preferences. In contrast, the populations of Finland (0.806), Bulgaria (0.801), and Hungary (0.800) like highly energetic music. This is further evidenced when investigating their preferred music styles, which include several variants of the genre metal. As for speechiness, lowest figures are found in Indonesia and Argentina (both 0.048), whereas music listeners in Poland (0.065) tend to commonly listen to music featuring spoken words such as hip-hop or rap. Acousticness is lowest for Finland (0.062) and Bulgaria (0.063); highest for Iran (0.278), China (0.232), and Turkey (0.199). As for instrumentalness, the lowest-scoring countries are Brazil (0.029), Indonesia (0.040), and Argentina (0.059). On the other end, users in Romania (0.224) and Greece (0.198) particularly like non-vocal instrumental music. Regarding liveness, Iran (0.133) and Turkey (0.137) show lowest values, whereas Finland (0.166) has the highest figures for this attribute. This may be explained by Finns having a particular preference for live music and by Finland having a very vivid music performing culture and therefore a large number of hobby musicians as well as (semi-)professional bands. Music listened to by Iranian users scores lowest on the dimension of valence, on average (0.298). In stark contrast, music consumed in South and Middle America scores highest on this dimension; in particular, users in Colombia (0.486), Mexico (0.485), Argentina (0.482), and Brazil (0.478) tend to listen to music that may be suited to reflect or evoke positive emotions. Finally, when it comes to tempo, users in Iran and Turkey tend to prefer slower music, on average (both around 120 bpm). On the other hand, Venezuela, New Zealand, Hungary, and Germany are more into faster music, on average (around 125 bpm).

8.7.2. Correlations Between Cultural Background and Music Preferences

To uncover possible relationships between acoustic properties of a country's inhabitants' music preferences and the cultural or socio-economic characteristics, we investigate the correlation between each of the acoustic features and the cultual/socio-economic dimensions. Tables 8.7 and 8.8 depict Spearman's rank-order correlation coefficients for Hofstede's cultural features and WHR socio-economic characteristics, respectively. We use rank-order correlation to cope with the different value ranges of the various dimensions investigated and compute these correlations considering all users in our dataset as observations. To describe each user's aggregated musical feature vector, we follow the same approach as detailed in Section 8.4.3. Correlations larger than 0.1 (or lower than -0.1) are highlighted in bold. Statistically significant correlations are marked with an asterisk.

	PD	IDV	MAS	UA	LTO	IND
Danceability	-0.035*	0.044*	0.023*	-0.052*	-0.024*	0.072*
Energy	0.056*	-0.102*	-0.014	**0.116***	0.076*	**-0.115***
Speechiness	0.022*	-0.034*	0.016*	0.085*	0.065*	-0.096*
Acousticness	-0.056*	**0.105***	0.026*	**-0.122***	-0.086*	**0.125***
Instrumentalness	-0.012	0.011	-0.029*	0.038*	0.055*	-0.055*
Liveness	0.021*	-0.042*	-0.014	0.059*	0.035*	-0.065*
Valence	-0.042*	0.059*	0.047*	-0.076*	-0.063*	**0.114***
Tempo	0.009	-0.041*	0.008	0.031*	0.043*	-0.025*

Table 8.7.: Spearman rank-order correlations between users' acoustic properties of listening behavior and cultural features (Hofstede). Correlations >0.1 are highlighted in **bold** face. Statistically significant correlations at $p < 0.001$ are marked with an asterisk (*).

	Happiness	GDP	Social Sup.	Life Exp.	Freedom	Trust	Generosity
Danceability	0.035*	0.036*	-0.010	0.049*	0.037*	0.051*	0.052*
Energy	-0.036*	-0.067*	0.056*	-0.056*	-0.026*	-0.033*	**-0.101***
Speechiness	-0.018*	-0.007	0.059*	-0.017*	0.011	-0.004	-0.067*
Acousticness	0.055*	0.079*	-0.046*	0.070*	0.039*	0.048*	**0.118***
Instrumentalness	-0.031*	0.030*	0.042*	0.040*	0.006	0.001	-0.044*
Liveness	0.005	-0.019*	0.056*	-0.030*	0.001	-0.008	-0.048*
Valence	0.071*	0.047*	0.008	0.051*	0.044*	0.064*	0.084*
Tempo	0.004	-0.025*	0.046*	-0.015*	0.001	0.003	-0.016*

Table 8.8.: Spearman rank-order correlations between users' acoustic properties of listening behavior and socio-economic features (WHR). Correlations >0.1 are highlighted in **bold** face. Statistically significant correlations at $p < 0.001$ are marked with an asterisk (*).

As a general observation, while almost all correlations are significant (even at $p < 0.001$), most are only weak, which hints at the different nature of aspects to compare. Nevertheless, some interesting observations can be made. Focusing on Table 8.7, we observe notable correlations for the cultural trait of indulgence (IND). More precisely, a positive correlation between IND and acousticness (0.125) as well as valence (0.114) is identified. This means that societies that like to engage in joyful activities tend to listen to music that has a higher probability of being acoustic and to music that evokes positive emotions, which makes sense. At the same time, indulging populations tend to prefer lower energy levels in music (correlation of -0.115), which hints at a preference for more relaxing music. Furthermore, uncertainty avoidance (UA) is positively correlated with music energy level (0.116), but negatively with acousticness (-0.122). Societies characterized by stiff codes and laws therefore tend to prefer more energetic music, but lower amounts of acoustic tracks. Also, there is a positive correlation between individualism (IDV) and acousticness (0.105).

Comparing the acoustic features with the WHR dimensions, cf. Table 8.8, we can only observe two correlations exceeding the threshold. Both relate to the aspect of generosity. More precisely, we see a positive correlation between generosity and acousticness (0.118), whereas a negative one with energy (-0.101). More generous populations therefore tend to prefer lower energetic music, but larger amounts of acoustic tracks.

8.8. Conclusion and Future Work

The contributions of this work are two-fold: (i) we introduced a novel music-cultural user model that jointly relies on acoustic song features and culture-related features to describe the user's musical preferences and cultural background and (ii) we proposed a recommender system that leverages these features as contextual information. Our evaluations based on a dataset comprising more than 55,000 users showed that the proposed user model is able to outperform models that incorporate either solely musical aspects or cultural aspects and the evaluated baseline methods (relying on user's country as a proxy for culture, utilizing solely the user's and track's identifiers). In regard to both recall and precision, we show that adding contextual information obtained via incorporating audio features of tracks, data extracted from the World Happiness Report and Hofstede's cultural dimensions contributes to improved recommendations when compared to the baseline approaches. Particularly, we find that a combination of acoustic features of the songs a user listened to (describing the individual music preferences of a user) and the World Happiness Report as a description of the cultural/socio-economic background of the user performs best.

Future work includes extending the user models with further data utilized for capturing cultural aspects of users (e.g., the Quality of Government dataset [458]). Moreover, we are particularly interested in analyzing the country-specific influence of each of the individual features of the proposed user models on the overall recommendation performance to get a deeper understanding for features that are able to capture country-specific listening patterns. Regarding the representation of both the musical preferences and cultural aspects, we plan to investigate more sophisticated modeling approaches. Particularly regarding the representation of musical preferences of users, we believe that, e.g., using Gaussian mixture models will allow for a more differentiated representation of users and their (possibly diverse and broad) preferences. Finally, we aim to transcend the country level for our culture-based analyses, e.g., focusing on culturally similar users that live in the same cultural region (but not necessarily in the same country).

9. Support the Underground: Characteristics of Beyond-Mainstream Music Listeners

Publication

D. Kowald, P. Müllner, E. Zangerle, C. Bauer, M. Schedl, and E. Lex. Support the Underground: Characteristics of Beyond-Mainstream Music Listeners. *EPJ Data Science*, 10(1):1–26, 2021. DOI: 10.1140/epjds/s13688-021-00268-9

Abstract

Music recommender systems have become an integral part of music streaming services such as Spotify and Last.fm to assist users navigating the extensive music collections offered by them. However, while music listeners interested in mainstream music are traditionally served well by music recommender systems, users interested in music beyond the mainstream (i.e., non-popular music) rarely receive relevant recommendations. In this paper, we study the characteristics of beyond-mainstream music and music listeners and analyze to what extent these characteristics impact the quality of music recommendations provided. Therefore, we create a novel dataset consisting of Last.fm listening histories of several thousand beyond-mainstream music listeners, which we enrich with additional metadata describing music tracks and music listeners. Our analysis of this dataset shows four subgroups within the group of beyond-mainstream music listeners that differ not only with respect to their preferred music but also with their demographic characteristics. Furthermore, we evaluate the quality of music recommendations that these subgroups are provided with four different recommendation algorithms where we find significant differences between the groups. Specifically, our results show a positive correlation between a subgroup's openness towards music listened to by members of other subgroups and recommendation accuracy. We believe that our findings provide valuable insights for developing improved user models and recommendation approaches to better serve beyond-mainstream music listeners.

9.1. Introduction

In the digital era, users have access to continually increasing amounts of music via music streaming services such as Spotify and Last.fm. Music recommender systems have become an essential means to help users deal with content and choice overload as they assist users in searching, sorting, and filtering these extensive music collections. Simultaneously, both music listeners and artists benefit from the employed segmentation and personalization approaches that are typically leveraged in music recommendation approaches [418]. As a result, users with different preferences and needs can be targeted in various ways with the goal that all users are presented the information and content that they need or prefer. This also means that current recommendation techniques should serve all users equally well, independent of their inclination to popular content.

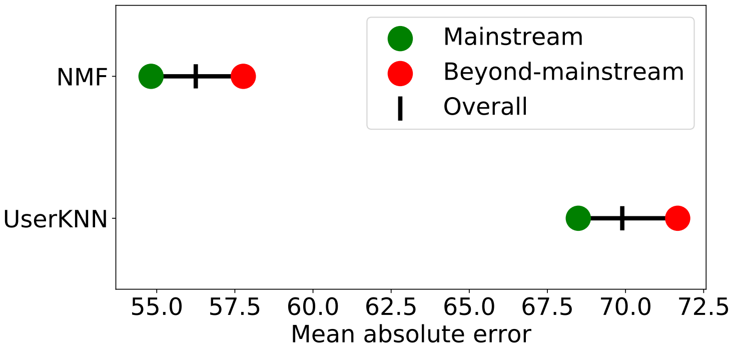


Figure 9.1.: Recommendation accuracy measured by the mean absolute error (MAE) of a non-negative matrix factorization-based approach (i.e., NMF [302]) and a neighborhood-based approach (i.e., UserKNN [190]) for mainstream and beyond-mainstream user groups in Last.fm. We see that beyond-mainstream users receive a substantially lower recommendation quality (i.e., higher MAE) compared to mainstream music listeners. Thus, for recommender systems, it is harder to provide high-quality recommendations to beyond-mainstream music listeners than to mainstream music listeners.

Present work. In the paper at hand, we focus on music consumers who listen to music beyond the mainstream (i.e., users who listen to non-popular music) in the music streaming platform Last.fm[1]. As highlighted in Figure 9.1, current recommender systems do not work well for consumers of beyond-mainstream music (see Section 9.3.5 for details on this analysis). In contrast, music consumers who listen to popular music seem to get better recommendations. This finding is not essentially new. In fact, it is a widely-known problem that recommender systems (and those based on collaborative filtering, in particular) are prone to popularity bias, which leads to the behavior that long-tail

[1] https://www.last.fm/

items (i.e., items with few user interactions) have little chance being recommended. This phenomenon is also present across different application domains such as movies [4] or music [82].

Our previous work [279] has shown that users interested in beyond-mainstream music tend to have larger user profile sizes (i.e., individual users show a high(er) number of distinct artists they have listened to) compared to users interested in mainstream music. The observation that beyond-mainstream music listeners produce a substantial amount of digital footprints motivates the need to improve the recommendation quality for this group. However, although related research has already studied the long-tail recommendation problem (e.g., [79, 80, 162, 470]; see Section 9.2 for a more detailed discussion of related work), it is still a fundamental challenge to understand and identify the characteristics of beyond-mainstream music and beyond-mainstream music listeners. Additionally, related work [460] has shown that the group-specific concepts of openness and diversity influence recommendation quality, where openness is defined as across-group diversity (i.e., do users of one group listen to the music of other groups?) and diversity is defined as within-group variability (i.e., how dissimilar is the music listened to by users within groups?). Thus, we are also interested in the correlation between the characteristics of beyond-mainstream music and music listeners with openness and diversity patterns as well as with recommendation quality. Concretely, our work is guided by the following research question:

RQ: What are the characteristics of beyond-mainstream music tracks and music listeners, and how do these characteristics correlate with openness and diversity patterns as well as with recommendation quality?

To address this research question, we create, provide, and analyze a novel dataset called *LFM-BeyMS*, which contains complete listening histories of more than 2,000 beyond-mainstream music listeners mined from the Last.fm music streaming platform. Besides, our dataset is enriched with acoustic features and genres of music tracks. Using this enriched dataset, we identify different types of beyond-mainstream music via unsupervised clustering applied to the acoustic features of music tracks. We then characterize the resulting music clusters using music genres. Then, we assign beyond-mainstream users to the clusters to further divide the beyond-mainstream users into subgroups. We study how the characteristics of these beyond-mainstream subgroups correlate with openness and diversity patterns as well as with recommendation quality measured through prediction accuracy.

Findings and contributions. We identify four clusters of beyond-mainstream music in our dataset: (i) C_{folk}, music with high acousticness such as “folk”, (ii) C_{hard}, high energy music such as “hardrock”, (iii) C_{ambi}, music with high acousticness and high instrumentalness such as “ambient”, and (iv) C_{elec}, music with high energy and high instrumentalness such as “electronica”. By assigning users to these clusters, we get four distinct subgroups of beyond-mainstream music listeners: (i) U_{folk}, (ii) U_{hard}, (iii) U_{ambi}, and (iv) U_{elec}. We also find that these groups differ considerably with respect to the accuracy of recommendations they receive, where group U_{ambi} gets significantly better recommendations than U_{hard}. When relating our results to openness and diversity patterns of the subgroups, we find that U_{ambi} is the most open but least diverse group, while U_{hard} is the least open but most diverse group. This is in line with related research [460], which has shown that openness is stronger correlated with accurate recommendations than diversity. This means that users are more likely to accept recommendations from different groups (i.e., openness) rather than varied within a group (i.e., diversity).

Summed up, our contributions are:

- We identify more than 2,000 beyond-mainstream music listeners on the Last.fm platform and enrich their listening profiles with acoustic features and genres of music tracks listened to (Sections 9.3.1–9.3.4).

- We validate related research by showing that beyond-mainstream music listeners receive a significantly lower recommendation accuracy than mainstream music listeners (Section 9.3.5).

- We identify four clusters of beyond-mainstream music using unsupervised clustering and characterize them with respect to acoustic features and music genres (Section 9.4.1).

- We define four subgroups of beyond-mainstream music listeners by assigning users to the music clusters and discuss the relationship between openness, diversity, and recommendation quality of these groups (Section 9.4.2).

- To foster reproducibility of our research, we make available our novel *LFM-BeyMS* dataset via Zenodo[2] and the entire Python-based implementation of our analyses via Github[3].

We believe that our findings provide useful insights for creating user models and recommendation algorithms that better serve beyond-mainstream music listeners.

9.2. Related Work

We identify three strands of research that are relevant to our work: (i) modeling of music preferences, (ii) long-tail recommendations, and (iii) popularity bias in music recommender systems.

[2] https://doi.org/10.5281/zenodo.3784764

[3] https://github.com/pmuellner/supporttheunderground

Modeling of music preferences. A multitude of factors [423] influences musical tastes and musical preferences of users. Characteristics of music listeners and music preferences have been studied in various research domains [177], ranging from music sociology [13] and psychology [118] to music information retrieval and music recommender systems [418]. Studies on music listening behavior showed that personal traits and long-term music preferences are correlated as people tend to prefer music styles that align with their personalities [273, 376]. Furthermore, related work found a relationship between music and motivation [240], music and emotion [226, 227, 492, 515] or both personality and emotion [148]. Openness, a personality trait from the Five Factor Model [163], has also been shown to positively influence a user's preference for music recommendations [460]. Specifically, the authors of [460] found that people tend to prefer recommendations from different kinds of music (i.e., openness) rather than varied within a specific kind of music (i.e., diversity). Others showed that familiarity has a positive influence on music preferences [354, 426] and that music preferences may change over time [323]. Another strand of research on modeling users' music preferences leverages content features, e.g., acoustic features. It has been shown that the distribution of acoustic features of a user's preferred genre substantially influences the user's choice of music within other genres [36]. Also, acoustic features have been utilized to model users' preferences under different contextual conditions, in order to refine recommendation quality [167]. Based on tracks' acoustic features, the authors of [508] identified several types of music, and subsequently modeled each user by linearly combining the acoustic features of the music types. In contrast to these works, we focus on using acoustic features of music tracks for modeling and clustering beyond-mainstream music. Additionally, we link these beyond-mainstream music clusters to music genres and users in our Last.fm data sample.

Long-tail recommendations. Related research [79, 470] has found that individual music consumption is biased towards popular music and that usage data for less popular music is scarce. Due to the scarcity problem, items with no or few ratings (i.e., long-tail items) have little chance of being recommended [80]. As a consequence, users that particularly favor items with few ratings or interactions are less likely to be recommended those items that they like [82]. That is problematic because many users, from time to time, prefer niche music [162]. Therefore, such users are not well served as a result of their preference for less popular items. That has been attributed to *popularity bias*, which corresponds to over-representation of popular items in the recommendation lists [65, 138, 211]. Abdollahpouri et al. [4] studied popularity bias in a dataset of movies (i.e., the MovieLens 1M dataset [181]) from the user perspective. Their study showed that commonly used recommendation techniques tend to deliver worse recommendations to users who prefer less popular movies. In our work [279], we found evidence for popularity bias in a Last.fm dataset and showed that traditional personalized recommendation algorithms such as collaborative filtering deliver worse recommendations for consumers of niche music. In the present work, we aim to gain a deeper understanding of the behavior and preferences of this beyond-mainstreaminess user group. Thus, in contrast to existing works in long-tail recommendations, we focus on the user rather than the item perspective.

Popularity bias in music recommender systems. Music recommender systems [418] are crucial tools in online streaming services such as Last.fm, Pandora, or Spotify. They help users find music that is tailored to their preferences. The basis of music recommender systems are user models derived from users' listening behavior, user properties such as personality (e.g., [95]), content features of music, or hybrid combinations of both, e.g., [15, 123, 234, 508]. As discussed earlier, due to insufficient amounts of usage data for less popular items, many music recommendation algorithms do not provide useful recommendations for consumers of less popular and niche items. As a remedy, in [277], an approach is suggested that divides music consumers into experts and novices according to their long tail distribution in their playlists. These experts are then converted to nodes with bidirectional links connecting all the experts. These links are created to perform link analysis on the graph and to assign fine-grained weights to songs. The presented approach helps add music from the long-tail into the recommendation list. In our previous research [281], we use a framework [266] that employs insights from human memory theory to design a music recommendation algorithm that provides more accurate recommendations than collaborative filtering-based approaches for three groups of users, i.e., low-mainstream, medium-mainstream and high-mainstream users. While the awareness of popularity bias in music recommender systems increases (e.g., [38]), the characteristics of music consumers whose preferences lie beyond popular, mainstream music are still not well understood. In the present work, we shed light on the characteristics of such beyond-mainstream music consumers and relate them to openness and diversity patterns as well as recommendation quality. With this, we aim to provide useful insights for creating novel music recommendation models that mitigate popularity bias.

9.3. Preliminaries

We investigate the characteristics of beyond-mainstream music listeners in a dataset mined from Last.fm, a popular music streaming platform. We characterize the tracks in our dataset with acoustic features. Besides, we compare the recommendation accuracy of beyond-mainstream music listeners with the one of mainstream music listeners to motivate our subsequent analysis of the characteristics of beyond-mainstream music listeners.

9.3.1. Acoustic Music Features

For our analyses, we characterize music tracks using acoustic features that describe the content of a given track. Following the lines of, e.g., [21, 314, 362, 508], we rely on acoustic features provided by the Spotify API as a compact characterization of tracks[4]. The following eight features are extracted from the audio signal of a track:

[4] https://developer.spotify.com/web-api/get-several-audio-features/

Danceability captures how suitable a track is for dancing and is computed based "on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity".

Energy describes the perceived intensity and activity of a track and is based on the dynamic range, perceived loudness, timbre, onset rate, and general entropy of a track.

Speechiness captures the presence of spoken words in a track. High speechiness values indicate a high degree of spoken words (e.g., an audiobook), whereas medium values indicate tracks with both music and speech (e.g., rap music). Low values represent typical music tracks.

Acousticness measures the probability that the given track only contains acoustic instruments.

Instrumentalness quantifies the probability that a track contains no vocals, i.e., the track is instrumental.

Tempo measures the rate of the track's beat in beats per minute.

Valence describes the "emotional positiveness" conveyed by a track (i.e., cheerful and euphoric tracks reach high valence values).

Liveness measures the probability that a track was performed live, i.e., whether an audience is present in the recording.

9.3.2. Enriched Dataset of Music Listening Events

To study characteristics of beyond-mainstream users and their listening preferences, we create a novel dataset called *LFM-BeyMS* that contains the required information for such analyses. We base our work on a dataset gathered from the Last.fm music platform, which we considerably enrich with the music tracks' acoustic features (see Section 9.3.1) [512]. Additionally, we combine this data with mainstreaminess information of Last.fm users (see Section 9.3.3) as well as music genre information to identify beyond-mainstream listeners and music (see Section 9.3.4).

An overview of our new *LFM-BeyMS* dataset and its data sources is depicted in Figure 9.2. As shown, the starting point for our new dataset is the publicly available *LFM-1b* dataset[5] of music listening information shared by users of the online music platform Last.fm [408]. *LFM-1b* contains listening histories of 120,322 users; their listening records (or "listening events") have been created between January 2005 and August 2014.

[5] http://www.cp.jku.at/datasets/LFM-1b/

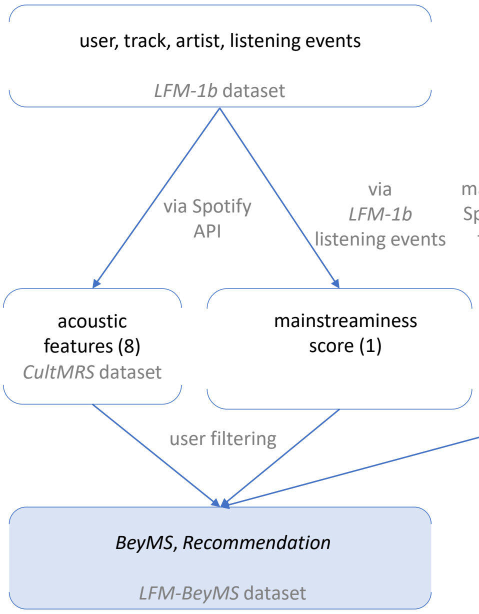


Figure 9.2.: Overview of our new *LFM-BeyMS* dataset and its data sources. We depict the different features, their origin, and relation, and show the feature groups with the number of contained features in brackets. *LFM-BeyMS* contains *BeyMS*, i.e., data to study the beyond-mainstream user group, and *Recommendation*, i.e., data to conduct recommendation experiments of beyond-mainstream and mainstream music listeners.

They sum up to over 1.1 billion listening events (LEs), where each LE is described by an (anonymous) user identifier, the artist name, the album name, the track name, and the timestamp of the listening event. Also, the *LFM-1b* dataset includes demographics of some users (i.e., country, age, and gender).

To enrich the *LFM-1b* dataset to suit our task, we utilize our previously created *CultMRS* music recommendation dataset [503]. This dataset contains 55,191 users, who have listened to a total of 26,022,625 distinct tracks, captured by a total of 807,890,921 listening events [512].

To further enrich the dataset with music acoustic features, we gather the acoustic features described in Section 9.3.1 for the tracks remaining in the dataset after the filtering described above. To this end, we rely on the Spotify API to gather content-based acoustic features for each track. Particularly, we search tracks using the <track, artist, album> triples extracted from the *LFM-1b* dataset using the Spotify search API[6] to gather the Spotify track URI of each track by using all three parts of the triple in a conjunctive query. In total, this allowed gathering 4,326,809 Spotify URIs. For the remainder of the tracks, we were not able to retrieve a URI. We attribute this to two factors: either the

[6] https://developer.spotify.com/web-api/search-item/

Item	*CultMRS* [503]	*LFM-BeyMS* (our novel dataset)	
		BeyMS	*Recommendation*
Users	55,149	2,074	4,148
Tracks	3,478,399	157,444	1,084,922
Artists	337,840	14,922	110,898
Listening Events (LEs)	394,944,868	4,916,174	16,687,363
Min. LEs per user	1	3	9
Q_1 LEs per user	1,442	1,254	2,604
Median LEs per user	5,667	2,048	3,766
Q_3 LEs per user	9,738	3,239	5,252
Max. LEs per user	399,210	10,536	11,177
Avg. LEs per user	7,161.41 (± 10,326.91)	2,371.526 (± 1,520.629)	4,022.990 (± 1,898.060)

Table 9.1.: Descriptive statistics of the *CultMRS* dataset and our novel *LFM-BeyMS* dataset. *CultMRS* comprises acoustic features of tracks. *LFM-BeyMS* is based on *CultMRS* and consists of *BeyMS* and *Recommendation*. Our analyses of beyond-mainstream music listeners utilize *BeyMS* and our recommendation experiments utilize *Recommendation*, which includes listening events of both users with beyond-mainstream and mainstream music taste.

searched track is not provided by Spotify or the track, artist, and album information cannot be matched to a Spotify track unambiguously. Subsequently, we use the obtained track URI to query the acoustic features API, which returns the acoustic features of a given track (cf. Section 9.3.1). In a subsequent cleaning step, we remove all tracks for which the Spotify API did not provide the full set of acoustic features.

That procedure provides us with a set of 3,478,399 unique tracks and their acoustic features. Within the LFM-1b dataset, this amounts to 13.36% of the distinct tracks. Overall, these account for as much as 48.89% of all listening events (i.e., the tracks listened to by users) of the LFM-1b dataset. The resulting dataset, now enriched by acoustic music descriptors, comprises a total of approximately 394 million listening events of 55,149 users. In Table 9.1 (column "*CultMRS*"), we provide further descriptive statistics of the *CultMRS* dataset. We refine this dataset to create our new *LFM-BeyMS* dataset (column "*BeyMS* in Table 9.1), which consists of *BeyMS*, i.e., data to study the characteristics of beyond-mainstream music listeners, and *Recommendation*, i.e., data to conduct recommendation experiments of beyond-mainstream and mainstream music listeners.

9.3.3. Identifying Beyond-Mainstream Music Listeners

To identify beyond-mainstream music listeners, for each user, we compute a mainstreaminess score, which is generally defined as the overlap between a user's individual listening history and the aggregated listening history of all Last.fm users in the dataset. In this vein, the mainstreaminess score reflects a user's inclination to music listened to by the

Last.fm mainstream listeners (i.e., the “average” Last.fm listener in the dataset). In [39], several measures of user mainstreaminess are defined. Out of these, we choose the *M-global-R-APC* definition since it yielded good results in context-based music recommendation experiments for the *LFM-1b* dataset, as evidenced in [39]. The *M-global-R-APC* measure approximates a user’s mainstreaminess score by computing Kendall’s τ [238] rank correlation between the user’s vector of artist play counts and the global vector of artist play counts (aggregated over all users in the dataset). This definition also explains the name of the measure, where “M” refers to mainstreaminess, “global” indicates the global perspective, “R” stands for rank correlation, and “APC” refers to artist play counts.

Next, we describe how we identify our beyond-mainstream users via filtering the users by the number of listening events (see Figure 9.3 and Section 9.3.3) and by mainstreaminess scores (see Figure 9.4 and Section 9.3.3).

Filtering Users by the Number of Listening Events

For our study, we select the users so that listeners of different levels of “listening activity” are equally represented. We conduct a Gaussian kernel density estimation (KDE) [431] on the distribution of listening events over users to estimate the continuous probability density function (PDF) [112]. However, KDE estimates the PDF via discrete bins and hence, it is necessary to approximate the gradient via the principle of finite differences. The gradient of the PDF helps us identifying regions of increasing or decreasing probability.

Figure 9.3 shows that two large subsets of users exist that exhibit either very few or an abundance of listening events. For our analysis, we consider only users who are not in one of the subsets as mentioned earlier. On the one hand, we exclude users with too little data available for studying their listening behavior; and on the other hand, we exclude so-called power listeners who might bias our analyses. Furthermore, such users with a very high number of listening events are often radio stations, which do not contribute reliable data to our investigations.

Hence, we define lower and upper bounds regarding the number of users’ listening events to include in our study, such that the rate of change in terms of the number of listening events is minimal and stable within these boundaries. That requires the gradient of the region within the lower and upper bound to be near zero (i.e., $\pm 10^{-6}$). By computing the second-order accurate central differences [369], we obtain an approximation of the gradient and find the longest cohesive region fulfilling the requirements between a lower bound of 4,688 and an upper bound of 14,787 listening events per user, which leads to 12,814 users.

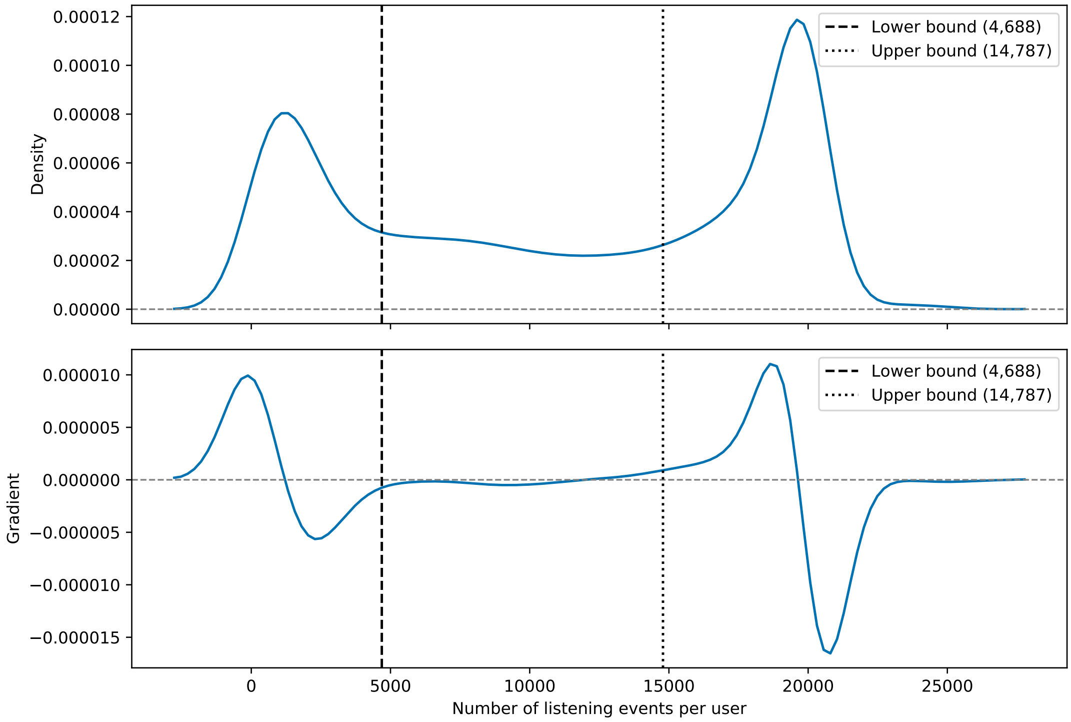


Figure 9.3.: Distribution of listening events in our set of Last.fm users. We set the lower and upper bound marked as dashed and dotted lines, respectively based on the gradient, which results in 12,814 users with a similar number of listening events.

Filtering Users by Mainstreaminess Scores

Figure 9.4 illustrates the mainstreaminess distribution of the 12,814 users that we have extracted based on the number of listening events. Here, mainstreaminess is defined according to the *M-global-R-APC* definition taken from [39] (explained in Section 9.3.3). By setting an appropriate upper bound, we aim to exclude mainstream music listeners. In other words, we aim to set the upper bound to the beginning of the distribution's bulk, which is motivated as follows: Firstly, the first inflection point (i.e., maximal gradient) of a Gaussian distribution is found at $\mathbb{E}[X] - std(X)$, where $\mathbb{E}[X]$ is the expectation, and $std(X)$ is the standard deviation of the Gaussian random variable X. Secondly, the first inflection point of a Gaussian distribution is equivalent to the 15.9-percentile. By setting the mainstreaminess threshold to this point, we intend to omit the majority of users and hence, only consider the 15.9% of users with the lowest mainstreaminess scores. Utilizing this upper bound on the mainstreaminess score, we obtain a set of 2,074 beyond-mainstream users. Furthermore, the Gaussian assumption can be strengthened by the observation that the 2,074 beyond-mainstream users represent 16.19% of users. In the remainder of this paper, we refer to this set of beyond-mainstream music listeners as *BeyMS*.

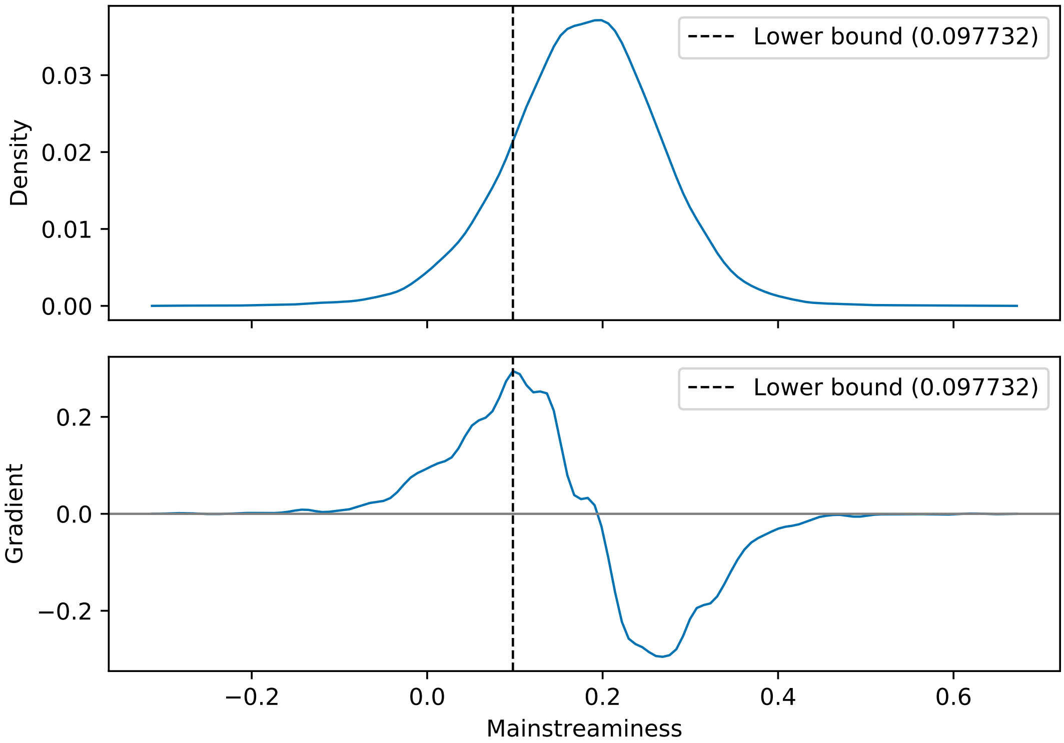


Figure 9.4.: Mainstreaminess distribution of the 12,814 users illustrated in Figure 9.3. Based on the maximum gradient, we select an upper bound of 0.097732 to identify the 2,074 beyond-mainstream users of the *BeyMS* user group.

9.3.4. Identifying Beyond-Mainstream Music

We aim to study beyond-mainstream listeners in terms of their music taste. We characterize music via its acoustic features, as described in Section 9.3.1, and also investigate genres as an alternative way to describe a music track via conventional categories. As the *LFM-1b* dataset does not contain genre annotations of tracks and the Spotify API only provides genres on artist level[7], we leverage the tags assigned to each track by Last.fm users to identify genre annotations. To obtain these tags, we use the respective Last.fm API endpoint[8]. After having fetched the tags for each track, we de-capitalize them and remove all non-alpha-numeric characters. Since not all tags used by Last.fm users correspond to actual music genres (e.g., the "seenlive" tag is used to indicate that a user has seen an artist performing this track live), we use a fine-grained music genre taxonomy consisting of 3,034 genres that are also utilized by Spotify, which we gather from the

[7] https://developer.spotify.com/documentation/web-api/reference-beta/#endpoint-get-an-artist

[8] https://www.last.fm/api/show/track.getTopTags

EveryNoise service (2019-07-24)[9]. Specifically, for each track listened to by any of our *BeyMS* users, we remove all tags that are not part of the EveryNoise genre taxonomy, using a case-insensitive matching approach.

We note that Last.fm users tend to assign very general genre tags to a large number of tracks, such as "pop" or "rock". To remove these coarse-grained genres and to identify fine-grained beyond-mainstream music genres, we calculate the inverse document frequency (IDF) [440] metric of our genre-track distribution by treating genres as terms and tracks as documents, i.e., $IDF(g) = \log_{10} \frac{|T|}{|\{t \in T \text{ with } g \in G_t\}|}$. More precisely, the IDF-score of genre g is determined by relating the number of all tracks $|T|$ to the number of tracks annotated with genre g where $|G_t|$ is the set of genres assigned to track t. This way, a coarse-grained genre receives a small IDF-score, while a fine-grained genre receives a high IDF-score. Figure 9.5 shows the IDF-score distribution of the top-100 genres in ascending order (i.e., from coarse-grained to fine-grained). Here, we identify two groups of genres, where the first group consists of 6 genres with small IDF-scores, and the second group consists of 94 genres with high IDF-scores. The visual inspection of Figure 9.5 indicates that the lower bound of 0.90 serves as a discriminant between these two groups of coarse-grained and fine-grained genres. Consequently, we remove the 6 coarse-grained genres (i.e., "rock", "pop", "electronic", "metal", "alternativerock", "indierock") from the genre assignments of our tracks, which leads to 157,444 out of 799,659 tracks listened to by *BeyMS* users with at least one remaining genre. In total, these tracks are annotated with 1,418 unique genre identifiers.

We are aware of the fact to our track filtering procedure leads to incomplete listening profiles of users. Since we rely on genres to describe beyond-mainstream music, these filtering steps are necessary for our study. To ensure that the *BeyMS* users' reduced listening profiles are still representative of their music preferences, we further investigate the consequences of the filtering procedure. Here, we find that a user's listening history (i.e., the entirety of a user's listening events) is reduced to 61% on average. However, we also find that there are only 62 of the 2,074 *BeyMS* users, for whom the listening history is reduced to less than 20%. For these users most affected by the filtering, we compare the acoustic feature distributions of their listened tracks before and after the filtering steps, and find that filtering only marginally affects the acoustic feature distributions (i.e., average change in mean $= 0.0098 \pm 0.0148$). This means that the acoustic feature distribution contained in the user's profile is highly robust against the filtering. The statistics of *BeyMS* are summarized in column "*BeyMS*" in Table 9.1.

9.3.5. Recommendations for Beyond-Mainstream Music Listeners

In order to compare the recommendation accuracy of recommendations received by the users of our *BeyMS* group and by mainstream users, we construct a dataset consisting of *BeyMS*'s listening events and the listening events of an equally-sized group of mainstream

[9] http://everynoise.com/

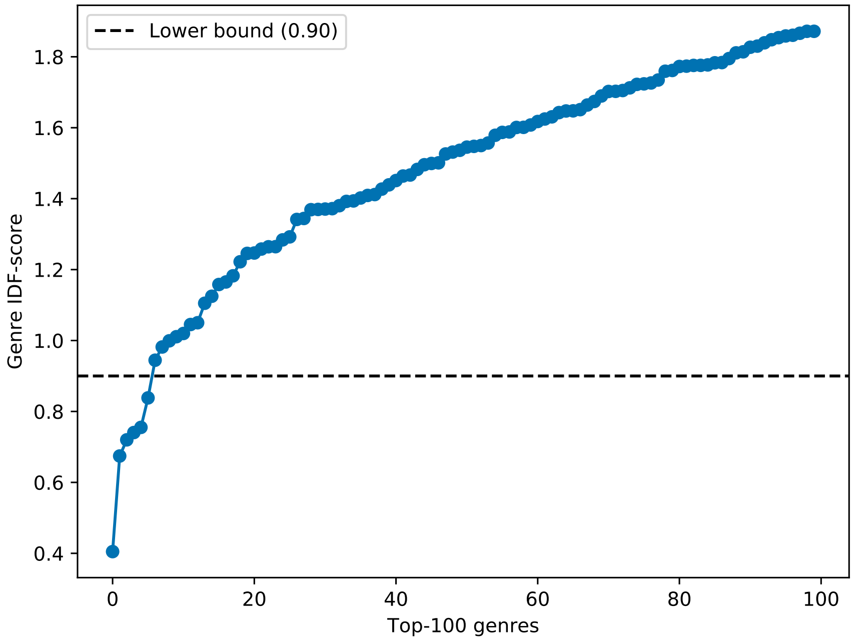


Figure 9.5.: IDF-score distribution of the top-100 genres in ascending order (i.e., from coarse-grained to fine-grained). The 6 coarse-grained genres below the lower bound of 0.90 are removed from the genre assignments, i.e., "rock", "pop", "electronic", "metal", "alternativerock", "indierock".

users. Therefore, we define the *MS* user group as 2,074 (i.e., the size of our *BeyMS* group) randomly-chosen users with a mainstreaminess score that is higher than the upper bound for low mainstreaminess, identified in Figure 9.4. Furthermore, the *MS* users are also in between the lower and upper bounds for listening events identified in Figure 9.3. As shown in Table 9.1 (column "*Recommendation*"), the dataset used for the evaluation of recommendations contains data of 4,148 distinct *BeyMS* and *MS* users, 1,084,922 distinct tracks, and 16,687,363 listening events.

We use the Python-based open-source recommendation library Surprise[10] to compute and evaluate recommendations. One advantage of using Surprise is that it provides built-in recommendation algorithms as well as a standardized evaluation pipeline, which enhances the reproducibility of our research. Since Surprise is focused on rating prediction, we formulate our music recommendation scenario also as a rating prediction problem, in which we predict the preference of a target user u for a target track t. As done in [409], we model the preference of t for u by scaling the play count (i.e., number of listening events) of t by u to a range of [1; 1,000] using min-max normalization. We perform this normalization on the individual user level to ensure that all users share the same

[10] http://surpriselib.com/

User group	UserItemAvg	UserKNN	UserKNNAvg	NMF
BeyMS	63.4608***	71.6694***	67.5770***	**57.7703*****
MS	61.2562	68.4894	63.3985	**54.8182**
Overall	62.2315	69.8962	65.2469	**56.2492**

Table 9.2.: Mean absolute error (MAE) results for the two user groups *MS* and *BeyMS* of different mainstreaminess and a selection of standard recommendation algorithms. A one-tailed Mann-Whitney-U test ($\alpha = .0001$) provides significant evidence, indicated by ***, that all algorithms perform worse on *BeyMS* than on *MS* in terms of MAE. Furthermore, NMF (as shown in bold) outperforms the other three approaches UserItemAvg, UserKNN and UserKNNAvg.

preference value ranges. Thus, with this method, we ensure that each user's most listened track has a preference value of 1,000, while their least listened track has a preference value of 1. To ensure that this min-max normalization procedure does not disrupt the play count distribution of our users, we compare the original play count distribution with the normalized distribution and find that both distributions are strongly right-skewed. Specifically, we find very similar distributions for large amounts of our play count data.

We utilize a selection of Suprise's built-in recommendation methods consisting of one baseline approach (i.e., UserItemAvg), two neighborhood-based approaches (i.e., UserKNN and UserKNNAvg), and one matrix factorization-based approach (i.e., NMF). Specifically, UserItemAvg predicts the average play count in the dataset by also accounting for deviations of u and t, for example, if a user u tends to have more listening events than the average Last.fm user [259]. UserKNN [190] is a user-based collaborative filtering approach and is calculated using $k = 40$ nearest neighbors and the cosine similarity metric, which are the default settings of Surprise. UserKNNAvg is an extension of UserKNN [190] that also takes the average rating of target user u into account. Finally, NMF, i.e., non-negative matrix factorization [302], is calculated using 15 latent factors, which is the default parameter in the Surprise library. As shown in our previous work [279], NMF is also capable of recommending non-popular items from the long tail and should therefore especially be of interest for our beyond-mainstream recommendation setting.

We use Surprise's default parameters and refrain from performing any hyperparameter tuning since we are only interested in assessing (relative) performance differences between the two user groups *BeyMS* and *MS*, and not in outperforming any state-of-the-art algorithm. This is also the reason why we focus on traditional algorithms instead of investigating the most recent deep learning architectures, which would also require a much higher computational effort.

The resulting mean absolute error (MAE) results can be observed in Table 9.2 (and correspond to the ones already shown in Figure 9.1). We favor MAE over the commonly used root mean squared error (RMSE) due to several pitfalls, especially regarding the comparison of groups with different numbers of observations [485]. Here, we perform 5-fold cross-validation leading to 5 different 80/20 train-test splits and average the MAE over the 5 folds. NMF clearly outperforms UserItemAvg as well as the two neighborhood-based methods (i.e., UserKNN and UserKNNAvg) both for the two user groups (see rows "*BeyMS*" and "*MS*") separately and overall without distinguishing between the user groups (see row "Overall"). Additionally, we conduct a one-tailed Mann-Whitney-U test ($\alpha = .0001$), where we define the null-hypothesis as the MAE for *MS* being larger than or equal to the MAE for *BeyMS*. Results marked with *** indicate that the null-hypothesis was rejected for every fold. Thus, all algorithms (including NMF) provide a significantly larger error for *BeyMS* than for *MS*. In other words, recommendation quality is significantly better for users with mainstream taste than for users who prefer beyond-mainstream music across all recommendation approaches.

These initial results underpin the need to study the characteristics of the *BeyMS* user group that receives worse recommendations. The corresponding experiments are presented in the next section.

9.4. Characteristics of Beyond-Mainstream Music and Listeners

We identify the types of beyond-mainstream music using unsupervised clustering and characterize these types with respect to acoustic features and music genres. Besides, we detect subgroups of beyond-mainstream music listeners by assigning users to these clusters and evaluate the recommendation quality obtained for these subgroups. Finally, we discuss the recommendation quality with respect to openness and diversity. For this, we relate to the definitions given by [460]:

Openness is the across-groups diversity (or categorical diversity) and describes if users of one group also listen to the music of other groups.

Diversity is the within-groups diversity (or thematic diversity) and describes the dissimilarity of music listened to by users within groups.

Based on the findings of [460], we would expect that subgroups with high openness should receive more accurate recommendations than subgroups with high diversity.

9.4.1. Clustering and Characterizing Beyond-Mainstream Music

To study the different types of music listened to by the users in our *BeyMS* group, we conduct a cluster analysis. Specifically, we cluster the 157,444 tracks listened to by *BeyMS* users, where each track is described by the eight acoustic features danceabil-

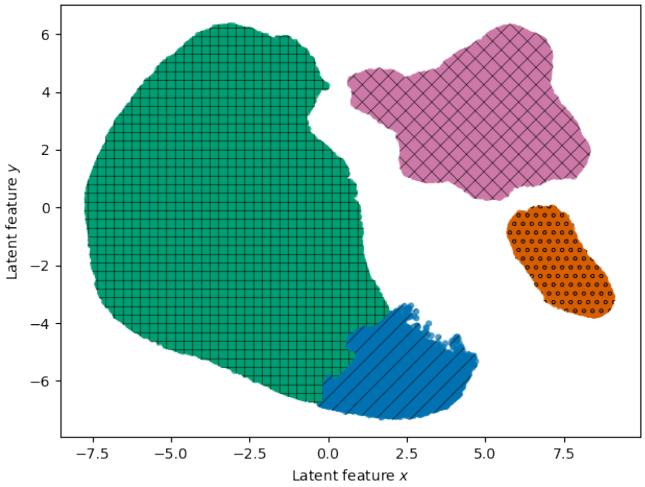


Figure 9.6.: Music clustering results obtained with HDBSCAN* and UMAP for the 2-dimensional mapping. The outputs are four clusters with the following cluster sizes: 12,148 (blue, hatch: /), 92,798 (green, hatch: +), 7,629 (orange, hatch: o) and 30,379 (pink, hatch: x) tracks. 14,490 of our 157,444 *BeyMS* tracks have not been assigned to a cluster.

ity, energy, speechiness, acousticness, instrumentalness, tempo, valence, and liveness (see Section 9.3.1). We scale the value ranges of these features to [0, 1] using min-max normalization. The use of latent representations of musical elements such as tracks was shown to be efficient in the area of music information retrieval [92, 280, 508]. Furthermore, for visually analyzing the obtained music clusters and decreasing computation time, we favor a reduction of dimensionality to two dimensions.

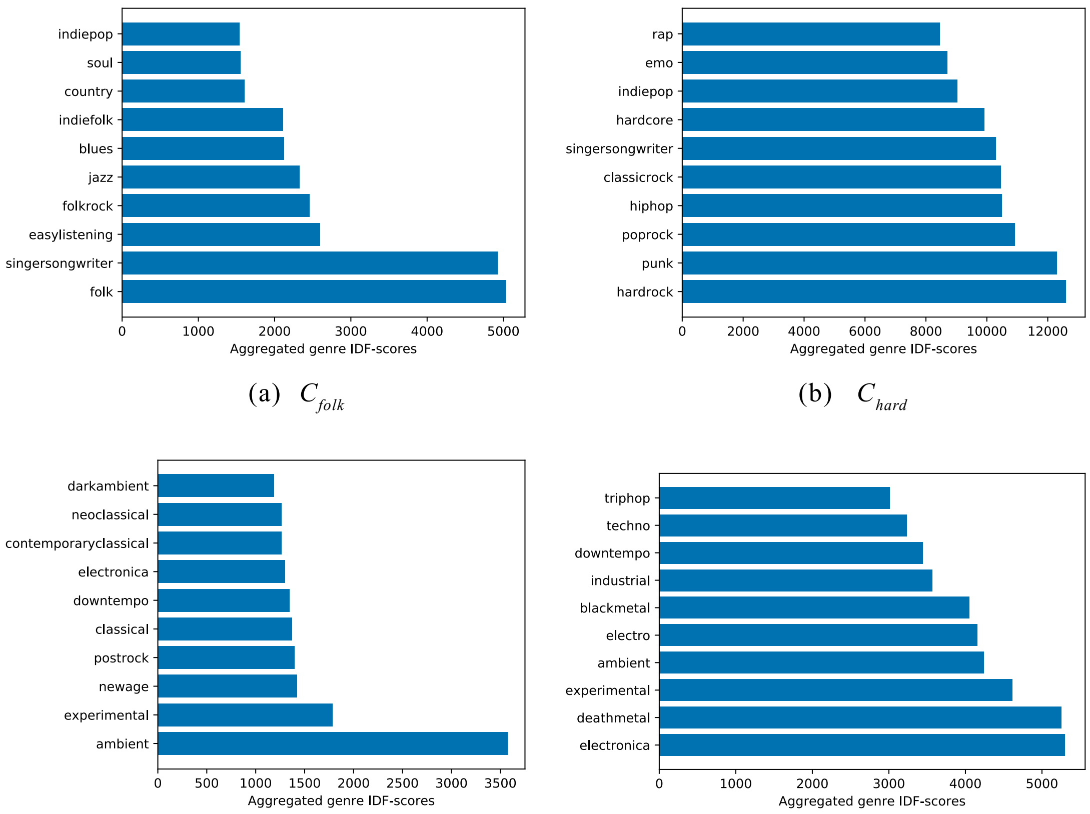
We conduct experiments with a broad body of dimensionality reduction methods, i.e., linear and nonlinear principal component analysis (PCA) [462], locally linear embedding [389], multidimensional scaling [268], Isomap [457], spectral embedding [332], t-distributed stochastic neighbor embedding (t-SNE) [471] and uniform manifold approximation and projection (UMAP) [311]. We visually inspected the 2-dimensional feature spaces created by these methods with regards to the clustering quality, and we obtained the visually most homogeneous results with UMAP. Moreover, UMAP has already been successfully used in the music domain [508] and thus, we use it for the remainder of our experiments. Specifically, we utilize the open-source implementation of UMAP [311], which requires four parameters: (i) the distance metric M in the input space, (ii) the number of latent dimensions D, (iii) the minimum distance of points in the latent space d_{min}, and (iv) the number of neighbors of a point N. Based on experimentation and related literature (e.g., [311]), we set the distance metric M to the Euclidean distance, the number of latent dimensions D to 2, the distance d_{min} to 0.1 and the number of neighbors N to 15.

In a next step, we perform clustering on the dimensionality-reduced acoustic features of tracks. Again, we conduct experiments with various clustering methods, i.e., DBSCAN [489], K-Means [51], Gaussian mixture models [380], affinity propagation [153], spectral clustering [432], hierarchical agglomerative clustering [327], OPTICS [26] and HDBSCAN* [309]. Here, we obtain the best results with respect to cluster cohesion and separation using HDBSCAN*. Furthermore, HDBSCAN* was also already used by related work to cluster music items [499]. We employ the open-source implementation of HDBSCAN* [310] that requires four parameters: (i) the minimum cluster size s_{min} that defines the minimum size of a group of points to consider a cluster, (ii) the minimum number of samples in the neighborhood of a core point N_{min}, which quantifies how conservative the clustering is, (iii) ε, which enables the recovery of DBSCAN clusters if the s_{min} value is not reached, and (iv) the scaling of the distance α, which is another measure of the clustering's conservativeness. In detail, α scales the distance between two points, which determines whether these points are merged into a cluster. This scaling is used in the construction of HDBSCAN*'s hierarchy of clusterings. Again, we find the best-suited parameters based on experimentation and related literature (e.g., [309]). Specifically, we require each cluster to comprise a sufficiently large number of tracks to increase the level of significance of our subsequent experiments. We expect the existence of very small music clusters and thus, search for the optimal value of the minimal cluster size s_{min} in the search space of $\{1{,}000; 1{,}025; \ldots; 1{,}475; 1{,}500\}$, where we obtain the best results with respect to the within-cluster variance for $s_{min} = 1,375$. Furthermore, tightly packed clusters without any contribution of noise should be favored. In other words, all points within a cluster should be within the neighborhood of at least one core point. Thus, we set the minimal number of samples in the neighborhood $N_{min} = s_{min} = 1,375$. The remaining two parameters are set to their default values, i.e., $\varepsilon = 0$ and $\alpha = 1$.

Figure 9.6 shows the results of the clustering process using HDBSCAN* and UMAP for the 2-dimensional mapping. This process leads to four music clusters. Here, the green cluster (hatch: $+$) is the largest one with 92,798 tracks, followed by the pink cluster (hatch: x) with 30,379 tracks and the blue cluster (hatch: /) with 12,148 tracks. The smallest cluster is the orange one (hatch: o) as it contains 7,629 tracks. The remaining 14,490 of our 157,444 *BeyMS* tracks have not been assigned to a cluster and thus, will not be included in further analyses and interpretations. Next, we describe how we name these clusters based on their music genre distributions.

Genre Distributions

In Figure 9.7, we illustrate the top-10 genres of the four music clusters. For this, we refer to the genre IDF-scores presented in Section 9.3.4 and weight each genre assigned to a track in a cluster with its corresponding IDF-score. For example, if a genre with an IDF-score of 1.4 is assigned to 1,000 tracks in a cluster, it is visualized as an aggregated genre IDF-score of 1,400 in the corresponding plot of Figure 9.7. Based on the genre distributions, we label each cluster according to its top genre.



(a) C_{folk} (b) C_{hard}

(c) C_{ambi} (d) C_{elec}

Figure 9.7.: Top-10 genres of the four music clusters C_1–C_4 according to the aggregated genre IDF-scores. We name the clusters according to the top genre, i.e., (a) blue (hatch: $/$) $\rightarrow C_{folk}$ ("folk"), (b) green (hatch: $+$) $\rightarrow C_{hard}$ ("hardrock"), (c) orange (hatch: o) $\rightarrow C_{ambi}$ ("ambient"), and (d) pink (hatch: x) $\rightarrow C_{elec}$ ("electronica").

With respect to the blue cluster (hatch: $/$) in Plot (a), we find top genres such as "folk" and "singersongwriter", which typically reflect music with high acousticness. In the remainder of this paper, therefore, we refer to this cluster as C_{folk}. The top genres of the green cluster (hatch: $+$) in Plot (b) are typical high energy music genres such as "hardrock", "punk", "poprock", and "hiphop". Based on this, we name this cluster C_{hard}.

For the orange cluster (hatch: o) in Plot (c), we find genres that reflect music with high acousticness and high instrumentalness such as "ambient", "experimental", "newage", and "postrock". As "ambient" clearly dominates the genre distribution for this cluster, we name this cluster C_{ambi}. Similarly to C_{folk}, this cluster contains music with high acousticness; yet, while C_{folk} is characterized by low instrumentalness music, C_{ambi} is

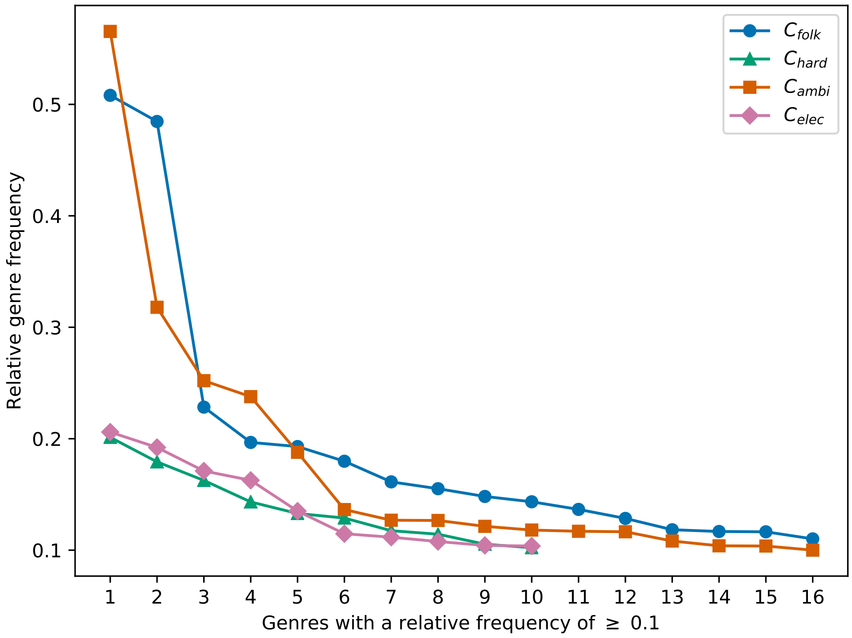


Figure 9.8.: Relative genre frequency distribution of the four music clusters. While there are dominating genres in C_{folk} and C_{ambi}, the genre distribution is more diverse in C_{hard} and C_{elec}.

characterized by a high level of instrumentalness. Finally, Plot (d) shows the genre distribution of the pink cluster (hatch: x) with "electronica" as the top genre, which leads to the name C_{elec} for this cluster.

Thus, both, C_{elec} and C_{hard}, consist of high energy music but in contrast to C_{hard}, C_{elec} also comprise high instrumentalness values. This also makes sense when looking at other top genres of C_{elec} such as "deathmetal" and "blackmetal" where guttural vocal techniques are often mistakenly classified as another type of instrument [500].

To compare the genre distributions among the four music clusters, we illustrate the relative genre frequency distribution of the clusters in Figure 9.8. The relative frequency of a genre g depicts the fraction of listening events of tracks within a cluster c that are annotated with g. Here, we only show genres with a minimum relative genre frequency of 0.1. We see that there are clearly dominating genres in C_{folk} and C_{ambi}, whereas the genre distributions in C_{hard} and C_{elec} are more evenly distributed. When relating this finding to the findings of Figure 9.7, we clearly see that the results correspond to each other: C_{hard} and C_{elec} contain a more diverse genre spectrum (e.g., "hardrock" and "hiphop" are both part of C_{hard}'s top genres) than C_{folk} and C_{ambi} (e.g., in C_{ambi}'s top genres, we find "ambient" and "darkambient").

Acoustic Feature Distributions

To understand the musical content of these four music clusters, we analyze the acoustic feature distributions of the four music clusters using boxplots in Figure 9.9. This visualization does not show any obvious differences with respect to danceability and tempo among the four clusters. For the acoustic features energy, speechiness, acousticness, valence, and liveness, there are similar values for the cluster pairs C_{folk} and C_{ambi}, and C_{hard} and C_{elec}. We observe differences between these two cluster pairs with respect to energy and acousticness. While C_{hard} and C_{elec} provide high energy values and small acousticness values, C_{folk} and C_{ambi} feature small energy values and high acousticness values.

In contrast, for instrumentalness, we see similar values for the cluster pairs C_{folk} and C_{hard} as well as for C_{ambi} and C_{elec}. We observe very high values for C_{ambi} and C_{elec}, and very small values for C_{folk} and C_{hard}. This difference is also visible in Figure 9.6 in the form of the gap between C_{folk} and C_{hard} on the left, and C_{ambi} and C_{elec} on the right.

Summing up, in C_{folk}, we find music with low energy, high acousticness, and low instrumentalness; C_{hard} contains music with high energy, low acousticness, and low instrumentalness; in C_{ambi}, we observe music with low energy, high acousticness, and high instrumentalness; and in C_{elec}, we find high energy, low acousticness, and high instrumentalness. Thus, these findings are in line with the genre distributions presented in Figure 9.7.

9.4.2. Assigning and Studying Beyond-Mainstream Music Listeners

In the next step, we assign the 2,074 *BeyMS* users to the four music clusters to categorize them into four distinct beyond-mainstream subgroups for further analyses.

For each user u, we count the number of listening events $LE_{u,c}$ that u has contributed to the tracks in each cluster c, where $c \in C = \{C_{folk}, C_{hard}, C_{ambi}, C_{elec}\}$. Then, we assign u to the cluster c for which the number of contributed listening events $LE_{u,c}$ is the highest. However, because we have varying cluster sizes, the probability of u listening to a track t of the two larger clusters C_{hard} and C_{elec} is much higher than for the two smaller clusters C_{folk} and C_{ambi}, although C_{folk} and C_{ambi} could be more representative choices for u. Thus, similar to the IDF distribution of genres (see Figure 9.5), we take advantage of the IDF scoring to reduce the influence of the larger clusters and to assign higher weights to the smaller clusters. Specifically, these cluster IDF-scores are given by $IDF(c) = \log_{10} \frac{|T|}{|\{t \in T \text{ with } c_t\}|}$, i.e., by relating the number of all tracks $|T|$ to the number of tracks in cluster c where c_t is the music cluster assigned to track t. That lets us define the user–cluster weight $w_{u,c}$ for user u and cluster c as $w_{u,c} = IDF(c) \cdot LE_{u,c}$.

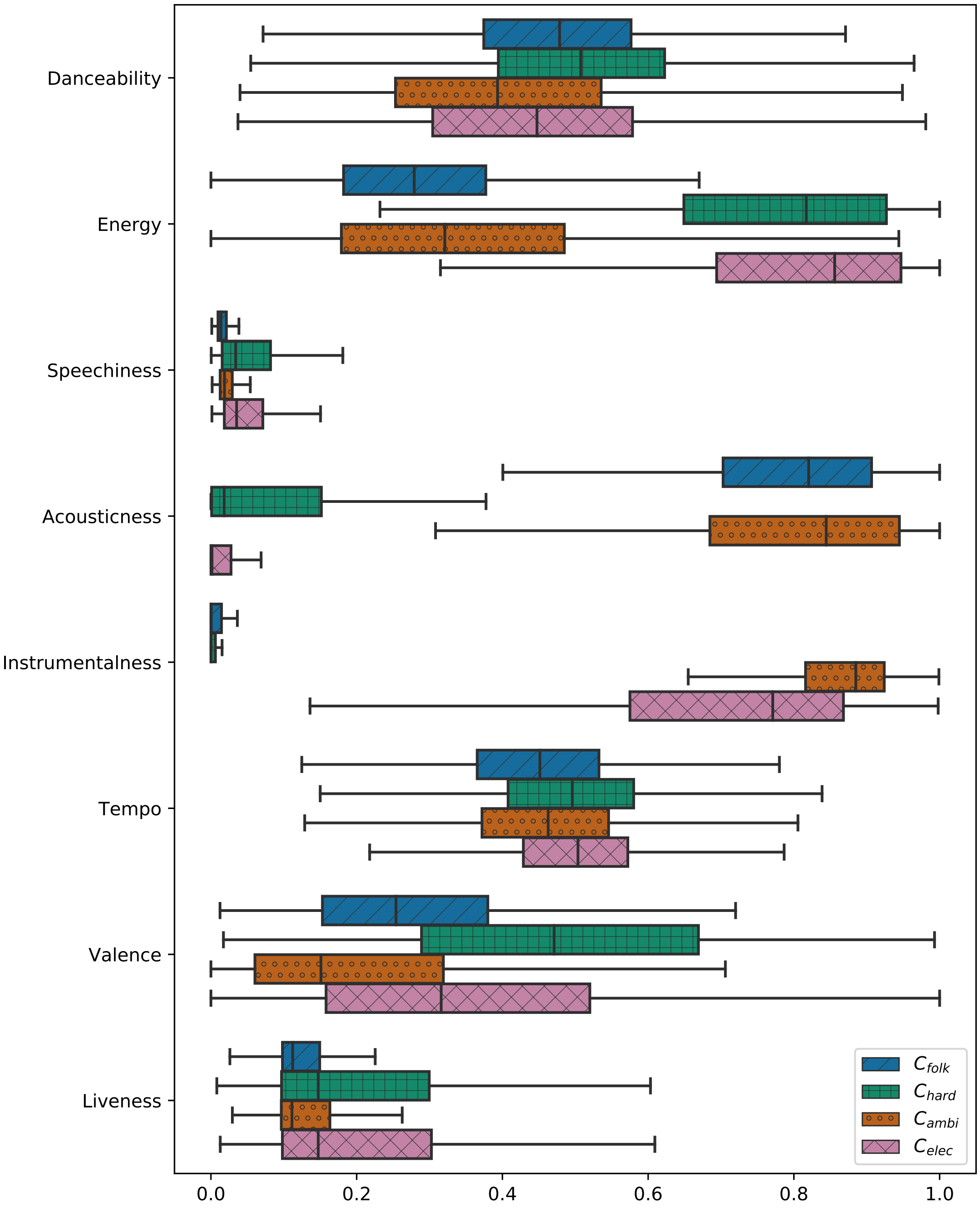


Figure 9.9.: Distribution of the eight acoustic features for the four music clusters. While the clusters do not show obvious differences with respect to danceability and tempo, we find large differences with respect to energy, acousticness and instrumentalness.

Subgroup	$\|U\|$	$\|A\|$	$\|T\|$	$\|LE\|$	$\|G\|$	$\overline{\|LE_u\|}$	$\overline{\|T_u\|}$	$\overline{Age}$ (std.)
U_{folk}	369	9,559	72,663	702,635	811	1,904.160	549.650	27.599 (± 10.369)
U_{hard}	919	11,966	107,952	2,150,246	1,274	2,339.767	557.470	23.867 (± 8.912)
U_{ambi}	143	6,869	39,649	224.327	918	1,568.720	473.308	29.571 (± 14.138)
U_{elec}	642	11,814	105,907	1,416,354	1,005	2,206.159	670.402	24.639 (± 7.886)

Table 9.3.: Descriptive statistics of the four subgroups. Here, $|U|$ is the number of users, $|A|$ is the number of artists, $|T|$ is the number of tracks, $|LE|$ is the number of listening events, $|G|$ is the number of genres, $\overline{|LE_u|}$ is the average number of listening events per user, $\overline{|T_u|}$ is the average number of tracks per user and $\overline{Age}$ is the average age (along with the standard deviation) of users in the group.

Consequently, users are assigned to the highest weighted music cluster and thus, a subgroup U_c for cluster c is given by $U_c = \{u \in U : \arg\max_{c \in C}(w_{u,c})\}$.

Out of the 2,074 $BeyMS$ users, we can assign 2,073 users to these subgroups. Thus, only 1 user listened to tracks not contained in any cluster in Figure 9.6. Similar to the naming scheme of music clusters, we label the subgroups according to the name of their assigned music cluster. Hence, we obtain four subgroups U_{folk}, U_{hard}, U_{ambi}, and U_{elec}.

Table 9.3 provides basic descriptive statistics of these four resulting subgroups. Here, U_{hard} is the largest subgroup with $|U| = 919$ users, followed by U_{elec} with $|U| = 642$ users, U_{folk} with $|U| = 369$ users, and U_{ambi} with $|U| = 143$ users. The differences with respect to the number of users also correspond to the differences regarding the number of artists $|A|$, the number of tracks $|T|$, and the number of listening events $|LE|$ contained in the clusters. In the case of the number of genres $|G|$, this differs slightly because the users in the smaller U_{ambi} cluster listen to more genres (i.e., 918) than the bigger U_{folk} cluster (i.e., 811). This indicates that the users in U_{ambi} listen to a broader set of music than the users in U_{folk}.

Considering the average number of listening events per user (i.e., $\overline{|LE_u|}$) and the average number of tracks per user (i.e., $\overline{|T_u|}$), we see that, while there is little difference between U_{hard} and U_{elec} with respect to $\overline{|LE_u|}$, $\overline{|T_u|}$ is much higher for U_{elec} (i.e., 670.402) than for U_{hard} (i.e., 557.470). This indicates that, although the number of listening events is nearly the same, users of U_{elec} tend to listen to a wider set of tracks than users of U_{hard}. With respect to the average age of the users $\overline{Age}$, we see that the users of U_{folk} and U_{ambi} are the oldest ones, and users of U_{hard} and U_{elec} are the youngest ones. However, it is worth noting that the group with the highest average age (i.e., U_{ambi}) also shows by far the highest standard deviation of age (i.e., 14.138 years).

In Figure 9.10, we show the contribution of each music cluster to each subgroup in the form of a radar plot. For this, we use the user-cluster weights $w_{u,c}$ introduced before and calculate the average weight over all users in cluster c. One consequence of the IDF

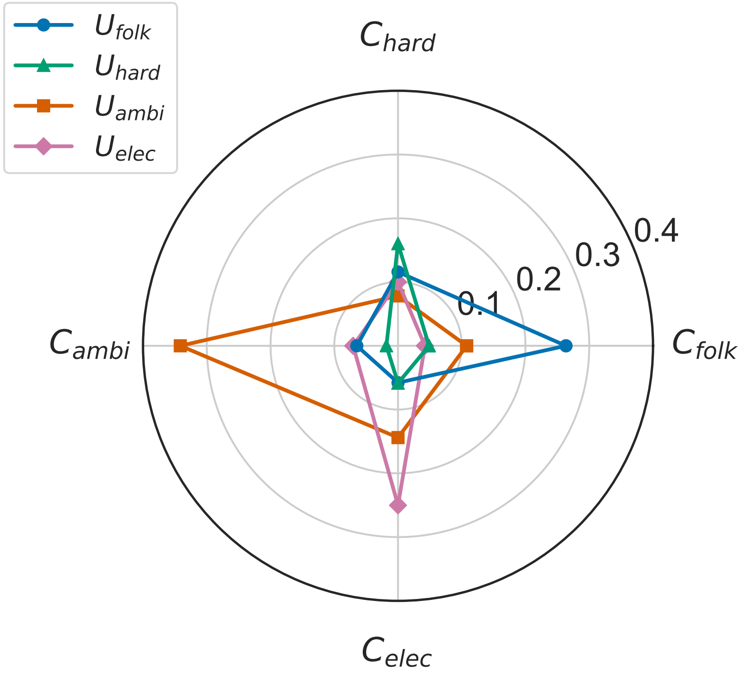


Figure 9.10.: Radar plot illustrating the contribution of each music cluster to a subgroup. While the weight distribution of U_{hard} and U_{elec} is rather narrow, it is more broad in case of U_{folk} and U_{ambi} suggesting that these groups are more open to music outside the own music cluster.

scoring applied to $w_{u,c}$ is that the weight contributions of a user group to the four clusters does not sum up to 1, which eventually influences the interpretation of the values shown in Figure 9.10. However, in return, these values account for the varying cluster sizes and can also be interpreted as preference weights for a user group towards a specific music cluster.

We observe that the weight distribution of the two larger subgroups U_{hard} and U_{elec} is rather narrow, which indicates that these users do not listen to many tracks of other clusters. Contrary to that, the weights of the two smaller subgroups U_{folk} and U_{ambi} are more broadly distributed over the four music clusters. This suggests that users of U_{folk} and U_{ambi} are more open to music outside of their own music cluster than users of U_{hard} and U_{elec}.

Correlation of Music Clusters and Beyond-Mainstream Subgroups

To better understand the correlations and connections between the music clusters and subgroups, we plot the Pearson correlation matrix of the four music clusters as a heatmap

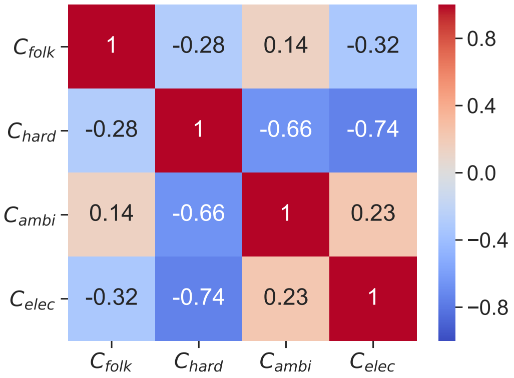


Figure 9.11.: Pearson correlation matrix of the four music clusters. While C_{hard} has solely negative correlations with all other clusters, and thus, listeners of C_{hard} seem to be the most closed subgroup, C_{ambi} has positive correlations with C_{folk} and C_{elec}, and thus, listeners of C_{ambi} seem to be the most open subgroup.

in Figure 9.11. Here, we represent each music cluster c by a 2,073-dimensional vector (i.e., one entry for each user) consisting of the user–cluster weights $w_{u,c}$, introduced before. Each element in the matrix is then calculated using the Pearson correlation measure based on these cluster vectors. For example, if there is a positive correlation between two clusters, we assume that a user who enjoys music from the one cluster likely also enjoys music from the other cluster. This can give us also an indication of the openness of a subgroup for music mainly listened to by other subgroups. Specifically, for C_{folk}, we see a positive correlation between C_{folk} and C_{ambi}, and a negative correlation between C_{folk} and both, C_{hard} as well as C_{elec}. Users listening to the music of C_{hard} seem to represent the most closed subgroup as C_{hard} because it solely has negative correlations with all other clusters, especially with C_{ambi} and C_{elec}. In contrast, users listening to the music of C_{ambi} seem to represent the most open subgroup as C_{ambi} has positive correlations with two other clusters, i.e., C_{folk} and C_{elec}. The fourth cluster, C_{elec}, is negatively correlated with C_{folk} and especially with C_{hard}, and positively correlated with C_{ambi}. These results are also in line with the ones shown in Figure 9.10, in which we identify the users of U_{ambi} as more open music listeners than the ones of U_{hard}.

In order to relate the openness of the subgroups to the diversity of the users within the subgroups, we calculate the average pairwise user similarity using the cosine similarity metric computed on the users' genre distributions, i.e., number of listening events per genre. Figure 9.12 shows the resulting boxplots for the four identified subgroups (i.e., C_{folk}, C_{hard}, C_{ambi}, and C_{elec}). Figure 9.12 shows that users in U_{hard} and U_{elec} have a rather small average pairwise user similarity and, thus, exhibit a more diverse listening behavior, whereas users in U_{folk} and U_{ambi} tend to listen to more similar music genres and, thus, have a narrow listening behavior within the group. Summed up, we find pronounced differences with respect to openness and diversity across the subgroups. Although U_{ambi}

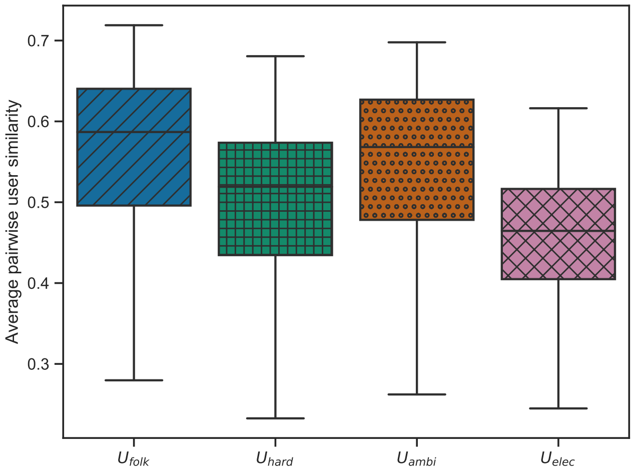


Figure 9.12.: Boxplots showing the average pairwise user similarity of the four subgroups using the cosine similarity calculated on the users' genre distributions. While the users in U_{hard} and U_{elec} exhibit a more diverse listening behavior, users in U_{folk} and U_{ambi} tend to listen to more similar, i.e., less diverse, music genres.

is the most open subgroup (i.e., also listens to music of other subgroups), it is also the least diverse subgroup (i.e., the users within the group listen to very similar music). That observation is in line with what is shown in Figures 9.7, and Figure 9.8. Here, we see that C_{ambi}, i.e., the most tightly connected music cluster to U_{ambi}, contains the dominating genre "ambient" as well as genres that are strongly associated with this dominating genre (e.g., "darkambient"). For U_{hard}, we observe the opposite. While it is the least open subgroup, it is also the most diverse one (e.g., it contains "hardrock" as well as "hiphop" listeners).

Recommendations for Beyond-Mainstream User Subgroups

In Section 9.3.5, we have shown that the recommendation accuracy of four personalized recommendation algorithms is significantly worse for *BeyMS* users than for *MS* users. Now, we extend this analysis and evaluate the recommendation accuracy of these algorithms for the four subgroups (i.e., U_{folk}, U_{hard}, U_{ambi}, and U_{elec}).

Table 9.4 shows our results with respect to the mean absolute error (MAE). Additionally, we analyze these results with respect to statistically significant differences in Table 9.5 by performing ANOVA ($\alpha = .01$) and a subsequent Tukey-HSD test ($\alpha = .05$). Here, we report pairwise differences as significant (marked with **), if both ANOVA and Tukey-HSD were significant across all five folds (see Section 9.3.5 for details on the experimental setup).

Subgroup	UserItemAvg	UserKNN	UserKNNAvg	NMF
U_{folk}	63.2143	70.3049	67.4406	**57.2278**
U_{hard}	65.1464	73.1949	69.2855	**59.6887**
U_{ambi}	60.5558	69.8315	65.5708	**54.2073**
U_{elec}	62.2894	71.0387	66.1499	**56.6209**
BeyMS	63.4608	71.6694	67.5856	**57.7703**
MS	61.2562	68.4894	63.3985	**54.8182**

Table 9.4.: Mean absolute error (MAE) measurements for the four subgroups and four personalized recommendation algorithms. NMF (in bold) outperforms all other algorithms for all subgroups. Among the subgroups, the best accuracy results (i.e., lowest MAE scores) are reached by U_{ambi}, while the worst accuracy results (i.e., highest MAE scores) are reached by U_{hard}. To facilitate comparison, we also show the MAE measurements for the *BeyMS* and *MS* user groups.

We see that among all algorithms, the significantly worst accuracy results (i.e., the highest MAE scores) are achieved for the U_{hard} subgroup. Next, U_{folk}, U_{ambi} and U_{elec} reach significantly better (i.e., lower MAE scores) than U_{hard} for all algorithms. However, there is no statistically significant difference between the recommendation accuracy of U_{folk} and U_{elec}. The overall best accuracy results (i.e., lowest MAE scores) are reached for the U_{ambi} subgroup. These results are also statistically significant when compared with the other subgroups for the NMF algorithm. NMF also gives the overall best accuracy results for all subgroups, which is in line with our results presented in Section 9.3.5 and in our previous work [279].

Furthermore, we find a relationship between openness, diversity, and recommendation quality. Here, U_{hard} is the least open but most diverse subgroup and gets the worst recommendations, while U_{ambi} is the most open but least diverse subgroup and gets the best recommendations. This is in line with the findings of [460], who have shown that users are more likely to accept recommendations from different groups (i.e., openness) rather than varied within a group (i.e., diversity). Thus, we find a relationship between the quality of recommendations provided to beyond-mainstream music listeners and openness as well as diversity patterns of these users.

Finally, in Figure 9.13, we visually compare the MAE scores reached by the best performing approach NMF for the four subgroups. Additionally, we depict the MAE score for *BeyMS* as a black dashed line and the MAE score for *MS* as a grey dashed line. We see that U_{hard} reaches worse results than *BeyMS* while U_{folk} and U_{elec} reach slightly better results than *BeyMS*. Interestingly, U_{ambi} not only reaches better results than *BeyMS* but also better results than *MS*. Although this improvement over *MS* is not statistically significant (according to a one-tailed Mann-Whitney-U test with $\alpha = .0001$), it shows

	UserItemAvg				UserKNN				UserKNNAvg				NMF			
Subgroup	U_{folk}	U_{hard}	U_{ambi}	U_{elec}	U_{folk}	U_{hard}	U_{ambi}	U_{elec}	U_{folk}	U_{hard}	U_{ambi}	U_{elec}	U_{folk}	U_{hard}	U_{ambi}	U_{elec}
U_{folk}		**	**			**				**				**	**	
U_{hard}	**		**	**	**		**	**	**		**	**	**		**	**
U_{ambi}	**	**				**				**			**	**		**
U_{elec}		**				**				**				**	**	

Table 9.5.: Statistically significant differences between pairs of subgroups, as determined by ANOVA ($\alpha = .01$) and a subsequent Tukey-HSD test ($\alpha = .05$).

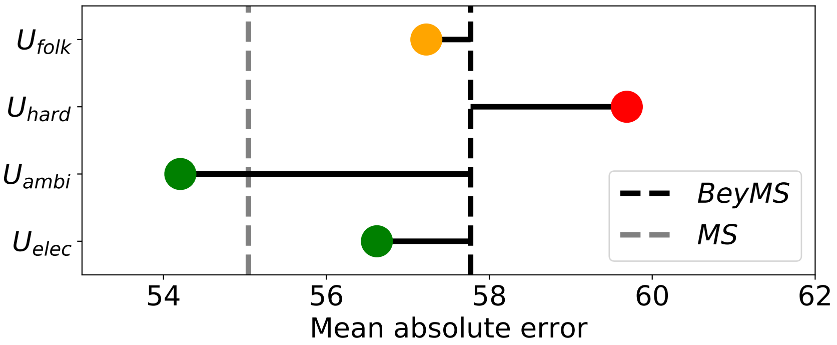


Figure 9.13.: Comparison of the mean absolute error (MAE) scores reached by NMF for the four subgroups with the ones reached by NMF for *BeyMS* (black dashed line) and *MS* (grey dashed line). While specific subgroups (i.e., U_{hard}) are treated in an unfair way by recommendation algorithms, others (i.e., U_{ambi}) are not.

that there is a large variety among *BeyMS* users, where specific subgroups (i.e., U_{hard}) are disadvantaged in terms of recommendation accuracy by recommendation algorithms while others (i.e., U_{ambi}) are not.

9.5. Conclusions and Future Work

In this paper, we shed light on the characteristics of beyond-mainstream music and music listeners. As our first contribution, we identified 2,074 beyond-mainstream music listeners (i.e., *BeyMS*) in the Last.fm platform, and subsequently created a novel dataset called *LFM-BeyMS* based on the listening histories of these users. We further enriched this dataset with (i) acoustic features of music tracks gathered from Spotify, and (ii) genre information of tracks derived from Last.fm tags and matched with the Spotify microgenre taxonomy. Additionally, for reasons of comparability, *LFM-BeyMS* contains data of 2,074 Last.fm users listening to mainstream music. Using this dataset, as our second contribution, we validated related research by showing that beyond-mainstream music listeners receive a significantly lower recommendation accuracy than mainstream music listeners by four standard recommendation algorithms (i.e., UserItemAvg, UserKNN, UserKNNAvg and NMF).

As our third contribution, we applied the clustering algorithm HDBSCAN* on the acoustic features of tracks listened by *BeyMS* and identified four clusters of beyond-mainstream music: (i) C_{folk}, music with high acousticness such as "folk", (ii) C_{hard}, high energy music such as "hardrock", (iii) C_{ambi}, music with high acousticness and instrumentalness such as "ambient", and (iv) C_{elec}, music with high energy and instrumentalness such as "electronica".

As our fourth contribution, we mapped these clusters to our *BeyMS* users, which led to four beyond-mainstream subgroups: (i) U_{folk}, (ii) U_{hard}, (iii) U_{ambi}, and (iv) U_{elec}. We analyzed these subgroups with respect to their openness (i.e., across-groups diversity – do users of one group listen to music of other groups?) and diversity (i.e., within-groups diversity – how dissimilar is the music listened to by users within groups?). Here, we found large differences between U_{hard} and U_{ambi}. Although U_{hard} is the most closed subgroup (i.e., users do not listen to music of other subgroups), it is also the most diverse subgroup (i.e., users listen to a diverse set of genres such as "hardrock" and "hiphop"). For U_{ambi}, we get opposite results: while it is the most open subgroup (i.e., users listen to music of other subgroups as well), it is also the least diverse one (i.e., the users within the group listen to very similar music such as "ambient" and "darkambient"). We related these characteristics of the subgroups to the recommendation quality of the four recommendation algorithms UserItemAvg, UserKNN, UserKNNAvg and NMF. Here, we found that U_{hard} got music recommendations with lowest accuracy, while U_{ambi} got music recommendations with highest accuracy. This is in line with related research [460], which has shown that openness is stronger correlated with accurate recommendations than diversity. U_{ambi} even received better recommendations than the group of mainstream music listeners. This result highlights that there are large differences between the subgroups of beyond-music listeners. Finally, to foster reproducibility of our research, we provide our novel *LFM-BeyMS* dataset via Zenodo as well as our source code via Github.

We believe that our findings provide useful insights for creating user models and recommendation algorithms that better serve beyond-mainstream music listeners. As it was shown in [279], beyond-mainstream music listeners tend to have larger user profile sizes than users interested in mainstream music, which means that they provide a substantial amount of listening interaction data for services such as Last.fm and Spotify. We assume that improving the recommendation quality for this active user group also leads to another effect, namely a more prominent exposure of (long-tail) music artists due to a better-connected recommendation network [271]. We leave such investigations to future work.

Limitations and future work. Despite the merits of this work, we are aware of its limitations. The first limitation we recognize is that our analyses are based on a sample of the Last.fm community. The extent to which their listening behavior is representative of the Last.fm community at large, or similar music streaming communities such as Spotify, needs further investigation.

Next, since we conducted a comparative study of the accuracy of recommender systems algorithms—and were therefore not interested to beat state-of-the-art algorithms—we focused on traditional algorithms (e.g., KNN-based collaborative filtering) instead of investigating the most current deep learning architectures, which would also require a much higher computational effort. Furthermore, an award-winning-paper by Dacrema et al. [144] has recently shown that traditional algorithms are able to outperform almost all deep learning architectures.

While our work serves as a first milestone towards better characterizing beyond-mainstream music and listeners of such music, future work should focus on user modeling techniques to individually target the different subgroups, for example by integrating knowledge about openness and diversity. With respect to analyzing openness and diversity of users and user groups, we would also like to work on a more formal definition of these dimensions, which would not only allow us to measure them more precisely but also to integrate them into the recommendation calculation process.

Additionally, since previous research has shown that the listener's cultural background impacts the quality of music recommendations [512], we plan to compare the cultural and socioeconomic aspects of beyond-mainstream and mainstream music listeners. We plan to employ these aspects by means of Hofstede's cultural dimensions [193] and the World Happiness Report [187].

Finally, another avenue for future work is the research in the area of fair music recommender systems. Here, we plan to build user models that are capable of accounting for the complex characteristics of beyond-mainstream music listeners presented in this paper. While we believe that more specialized user models could help to provide better recommendations for users who currently receive worse recommendations (e.g., the U_{hard} subgroup identified in this paper), we also aim to highlight that such user models still need to be generalizable to avoid any unfair treatment of other users. Hence, future research should work on achieving a specialization-generalization trade-off in music recommender systems. We hope that our open *LFM-BeyMS* dataset as well as our source code will be of use to the scientific community for subsequent analyses.

Availability of Data and Materials

The *CultMRS* dataset can be found on Zenodo https://doi.org/10.5281/zenodo.3477842. Additionally, we provide our novel *LFM-BeyMS* dataset via Zenodo: https://doi.org/10.5281/zenodo.3784764. Our Python-based implementations are available via Github https://github.com/pmuellner/supporttheunderground.

Competing interests

The authors declare that they have no competing interests.

Funding

This work is funded by the TU Graz Open Access Publishing Fund and the Austrian Science Fund (FWF): V579.

Author's contributions

All authors contributed to manuscript revision, read, and approved the submitted version.

Acknowledgements

This work is supported by the Know-Center GmbH within the Austria FFG COMET program.

10. Personality Bias of Music Recommendation Algorithms

Publication

A. B. Melchiorre, E. Zangerle, and M. Schedl. Personality Bias of Music Recommendation Algorithms. In *Fourteenth ACM Conference on Recommender Systems*, RecSys '20, pages 533–538. ACM, 2020. DOI: 10.1145/3383313.3412223

Abstract

Recommender systems, like other tools that make use of machine learning, are known to create or increase certain biases. Earlier work has already unveiled different performance of recommender systems for different user groups, depending on gender, age, country, and consumption behavior. In this work, we study user bias in terms of another aspect, i.e., users' personality. We investigate to which extent state-of-the-art recommendation algorithms yield different accuracy scores depending on the users' personality traits. We focus on the music domain and create a dataset of Twitter users' music consumption behavior and personality traits, measuring the latter in terms of the OCEAN model. Investigating recall@K and NDCG@K of the recommendation algorithms SLIM, embarrassingly shallow autoencoders for sparse data (EASE), and variational autoencoders for collaborative filtering (Mult-VAE) on this dataset, we find several significant differences in performance between user groups scoring high vs. groups scoring low on several personality traits.

10.1. Introduction

Recommender systems in the multimedia domain—in particular in the music domain—have been shown to exhibit various kinds of biases, most notably on the item level (e.g., long-tail items are less frequently recommended [4, 78, 279]) and on the user level (e.g., users of a certain gender, belonging to a certain age group, or living in a certain country receive recommendations of different quality [415]). However, one important user characteristic that has not been studied yet under the perspective of recommender systems bias is personality. Personality traits are stable over a longer period of time and can, therefore, be considered in a way similar to gender when it comes to investigating bias [100]. Against this background, we address the following research questions: *Do state-of-the-art recommender algorithms yield different performance scores for different user groups in terms of personality traits? If so, how can these differences be characterized?*

In the study presented here, we focus on the music domain since some personality traits have already been shown to correlate with music preferences [377] and usage of music [84]. We, therefore, speculate that music listeners with different personality profiles might be treated differently by music recommender systems.

In this paper, we present related literature (Section 10.2), detail our methodology and data (Section 10.3), describe experimental setup and results (Section 10.4), present conclusions, limitations, and future research avenues (Section 10.5).

10.2. Related Work

Related literature can be categorized into recommender systems research that considers personality in the recommendation process and research on bias and fairness of recommender systems.

Personality traits are a psychological construct that remains stable over the years [100]. They are known to influence our preferences and consumption behaviors, e.g., towards music [149]. Research that integrates users' personality into the recommendation process has emerged only recently, though [463]. The most common personality model adopted in recommender systems research is the OCEAN model [305], which describes personality traits along five dimensions: *openness to experience* (conventional vs. creative thinking), *conscientiousness* (disorganized vs. organized behavior), *extraversion* (engagement with the external world), *agreeableness* (need for social harmony), and *neuroticism* (emotional instability).

While *personality-aware recommender systems* have been proposed in domains other than music (e.g., movies [330], food/recipes [5], and computer games [490]), we focus our discussion on music recommendation due to the scope of this paper. Lu and Tintarev propose a system that adapts according to users' personality traits and their diversity needs [300]. To this end, results of a collaborative filtering recommender are re-ranked with respect to the level of diversity each item, i.e., song, contributes to the recommendation list. Intra-list diversity is computed on item features such as music key, genre, and number of artists. Based on previously identified correlates between personality traits and diversity needs, the authors map each personality trait to a desired level of diversity and integrate this information as weighting term into the objective function used for re-ranking. Fernández-Tobías et al. present different personality-aware recommender systems to alleviate the cold-start problem in book, movie, and music recommendation [142]. In particular, they propose a matrix factorization approach for model-based collaborative filtering that integrates a user latent factor describing personality traits in terms of the five dimensions of the OCEAN model.

The concept of *fairness* requires systems not to discriminate against either a group [351] or individuals [128] in terms of recommendation quality. Establishing fairness typically involves identifying discriminated individuals or groups and, subsequently, developing algorithms that eliminate this discrimination [70]. Burke extended the concept of fairness to multisided fairness, noting that recommender systems have to consider the interests of all stakeholders of the system [68].

Recent research revealed a popularity bias in current recommendation algorithms. In particular, it was shown that users are recommended items that do not match their preference towards a certain popularity level (niche songs/artists are undervalued) [4, 279]. Ekstrand et al. investigated demographic biases in collaborative filtering scenarios with regards to age and gender and found that biases do not necessarily correlate with user group size [138]. Schedl et al. showed that users of different gender, age, and country receive (music) recommendations of different quality [415]. Our work, in contrast, is the first to investigate biases that may result from different personality traits.

10.3. Materials and Methods

In the following, we describe the creation of the used dataset (Section 10.3.1) and its composition (Section 10.3.2), the investigated recommendation algorithms (Section 10.3.3), and the evaluation metrics we adopt (Section 10.3.4). We publicly release the dataset and code needed to reproduce the experiments at `https://github.com/CPJKU/pers_bias`.

10.3.1. Data Acquisition

To obtain *behavioral data on music consumption* as well as information on users' personality, we exploit microblogs shared on Twitter, and particularly leverage so-called #nowplaying tweets in which users tweet about the music they are currently listening to. Along the lines of [183, 509], we utilize #nowplaying tweets stemming from 2018 and 2019 (256,705,566 tweets in total, gathered via the Twitter Streaming API,[1] searching for the keywords #nowplaying, #listento, or #listeningto). To extract track and artist information from those tweets, we use the MusicBrainz database[2] [450], an openly available database of music metadata. It provides metadata on artists, recordings, releases, etc., which is obtained through crowd-sourcing. For extracting artist names and track titles from tweets, we firstly strip URLs, mentions, and hashtags from the tweet text. Subsequently, we tokenize the text and identify the longest subsequence of tokens that corresponds to an artist entry in the MusicBrainz database. If we detect a matching artist, we remove the tokens constituting the artist name from the tweet and try to match the remaining text to a track of the detected artist, again using MusicBrainz metadata. If we cannot match a tweet against both, a track name and an artist name, we discard it.

We further refine the dataset by heuristically removing alleged radio stations through a careful check of the occurrence of certain words in the tweets, the number of shared links, and the number of listening events (user–item interactions). We identify a set of words hinting at radios (e.g., #listenlive and radio) and drop a "user" if at least half of their tweets contain any of these words. Since radio stations tend to share many tweets with links in it, we also drop a user if the majority of the user's tweets contain at least one link. Lastly, we remove all users above the 99.99% percentile of the number of listening events as radios commonly create an exorbitant number of listening events.

To obtain *personality* information of the users, we query the Twitter API[3] to get their most recent 1,000 tweets, excluding retweets.[4] Users with private or deleted profiles are discarded. These tweets are then fed to the IBM Personality Insight API,[5] which returns the personality estimates for each user according to the OCEAN model [305] (cf. Section 10.2), scaled to [0,1] in terms of percentile ranges. To achieve the maximum accuracy for trait prediction with the service,[6] we only keep users that tweet in English and use more than 3,000 words across their tweets. Lastly, we drop users with fewer than 5 listening events, as commonly done in related research [285, 406], and to enable the evaluation protocol (80:20 training/test split) detailed in Section 10.4.1.

10.3.2. Dataset Description

The processing steps described above eventually lead to a final dataset comprising 395,056 total listening events, 18,310 users with personality values, and 15,753 unique tracks.

[1] https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

[2] https://musicbrainz.org

[3] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline

[4] Personality is assumed to be stable through time, so it is reasonable to use recent behavioral data to predict personality traits.

[5] https://www.ibm.com/watson/services/personality-insights

[6] https://cloud.ibm.com/docs/personality-insights?topic=personality-thatut#sufficient

Basic statistics on the behavioral data in our final dataset, i.e., related to user–item interactions, can be found in Table 10.1. On the right side, a statistical summary of the number of tracks per user (user playcounts) and the number of users per track (track playcounts) is provided. The distribution of users' personality traits among the [0,1]-normalized scores is depicted in Figure 10.1, where the vertical lines denote the median values. To assess whether users with different personality profiles are treated differently by the mentioned approaches, we perform a median split over each personality trait, thus, effectively, creating two groups of users for each trait: the *high* group, scoring above the median, and the *low* group, scoring lower. Table 10.2 shows a set of statistics for each personality trait (in the columns) and user group (high vs. low on that trait; in the upper and lower part of the table, respectively). We observe that the total number of unique tracks is quite similar, regardless of personality trait and user group (within range [15,600, 15,700]), except for highly neurotic people who cover fewer items in their listening habits. In terms of the number of listening events, except for neuroticism, the high groups consistently show higher numbers, with a particularly pronounced difference between high and low groups for the traits extraversion and openness. This does not seem overly surprising since we expect that people who are extraverted and open to experience will listen to (and share) more music than introverts and less open users.

No. LEs	No. tracks	No. users
395,056	15,753	18,310

	Mean	Std.	Min.	25%	50%	75%	Max.
User playcounts	21.6	34.3	5.0	8.0	12.0	21.0	950.0
Track playcounts	25.1	33.1	8.0	11.0	16.0	26.0	986.0

Table 10.1.: Statistical summary of the behavioral data (users sharing listening events) in our dataset.

Group		Agr.	Con.	Ext.	Neu.	Ope.
High	No. unique tracks/user	19.1 ± 24.4	19.2 ± 25.5	20.0 ± 26.3	16.2 ± 18.4	19.5 ± 24.9
	No. unique tracks	15,694	15,674	15,655	15,429	15,652
	No. listening events	208,054	206,179	217,895	177,892	209,741
Low	No. unique tracks/user	17.3 ± 21.7	17.2 ± 20.4	16.4 ± 19.2	20.3 ± 26.9	16.9 ± 21.1
	No. unique tracks	15,664	15,695	15,672	15,607	15,619
	No. listening events	187,002	188,877	177,161	217,164	185,315

Table 10.2.: Mean and standard deviation of the number of unique tracks per user, for each personality trait and group; as well as total numbers of unique tracks and listening events created by all users in the low and in the high group.

10.3.3. Recommendation Approaches

We investigate to what extent the following three state-of-the-art recommendation approaches for implicit data yield different accuracy measures, depending on users' person-

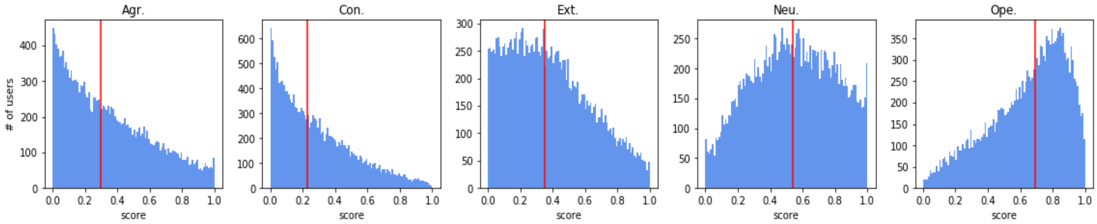


Figure 10.1.: Distribution of personality traits. The x-axis represents the (scaled) score for each trait while the y-axis represents the number of users. The red line represents the median for each trait.

ality traits. They have been shown, in extensive experiments, to perform well [110]; and we adapt them, where necessary, to cope with the non-binary nature of our interaction data. This selection of algorithms allows us to investigate both deep learning (non-linear) and traditional (linear) approaches:

- *Sparse Linear Methods (SLIM)* [336]: SLIM is a linear model that aims to compute top-n recommendations by factorizing the item–item co-occurrence matrix under non-negativity and L_1 and L_2 normalization constraints. The learned item coefficients are then used to sparsely aggregate past user interactions and predict the items the user will interact with in the future.
- *Embarrassingly Shallow Autoencoders (EASE)* [443]: EASE is a shallow linear model that could be considered as an extension of SLIM. Since EASE keeps only the L_2 norm constraint, a closed-form solution exists, making it computationally more efficient to train the model.
- *Variational Autoencoders (Mult-VAE)* [285]: Mult-VAE is a variational autoencoder architecture, i.e., a non-linear, probabilistic model, that uses multinomial conditional likelihood for collaborative filtering. Annealing is used to apply regularization for the learning objective.

10.3.4. Evaluation Metrics

We assess performance using *recall@K* and *normalized discounted cumulative gain@K* (NDCG@K) and report values averaged over all user groups in the test set.[7] Recall@K for user u is defined as

$$Recall@K(u) = \frac{1}{\min(K, N_u)} \sum_{i=1}^{K} rel(i) \tag{10.1}$$

where N_u is the number of items in the test set which are relevant to u, K is the length of the recommendation list, and $rel(i)$ is an indicator function signaling whether the recommended track at rank i is relevant to u (i.e., $rel(u) = 1$) or not relevant to u (i.e., $rel(u) = 0$).

[7] We will investigate beyond-accuracy metrics [230], such as coverage and diversity, as part of future work.

NDCG@K is defined as

$$NDCG@K(u) = \frac{DCG@K(u)}{IDCG@K(u)} \tag{10.2}$$

where $IDCG@K(u)$ is the ideal $DCG@K$ for user u, obtained when all items in u's test set are ranked in decreasing order of their play count, and $DCG@K(u)$ is the discounted cumulative gain at position k for user u, given by

$$DCG@K(u) = \sum_{i=1}^{K} \frac{rel(i)}{\log_2{(i+1)}} \tag{10.3}$$

where $rel(i)$ is the same indicator function as above. In our experiments, we compute recall@K and NDCG@K for $K = \{5, 10, 50\}$, to model different user needs, ranging from a user interested in only a few top recommendations to a user who inspects a large part of the recommendation list.

10.4. Experiments and Results

10.4.1. Experimental Setup

For our experiments, we apply a similar data splitting procedure as used in [285], i.e., we split the users in training/validation/test sets (80%/10%/10%) and for the held-out users we use 80% of their items for training and the remaining 20% as test set to compute the metrics.

We select the hyperparameters of the algorithms under investigation by performing a grid search over different parameters and optimizing for NDCG@50 across all validation users. For SLIM, we explore different α values (sum of the L_1 and L_2 coefficients) and L_1 *ratios* (ratio of L_1 coefficient in α). In detail, we search α in {.5, .1, .01, .001} and L_1 *ratio* in {1, .1, .01}. For EASE, we explore different weights for the L_2 norm in {1, 10, 10^2, $5 \cdot 10^2$, 10^3, 10^4, 10^5, 10^6, 10^7}. For Mult-VAE, we re-use most of the hyperparameters proposed in the original paper [285], except for the architecture and the annealing procedure. We set the total number of epochs to 100. We explore different (symmetric) architectures,[8] comprising 0 or 1 hidden layer(s) with fewer than 500 units for each layer.[9] As for the annealing procedure, we either anneal the regularization parameter through the end of training or stop it half-way by changing the annealing steps in {10,000, 20,000}. In addition, we explore caps for the annealing procedure in {0.5, 1}. After validation, the best model is selected and evaluated for each user group (defined by trait and high vs. low

[8] I-100-I, I-500-I, I-200-50-200-I, I-200-100-200-I, I-500-200-500-I, where I is the total number of tracks.

[9] Increasing the layers and/or the units did not improve results.

characteristic) in the test set. We conduct all experiments across 10 (random) splits of users among the sets, using 10 different seeds for splitting. Results are then averaged across the seeds.[10]

10.4.2. Results and Discussion

Tables 10.3 and 10.4 show the results for all algorithms, personality traits, and user groups, in terms of recall@K and NDCG@K, respectively. The values represent the performance scores averaged across the 10 runs/seeds. Note that the standard deviation of the results across these 10 runs is very low,[11] indicating that results are robust and stable across runs.

To assess the statistical significance of the differences between the high and low user group for each trait, we apply the two-tailed Mann-Whitney-U test on the high and low user scores [12] (NDCG@K and recall@K) and highlight their respective means in the tables in bold, with asterisks denoting the different alpha levels. The most notable observation is that for the traits neuroticism and openness most of all differences between the high and low groups are highly significant ($p < .001$), both in terms of recall@{5, 10, 50}, and NDCG@{5, 10, 50}. As a second observation, we find that the direction of difference in performance is nearly always consistent between all investigated algorithms, i.e., all algorithms treat the high vs. low groups unfairly in the same manner or direction; though the absolute value of the difference varies between algorithms, of course.

While performance for highly neurotic users is consistently better than the performance of low-neurotic-users, the opposite is true for all the other traits. These results seem to be correlated with the data consumption statistics shown previously (cf. Table 10.2), namely, a higher number of listening events and higher number of average tracks per user suggest a negative impact on the performance metrics. Furthermore, for conscientiousness and extraversion, the unfair treatment of user groups mostly appears for EASE and SLIM but not for Mult-VAE, while the opposite is true for agreeableness. This suggests that different models trained on the same data will lead to different kind of biases.

To finally answer our research questions: *Do state-of-the-art recommender algorithms yield different performance scores for different user groups in terms of personality traits?* They do indeed for some personality traits, in terms of recall@K and NDCG@K; most notably for the traits openness and neuroticism, and to a smaller extent for the other

[10] Note that the random split stated previously could in theory create unbalanced training/validation/test sets where some user groups may be underrepresented. We also carried out additional experiments where we enforced an equal split in each set for each group (one trait at the time). Results were consistent with the findings reported in this paper.

[11] Standard deviations are 0.0044, 0.0044, and 0.0042 for NDCG@5, 10, and 50, respectively; 0.0049, 0.0052, and 0.0064 for recall@5, 10, and 50, respectively.

[12] The results follow the same trend when using the Fisher's method to aggregate the p-values across the seeds, although with decreased significance level except for openness.

traits. *If so, how can these differences be characterized?* Scoring low on the personality trait results in higher performance for openness, extraversion, and conscientiousness, but in lower performance for neuroticism and agreeableness.

10.5. Conclusions, Limitations, and Future Work

In this work, we presented a first study to investigate the extent to which state-of-the-art recommendation algorithms (EASE, SLIM, and Mult-VAE) treat users with different personality traits in different ways, in terms of accuracy metrics (recall@K and NDCG@K). We found highly significant differences ($p < .001$) in both performance scores in particular for the traits neuroticism and openness as well as significant differences at $p < .01$ and $p < .05$ for the other traits.

While results are noteworthy, we also identify several *limitations* of the study at hand. First, like every research that leverages user data shared in online social networks, results obtained for Twitter users may not generalize to the population at large, or even to other platforms. Also, since Twitter's Streaming API only provides access to a small percentage of all shared tweets, the data is incomplete, though still substantial in size. Third, since we rely on self-disclosed information of Twitter users, the listening data we extract from their tweets may not accurately reflect the actual behavior of users, rather how the users want to be perceived (e.g., by avoiding to share guilty pleasure songs).

There are several directions we contemplate for *future research.* In the initial study presented here, we identified certain biases in terms of unequal treatment of different personality groups. However, the exact origin of these biases still needs to be investigated further. In particular, to which extent differences in accuracy can be explained by different consumption patterns of users with different personality (data bias), and to which extent these differences are introduced by the recommender system itself (algorithmic bias) remains an open question that will be addressed in the future. In addition, we plan to include beyond-accuracy metrics [230], e.g., diversity, serendipity, and coverage in our investigation. Finally, we would like to investigate the extent to which results generalize to platforms other than Twitter and additional recommendation algorithms.

Acknowledgements

This research is supported by Know-Center Graz, through project "Theory-inspired Recommender Systems".

		@5			@10			@50		
Trait	Algorithm	All	High	Low	All	High	Low	All	High	Low
Agr.	EASE	0.0366	0.0348	0.0385	0.0537	0.0534	0.0540	0.1122	0.1129	0.1113
	SLIM	0.0334	0.0320	0.0348	0.0486	0.0478	0.0494	0.1014	**0.1025***	**0.1002***
	Mult-VAE	0.0433	**0.0443***	**0.0423***	0.0634	**0.0655*****	**0.0611*****	0.1456	**0.1504*****	**0.1407*****
Con.	EASE	0.0366	**0.0328***	**0.0406***	0.0537	**0.0495***	**0.0580***	0.1122	0.1096	0.1148
	SLIM	0.0334	**0.0292*****	**0.0377*****	0.0486	**0.0447***	**0.0527***	0.1014	0.0989	0.1040
	Mult-VAE	0.0433	0.0405	0.0462	0.0634	0.0602	0.0665	0.1456	0.1424	0.1488
Ext.	EASE	0.0366	**0.0312****	**0.0420****	0.0537	**0.0467***	**0.0605***	0.1122	0.1032	0.1211
	SLIM	0.0334	**0.0284****	**0.0384****	0.0486	**0.0425***	**0.0547***	0.1014	0.0926	0.1101
	Mult-VAE	0.0433	**0.0378****	**0.0488****	0.0634	0.0568	0.0698	0.1456	0.1348	0.1560
Neu.	EASE	0.0366	**0.0422*****	**0.0311*****	0.0537	**0.0608****	**0.0466****	0.1122	0.1216	0.1028
	SLIM	0.0334	**0.0396*****	**0.0272*****	0.0486	**0.0562*****	**0.0411*****	0.1014	**0.1128***	**0.0900***
	Mult-VAE	0.0433	**0.0500*****	**0.0367*****	0.0634	**0.0721*****	**0.0547*****	0.1456	**0.1588****	**0.1324****
Ope.	EASE	0.0366	**0.0265*****	**0.0468*****	0.0537	**0.0410*****	**0.0663*****	0.1122	**0.0935*****	**0.1307*****
	SLIM	0.0334	**0.0232*****	**0.0436*****	0.0486	**0.0366*****	**0.0605*****	0.1014	**0.0841*****	**0.1186*****
	Mult-VAE	0.0433	**0.0316*****	**0.0550*****	0.0634	**0.0479*****	**0.0787*****	0.1456	**0.1232*****	**0.1678*****

Table 10.3.: Recall@5, 10, and 50 for each algorithm, personality trait, and group (high vs. low; and for all users). Significant differences between high and low groups are marked in bold and with an asterisk (Mann-Whitney-U test, * $p < .05$, ** $p < .01$, *** $p < .001$).

		@5			@10			@50		
Trait	Algorithm	All	High	Low	All	High	Low	All	High	Low
Agr.	EASE	0.0311	0.0295	0.0327	0.0392	0.0386	0.0399	0.0576	**0.0575***	**0.0577***
	SLIM	0.0279	0.0263	0.0295	0.0351	0.0340	0.0363	0.0517	**0.0514****	**0.0520****
	Mult-VAE	0.0380	**0.0385***	**0.0374***	0.0474	**0.0485*****	**0.0462*****	0.0724	**0.0747*****	**0.0701*****
Con.	EASE	0.0311	**0.0274***	**0.0349***	0.0392	**0.0352***	**0.0433***	0.0576	0.0542	0.0611
	SLIM	0.0279	**0.0241*****	**0.0319*****	0.0351	**0.0312***	**0.0391***	0.0517	0.0484	0.0551
	Mult-VAE	0.0380	0.0353	0.0407	0.0474	0.0445	0.0503	0.0724	0.0697	0.0752
Ext.	EASE	0.0311	**0.0266****	**0.0355****	0.0392	**0.0342***	**0.0441***	0.0576	0.0525	0.0626
	SLIM	0.0279	**0.0242****	**0.0317****	0.0351	**0.0310***	**0.0392***	0.0517	0.0474	0.0560
	Mult-VAE	0.0380	**0.0340****	**0.0417****	0.0474	0.0433	0.0513	0.0724	0.0678	0.0769
Neu.	EASE	0.0311	**0.0366*****	**0.0257*****	0.0392	**0.0454****	**0.0331****	0.0576	0.0639	0.0513
	SLIM	0.0279	**0.0335*****	**0.0224*****	0.0351	**0.0413*****	**0.0290*****	0.0517	0.0585	0.0449
	Mult-VAE	0.0380	**0.0436*****	**0.0324*****	0.0474	**0.0539*****	**0.0409*****	0.0724	**0.0798***	**0.0652***
Ope.	EASE	0.0311	**0.0221*****	**0.0400*****	0.0392	**0.0293*****	**0.0491*****	0.0576	**0.0463*****	**0.0688*****
	SLIM	0.0279	**0.0196*****	**0.0363*****	0.0351	**0.0261*****	**0.0441*****	0.0517	**0.0413*****	**0.0620*****
	Mult-VAE	0.0380	**0.0285*****	**0.0473*****	0.0474	**0.0366*****	**0.0581*****	0.0724	**0.0600*****	**0.0848*****

Table 10.4.: NDCG@5, 10, and 50 for each algorithm, personality trait, and group (high vs. low; and for all users). Significant differences between high and low groups are marked in bold and with an asterisk (Mann-Whitney-U test, * $p < .05$, ** $p < .01$, *** $p < .001$).

11. Evaluating Recommender Systems: Survey and Framework

Publication

E. Zangerle and C. Bauer. Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 2022. ISSN: 0360-0300. DOI: 10.1145/3556536

Abstract

The comprehensive evaluation of the performance of a recommender system is a complex endeavor: many facets need to be considered in configuring an adequate and effective evaluation setting. Such facets include, for instance, defining the specific goals of the evaluation, choosing an evaluation method, underlying data, and suitable evaluation metrics. In this paper, we consolidate and systematically organize this dispersed knowledge on recommender systems evaluation. We introduce the "Framework for EValuating Recommender systems" (FEVR) that we derive from the discourse on recommender systems evaluation. In FEVR, we categorize the evaluation space of recommender systems evaluation. We postulate that the comprehensive evaluation of a recommender system frequently requires considering multiple facets and perspectives in the evaluation. The FEVR framework provides a structured foundation to adopt adequate evaluation configurations that encompass this required multi-facettedness and provides the basis to advance in the field. We outline and discuss the challenges of a comprehensive evaluation of recommender systems, and provide an outlook on what we need to embrace and do to move forward as a research community.

11.1. Introduction

Recommender systems (RS) have become important tools in people's everyday life, as they are efficient means to find and discover relevant, useful, and interesting items such as music tracks [79], movies [58, 101], or persons for social matching [86]. A RS elicits the interests and preferences of individual users (e.g., by explicit user input or via implicit information inferred from the user's interactions with the system) and tailors content and recommendations to these interests and needs [488]. As for most systems, the evaluation of RS demands attention in each and every phase throughout the system life cycle—in design and development as well as for continuous improvement while in operation. Delivering quality is a necessary factor for a system to be successful in practice [32]. The evaluation may assess the core performance of a system in its very sense or may embrace the entire context in which the system is used [45, 190, 215, 390, 399]. Research on RS typically differentiates system-centric and user-centric evaluation, where the former refers to the evaluation of algorithmic aspects (e.g., the predictive accuracy of recommendation algorithms). The latter targets the users' perspective and evaluates how users perceive its quality or the user experience when interacting with the RS. In other words, the evaluation of a RS may cover system- or user-centric aspects concerning the system's context of use; a comprehensive evaluation essentially needs to address both as, for instance, provided recommendations that are adequate in terms of system-centric measures—for instance, the predictive accuracy of recommendation algorithms—do not necessarily meet a user's expectations [256, 313].

As we will demonstrate in this paper, there is an extensive number of dimensions that need to be considered when assessing the performance of a RS [174]. Besides the various facets of system configurations and the multitude of tasks that users aim to address with a RS (for instance, finding good items to getting a recommendation for a sequence of items) [190], there are multiple stakeholders involved who may have varying perspectives on a RS' goals [40]. There is a rich evaluation design space (e.g., evaluation setup, data collection, employed metrics) to draw from and we have to specify evaluation configurations that meet the respective evaluation objectives. Such objectives may relate to, for instance, improving rating prediction accuracy, increasing user satisfaction and experience, or increasing click-through rates and revenue. As a consequence, the comprehensive evaluation of a RS is a very complex task. As the ultimate goal is that a RS functions well as a whole in various contexts (e.g., for different user groups, for different kinds of tasks and purposes), the evaluation needs to assess the various dimensions that make up a RS' performance. What is more, frequently, we might need to shed light on a single dimension from various angles. For instance, Kamehkhosh and Jannach [229] could reproduce—and, thus, confirm—the results of their offline evaluation in an online evaluation on users' perceived quality of recommendations. Matt, Hess, and Weiß [304] evaluated several recommender algorithms for diversity effects from various angles; in taking these different perspectives, they found that the level of recommendation diversity perceived by users does not always reflect the factual diversity.

While the knowledge about system evaluation—and RS evaluation in particular—is continuously growing, empirical evidence, insights, and lessons learned are scattered across papers and research communities. To fill this research gap, this paper's main objective and major contribution is to consolidate and systematically organize this dispersed knowledge on RS evaluation. Therefore, we introduce the "Framework for EValuating Recommender systems" (*FEVR*) that we derive from the discourse on RS evaluation. We categorize the evaluation design space—i.e., the space that spans all required design decisions when conducting comprehensive RS evaluations. With FEVR, we provide a systematic overview of the essential facets of RS evaluation and their application. As FEVR encompasses a wide variety of facets to be considered in an evaluation configuration, it can accommodate comprehensive evaluations that address the various multi-faceted dimensions that make up a RS' performance. Besides guiding novices to RS research and evaluation, FEVR is a profound source for orientation for scientists and practitioners concerned with designing, implementing, or assessing RS. In addition, FEVR provides a structured basis for systematic RS evaluation that the RS research community can build on. We expect FEVR to serve as a guide to facilitate and foster the repeatability and reproducibility of RS research for researchers and practitioners, from novices to experts. Yet, comprehensive evaluation comes with challenges. Thus, to date, RS literature seems to concentrate on accuracy-driven offline evaluations and does not reflect the existing knowledge about what a comprehensive evaluation requires [77, 206, 207]. We outline and discuss the challenges of comprehensive RS evaluation, and provide an outlook on what we need to embrace and do to move forward as a research community.

11.2. Conceptual Basis

In the following, we briefly describe the foundations of recommender systems (Section 11.2.1) and their evaluation (Section 11.2.2).

11.2.1. Recommender Systems

Recommender systems aim to help users to deal with information and choice overload [17] by providing them with recommendations for items that might be interesting to the user [378, 384]. In the following, we give a brief overview of the foundational recommendation approaches: collaborative filtering, content-based RS, and more recent advances.

The most dominant approach for computing recommendations is collaborative filtering [402, 403], which is based on the collective behavior of a system's users. The underlying assumption is that users who had similar preferences in the past will also have similar preferences in the future. Hence, recommendations are typically computed based on the users' past interactions with the items in the system [64, 132, 190, 402, 403]. These interactions are recorded in a user-item rating matrix, where the users' ratings for items are stored. Such ratings may either refer to explicit ratings where users assign scores on a scale of, e.g., 1 to 5, to items, or implicit ratings. Fig. 11.1 shows an example of such a user-item matrix. Note that user-item matrices are highly sparse, as users only rate a small fraction of items available in the system. The algorithmic task of a RS is that of matrix completion—i.e., predicting the missing ratings in the matrix. This prediction of ratings can be performed using various methods: from traditional matrix completion methods, over neighborhood-based methods to matrix factorization, machine or deep learning-based approaches. For further information on these approaches, we refer the interested reader to the existing literature on these topics (e.g., [132, 263, 335, 402, 403, 447, 518]).

		Items				
		i_1	i_2	i_3	i_4	i_5
Users	u_1	0	3			3
	u_2	2				
	u_3			2	4	
	u_4		3		1	
	u_5	5		1		1

Figure 11.1.: Exemplary user-item matrix M with 0–5 star ratings for items i_1–i_5 by users u_1–u_5.

For user-based collaborative filtering RS that leverage the neighborhood of users in the two-dimensional space of the matrix, the most similar users to the current users are detected (the so-called neighborhood) by comparing their interactions with the system. Analogously, in item-based (item-item) collaborative filtering [402], the most similar items to the ones the user has previously rated highly are recommended, where the similarities are again computed based on the user-item matrix. Subsequently, items the user has not interacted with are sorted by their predicted ratings and the top-n items are then recommended to the user.

For collaborative filtering tasks, Matrix Factorization (MF) [262] aims to find latent factors in a joint, lower-dimensional space that explain user ratings for a given item. Specifically, latent representations for users and items are computed such that user-item interactions can be modeled as the inner product of user and item representations. This is often performed by applying optimization approaches to decompose the user-item matrix into two lower-dimensional matrices (e.g., stochastic gradient descent or alternating least squares), mostly relying on a regularized model to avoid overfitting (e.g., [156, 199]). Furthermore, learning-to-rank approaches model the computation of recommendations as a ranking task and apply machine learning to model the ranking of recommendations. In principle, we differentiate three types of learning-to-rank approaches: (i) point-wise (compute a score for each item for ranking; used in traditional CF approaches), (ii) list-wise (compute an optimal order of a given list), and (iii) pair-wise (consider pairs of items to approximate the optimal ordering). Bayesian Personalized Ranking is a popular example for learning-to-rank models; it is a generic, model-agnostic learning algorithm for predicting a personalized ranking [372] based on training pairs that incorporate positive and negative feedback.

In contrast, the central idea of content-based approaches is to recommend items that share characteristics with items that the user has previously liked (for instance, items that have a similar description or genre) [7, 328, 348]. Based on these characteristics of the user's previously liked items, a user model (often referred to as user profile) is built that represents the user's preferences. For the computation of recommendations, the user model is matched against item characteristics, and the most similar, and hence, relevant items are subsequently recommended to the user. Hybrid RS aim to combine collaborative and content-based filtering to leverage the advantages of both [67].

Similar to many other fields, a multitude of machine and deep learning models have been adapted for use in recommender systems. These include, for instance, deep neural networks for collaborative filtering, where the user-item interactions are modeled by a neural network [186], deep factorization machines [176], or (variational) autoencoders [285]. Convolutional Neural Networks (CNN) are mostly used for learning features from (multimedia) sources. For instance, learning representations from audio signals and incorporating them in a CF approach [470], or extracting and modeling latent features from user reviews and items [521]. Recurrent Neural Networks (RNN) allow modeling sequences and, hence, are applied for sequential recommendation tasks such as playlist generation or next-item recommendation [191, 368]. The use of reinforcement learning models for recommendation tasks is often performed by formulating the task as a multi-armed bandit problem (contextual bandits) [283, 308, 520]. Here, the bandit sequentially provides recommendations to users by also incorporating their contexts while continuously updating and optimizing the recommendation model based on user feedback. Furthermore, graph convolutional networks (GCN) model users, items, and potential side information in a graph. Based on this information, latent representations for nodes are learned by aggregating feature information from local neighbors (e.g., [184, 347]). This allows using these representations for candidate generation by nearest-neighbor lookups [498] or performing link-prediction tasks [48]. For a survey on deep learning for recommender systems, please refer to Zhang, Yao, Sun, and Tay [518]. In the context of evaluating deep learning recommender systems, it is noteworthy that evaluation metrics (cf. Section 11.3.4) are frequently used as loss functions (i.e., during the training phase).

Besides traditional recommendation approaches, there are several important extensions and specialized recommender systems that allow to deal with further input data or adapt to more specific use cases. These include, amongst others, context-aware recommender systems [9], where further contextual factors that describe e.g., the user's situation (for instance, time, location, weather) are leveraged to compute recommendations that are suitable for a given user in a given context. Sequential (or sequence-aware) recommender systems [367] analyze the sequence of user interactions to compute sequences of recommendations (e.g., recommending the next song to listen to, given a sequence of songs the user has just listened to). Conversational recommender systems provide more sophisticated interaction paradigms for preference elicitation, item presentation, or user feedback [212]. All of these approaches go beyond traditional recommender systems and user interactions and, hence, also require more complex evaluation methods and setups. We refer the interested reader to the respective survey articles [9, 212, 367] for details on such evaluations.

11.2.2. Evaluation of Recommender Systems

An evaluation is a set of research methods and associated methodologies with the distinctive purpose of assessing quality [446]. In their book, Jannach, Zanker, Felfernig, and Friedrich [217] state that evaluations are "methods for choosing the best technique based on the specifics of the application domain, identifying influential success factors behind different techniques, or comparing several techniques based on an optimality criterion" and that they are all required for effective evaluation research.

One of the early works on evaluating RS by Herlocker, Konstan, Terveen, and Riedl [190] focuses on the evaluation of collaborative filtering RS. The authors stress that evaluating RS is inherently difficult as (i) algorithms may perform differently on different datasets, (ii) evaluation goals may differ, and (iii) choosing the right metrics to compare individual approaches is a complex task. Gunawardana, Shani, and Yogev [174] provide a general overview of evaluation methods for RS.

Beel, Gipp, Langer, and Breitinger [43] and Beel, Langer, Genzmehr, Gipp, Breitinger, and Nürnberger [45] investigated evaluation approaches in the field of research paper recommender systems. They find that 21% of all approaches do not include an evaluation and that 69% are evaluated using an offline evaluation. Furthermore, they also looked into baseline usage and the datasets utilized for the evaluation. The authors note that the wide usage of no or weak baselines, as well as the usage of very different datasets, makes it difficult to compare the performance of the individual approaches, which in turn severely hinders advancing research in the field. Dehghani Champiri, Asemi, and Siti Salwah Binti [114] performed a systematic literature review on evaluation methods and metrics for context-aware scholarly recommender systems. In a meta-analysis, they reviewed 67 studies and find that offline evaluations are the most popular experiment type.

Comparing a RS' performance results to existing approaches and to competitive, strong baselines is also an important aspect for assessing and contextualizing the performance of the system. In this regard, Rendle, Zhang, and Koren [375] show that several widely-used baseline approaches, when carefully set up and tuned, outperform many recently published algorithms on the MovieLens 10M benchmark [181]. Along the same lines, Ferrari Dacrema, Boglio, Cremonesi, and Jannach [143] and Ferrari Dacrema, Cremonesi, and Jannach [144] investigated the performance of deep learning recommendation approaches published at major venues between 2015 and 2018, particularly, when compared to well-tuned, established, non-neural baseline methods. They found that the majority of approaches were compared to poorly-tuned, weak baselines and that only one of twelve neural methods was consistently outperforming well-tuned learning-based techniques.

Complimentary to existing works on RS evaluation, we consolidate and systematically organize this knowledge in the proposed Framework for EValuating Recommender systems (FEVR).

11.3. Recommender Systems Evaluation: A Review

Fig. 11.2 presents an overview of the components and general factors to be considered for recommender systems evaluation. Along this framework, we present the conceptual basis and paradigms used in recommender system evaluations. We term the framework *FEVR: Framework for EValuating Recommender systems* and emphasize that not necessarily all of these components and factors might be required to conduct a comprehensive evaluation of RS (this particularly holds for the proposed evaluation aspects). We consider this framework a collection and overview of potentially relevant components; it is meant to provide researchers and practitioners with an overview of the choices to be made when setting up the evaluation design and procedure.

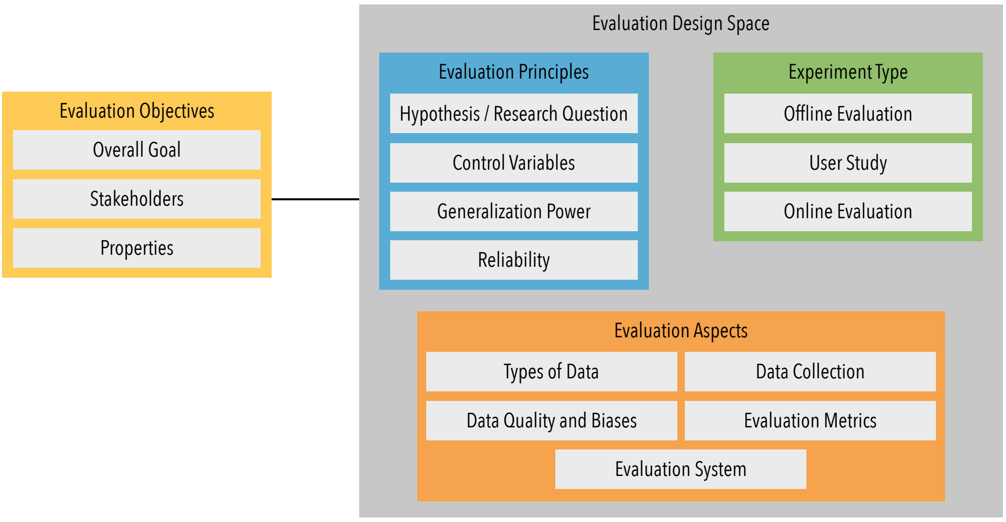


Figure 11.2.: Framework for EValuating Recommender systems (FEVR): evaluation objectives and the design space (along the orthogonal dimensions of evaluation principles, experiment type, and evaluation aspects).

The framework contains two main components: the *evaluation objectives* and the *evaluation design space*. When designing RS evaluations, deciding upon the objectives of the evaluation (*What should be evaluated? How can we measure this?*) has to be the first step because this directly influences the design decisions for the evaluation setup. The second component, the evaluation design space contains basic building blocks for the actual setup of the evaluation, which are assembled and configured based on the overall goal, the stakeholders involved, and the properties of the RS that need to be evaluated. In the evaluation design space, we distinguish three design blocks. The so-called *evaluation principles* describe the guiding principles of the evaluation—from the definition of the hypothesis underlying the evaluation to the generalizability of the conducted evaluation. These principles are tightly connected and influenced by the defined objectives because, for instance, the hypothesis to be evaluated needs to reflect the main objective of the evaluation (e.g., investigating whether algorithm A performs better than algorithm B). Given the objectives and principles, the *experiment type* can be considered a broad categorization of the type of experiment conducted to satisfy the objectives and principles (offline evaluation, user study, or online evaluation). *Evaluation aspects* can be considered a more fine-grained specification of the evaluation setup and, based on the defined requirements, control the detailed evaluation setup. They can be considered a set of configurations and decisions that do not necessarily all have to be considered for a RS evaluation; they should provide guidance for setting up and conducting comprehensive evaluations. We consider the choices of evaluation principles and experiment type rather high-level, whereas evaluation aspects cover more detailed and specific decisions regarding the evaluation setup.

In the following, we detail each of the framework's components and discuss their role in the activities of evaluating recommender systems.

11.3.1. Evaluation Objectives

At the heart of any evaluation, activity is the comparison of the objectives (target performance) to the observed results (actual performance) [352]. Thus—whether explicitly or implicitly stated—, evaluation is always based on one or more *evaluation objectives*. Evaluation objectives for evaluating a RS may take many forms. Essentially, objectives are shaped by the *overall goal* of academic and/or industry partners and the purpose of the system [206]. In this context, Herlocker, Konstan, Terveen, and Riedl [190] underline that any RS evaluation has to be goal-driven. Schröder, Thiele, and Lehner [425] emphasize that setting the goal of an evaluation has to be executed with sufficient care and it should be the first step of any evaluation to "define its goal as precisely as possible".

The underlying premise of any RS evaluation—in academia and industry—is that a RS is supposed to create value in practice [215] and have an impact in the real world [207]—in the long run, or even in the short run. Thus, overall goals that are typically investigated by academia, as well as industry, include, among others, a RS' contribution to increasing the user satisfaction [384], increase an e-commerce website's revenue [174], increase the number of items sold [384], sell more diverse items [384], help users understand the item space [206, 216], and engage users to increase their visit duration on a website, or return to the website [487]. Although several goals and purposes of RS are addressed in RS research and evaluation, it is remarkable that this variety of user tasks and RS purposes is not widely reflected in literature; instead, the main interpretation of the purpose of a RS seems to be "help users find relevant items," while other recommendation purposes are largely underexplored in the literature [206].

Concerning setting an evaluation goal, Schröder, Thiele, and Lehner [425] provide a vivid example of a precise evaluation goal: "Find the recommendation algorithm and parameterization that leads to the highest overall turnover on a specific e-commerce website, if four product recommendations are displayed as a vertical list below the currently displayed product." Crook, Frasca, Kohavi, and Longbotham [108] consider *prediction*, *ranking*, and *classification* as the most common tasks when viewed from the system's perspective. Considering the end consumers' perspective, Herlocker, Konstan, Terveen, and Riedl [190] discuss various end consumer tasks that a RS might be able to support (e.g., finding good items, finding all good items, recommending a sequence, discovering new items). Such tasks essentially describe the end consumers' overall goals that a RS might be evaluated for. A RS may, thus, be evaluated for their ability to find good items, find all good items, recommend a sequence, or discover new items. When describing pitfalls and lessons learned from their evaluation activities, Crook, Frasca, Kohavi, and Longbotham [108] emphasize the importance of choosing an overall evaluation goal that truly reflects business goals.

In general, evaluation objectives are shaped by the perspective that is taken in terms of the recommender's *stakeholders*. Beyond the end consumers, there are typically multiple stakeholders involved in and affected by recommender systems [1] with varying goals and potentially conflicting interests [40], which may manifest in different evaluation objectives. Currently, academic RS research tends to take the perspective of the end consumer [205], whereas research in industry is naturally built around the platform or system provider's perspective [513]. The item providers are a relatively new concern in RS research (e.g., [136, 145, 164]). To date, RS research that takes multiple stakeholders into account is scarce [1, 40, 122]. Table 11.1 provides an overview of evaluation papers that take different stakeholders' perspectives.

Stakeholder	**Examples**
Consumer	[155, 245, 247, 250, 364, 365]
Consumer Groups	[141]
Platform Provider	[43, 206, 208]
Item Provider	[379]
Multiple Stakeholders	[1, 40, 69]

Table 11.1.: Overview of papers on the evaluation of RS considering different stakeholders' perspectives.

While evaluating a recommender's overall goals (e.g., for an increase in a website's revenue) can be helpful, Gunawardana, Shani, and Yogev [174] point out that it can be most useful to evaluate how recommenders perform in terms of specific *properties*. This allows focusing on improving specific properties where they fall short (e.g., usage prediction accuracy, sales diversity, confidence in the recommendation, privacy level). The challenge is to identify the properties that are indeed relevant for a recommender's performance and show that it affects the users' experience [174], or the interests of other stakeholders. As different domains, applications, and consumer tasks have different needs, it is essential to decide on the most important properties to evaluate for the concrete RS at hand [174]. As already pointed out, Schröder, Thiele, and Lehner [425] emphasize the importance to define the evaluation goal as precisely as possible. Accordingly, specifying the relevant properties will provide the necessary fine granularity in defining the evaluation objective. As there might be trade-offs between sets of properties, it is often difficult to anticipate how these trade-offs affect the overall performance of the system [174]; this has to be considered in finding an appropriate evaluation design.

The evaluation objectives—including the overall goal, the stakeholder(s) being addressed, and the properties in the loop—are central to any evaluation effort and are, thus, the main drivers for configuring the evaluation design. We emphasize that poorly defined objectives will inevitably result in a poor evaluation.

11.3.2. Evaluation Design Space: Evaluation Principles

Closely related to the previously described evaluation objectives is a set of guiding principles for conducting evaluations [174]. These principles are pivotal in the process of designing and conducting RS evaluations because they lay the foundation of the evaluation procedure and provide the foundation of the setup. Hence, they should be considered and fixed early on in the process of evaluating a RS to shape the method and setup of the evaluation.

The first evaluation principle concerns hypotheses (or research questions) that capture the evaluation objectives. Depending on the overall goal and whether a problem can be clearly defined, the evaluation's overall goal may be translated to one or more a-priori formulated *hypotheses* that are grounded on prior knowledge (e.g., observations or theory) [478], or to one ore more exploratory-driven (broader) *research questions*.

Confirmatory evaluation involves testing one or more a-priori formulated *hypotheses*. Hence, a central starting point for confirmatory evaluation is the formulation of one or multiple *hypotheses* regarding the outcome of the evaluation. Defining a concise hypothesis is a highly important step as it allows to precisely define the evaluation's goal—the more precise the hypothesis, the clearer the evaluation setup as the hypothesis (in line with the evaluation objectives) shapes the evaluation design.[1] An example of a hypothesis for RS evaluation in the field of content-based video recommendations is "Our recommendation algorithm based on visual features leads to a higher recommendation accuracy in comparison with conventional genre-based recommender systems" [116]. Another example is Knijnenburg and Willemsen [248]'s hypothesis regarding preference elicitation (PE): "Novices have a higher satisfaction and perceive the system as more useful when they use the case-based PE method (compared to the attribute-based PE method), while experts have a higher satisfaction and perceive the system as more useful when they use the attribute-based PE method (compared to the case-based PE method)." Jannach and Bauer [207] claim that algorithmic RS research frequently comes without (appropriate) hypothesis development; they call for more theory-guided research with clear pointers to underlying theory (e.g., from psychology) that support the hypotheses.[2]

Yet, sometimes the evaluation objectives address a problem where little is known about the phenomenon. In such situations, the problem cannot be clearly defined at this state of research and the evaluation might, thus, be of exploratory nature (e.g., to get a better understanding of a problem or explore patterns). In such cases, it is not possible or suitable to formulate hypotheses. Instead, the evaluation's overall goal can be addressed by formulating research questions. For instance, Liang and Willemsen [286] seek to understand the effects of defaults in music genre exploration for which they formulate three research questions. Concerning author gender distribution in book recommendations, and Ekstrand and Kluver [136] explore how individual users' preference profiles propagate into the recommendations that they receive.

In hypothesis testing, all variables in the RS ecosystem that are not evaluated should be held fixed. Also in exploratory evaluation, the researcher exercises some control over the research conditions to explore the phenomenon of interest. The second evaluation principle, *control variables* (or short: controls) minimize the confounding variables and we eliminate potential external influences on the evaluation result [174, 476]. This allows a targeted evaluation and comparison of different algorithms and configurations by ensuring that only variables that are evaluated can be changed and that differences in the evaluation results are not due to some further, external factors. Going back to the previous example hypothesis regarding preference elicitation, the authors tested the hypothesis by utilizing the PE method, user expertise, and commitment as independent variables and measured satisfaction with the system, perceived usefulness, understand-

[1] For a discourse on the issues related to hypothesis-testing if a field is prone to produce "pseudo-empirical hypotheses" see Smedslund [436]. Smedslund [436] particularly emphasizes the problem that there is a prevailing belief (i.e., the current paradigm centered on the notion of probability) that "hypotheses that make sense are true, and hypotheses that do not make sense are false." For a discussion on the role of confirmation bias in making progress in research see Greenwald, Pratkanis, Leippe, and Baumgardner [170] or Wagenmakers, Wetzels, Borsboom, van der Maas, and Kievit [478].

[2] As an example, Jannach and Bauer [207] state that many works build on underlying assumptions such as "higher diversity is better" without providing any pointers to underlying research that would support such an assumption.

ability, and satisfaction with the chosen measures as dependent variables, while fixing all other variables. Jannach, Zanker, Felfernig, and Friedrich [217] refer to these controlled test conditions as the "internal validity" [73] of experiments.

The third important principle is the *generalization power* of evaluations, the extent to which the conclusions of the evaluation are generalizable beyond the current evaluation setup and experiments. The generalization power is tightly connected to the evaluation setup as e.g., varying the experimental setup, conducting experiments with different datasets, or extending the experiments to cover further application domains, user groups or stakeholders typically increases the generalization power of the evaluation [400]. Jannach, Zanker, Felfernig, and Friedrich [217] refer to this as "external validity" [73], namely the "extent to which results are generalizable to other user groups or situations [350]".

Reliability [217] is the fourth cornerstone of research evaluations as it demands evaluations to be consistent and free of errors (in both data and measurements). Particularly the consistency of multiple evaluation runs is crucial as this demonstrates the highly desirable repeatability of experiments, i.e., the ability to observe similar results of experiments conducted successively under the same (documented) settings and configurations, allowing consistent results describing the RS' performance. Tightly connected to repeatability is reproducibility, which refers to the ability "to duplicate the results of a prior study using the same materials as were used by the original investigator. That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results... Reproducibility is a minimum necessary condition for a finding to be believable and informative" [168]. Reproducible results require either access to the source code or a detailed description of the algorithm such that it can be re-implemented as well as having access to the dataset that was originally used. In this context, it is important to differentiate between reproducibility and replicability, which can be defined as the ability "to duplicate the results of a prior study if the same procedures are followed but new data are collected" [168]. In a nutshell, the three key concepts here can be defined as follows: reproducibility (different team, different experimental setup), repeatability (same team, same experimental setup), and replicability (different team, same experimental setup) as stated by the Association for Computing Machinery's badging initiative[3].

The ACM Conference on Recommender Systems (RecSys) has introduced a specific reproducibility track in 2020, which calls for "algorithmic papers that repeat and analyze prior work"[4]. Notably, this track calls for replicability as well as reproducibility papers. To further stress the importance of reproducibility, the best paper award of RecSys 2019 was awarded to Ferrari Dacrema, Cremonesi, and Jannach [144], in which the authors aim to reproduce the results of 18 papers from the field of deep learning recommender

[3] https://www.acm.org/publications/policies/artifact-review-badging, also following Hong [195].

[4] Call for Papers (Reproducibility Track) for RecSys 2022: https://recsys.acm.org/recsys22/call/#content-tab-1-1-tab

algorithms. In an extended version of that study, Ferrari Dacrema, Boglio, Cremonesi, and Jannach [143] find that only twelve out of the 26 evaluations had a reproducible setup, corresponding to a total of 46% of all systems. Here, the authors considered a paper to have a reproducible setup if (i) a working version of the source code is available or the code only has to be modified in minimal ways to work correctly, and (ii) at least one dataset used in the original paper is available (this also includes the train-test splits to be available or at least be reconstructible based on the description in the paper). The importance of documenting train/test splits (among other factors) is also highlighted by Cañamares, Castells, and Moffat [74], who show that different splitting methods and factors can lead to diverse evaluation results. On a similar note, Bellogıén and Said [47] make the case for accountability and transparency in RS research and argue that only if the conducted research and evaluation is reproducible, it is also accountable. They discuss the requirements for accountable RS research and derive a framework that allows for reproducible and, hence, accountable RS evaluation.

11.3.3. Evaluation Design Space: Experiment Type

In RS research, we distinguish three experiment types: offline evaluations, user studies, and online evaluations [42, 44, 154, 174, 190]. These different types describe the general experimental setup; Gunawardana and Shani [173] also refer to these types as “evaluation protocols”. The characteristics of these types include, among others, aspects of user involvement, utilized and obtainable data, or the type of insight that can be gained when using a specific experiment type. Please note that experiments of more than one type may be necessary to obtain a full picture of the performance of a RS. Offline evaluations are often the first step in conducting evaluations and there is a “logical evolution from offline evaluations, through user studies to online analyses” [154]. Fig. 11.3 shows an overview of the three experiment types, emphasizing that they represent a contrasting spectrum of experiments, covering diverse and different aspects of RS performance, where each type comprises a wide variety of evaluation setups and configurations.

Table 11.2 features an overview and comparison of the three established experiment types utilized in the RS research community. In the following, we further elaborate on their characteristics, goals, usage scenarios, and differences. Offline evaluations aim to compare different recommendation algorithms and settings; they do not require any user interaction and may be considered system-centric. In contrast, both, user studies and online evaluations, involve users and can be considered user-centric. Still, user involvement in evaluation does not necessarily target or capture the user experience, as discussed in Knijnenburg and Willemsen [247]. Also, for instance, Celma and Herrera [81] refer to leave-n-out methods, a typical offline evaluation method, as user-centric; while at the same time, they state that those evaluations measure accuracy and neglect (user-perceived) efficiency of recommendations.

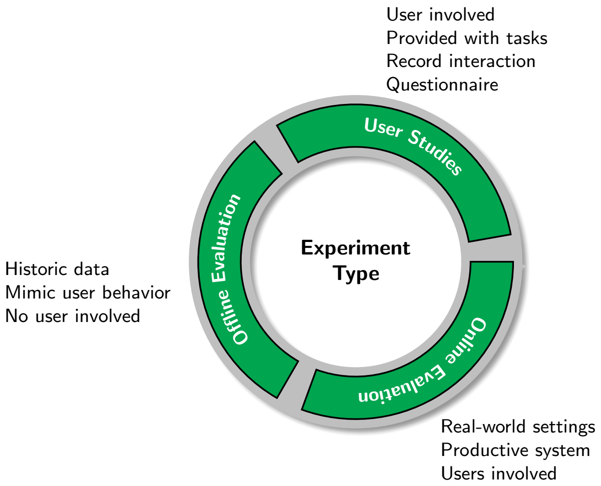


Figure 11.3.: Spectrum of experiment types.

Orthogonal to the distinction between online and offline experiments and user studies, Said, Tikk, Stumpf, Shi, Larson, and Cremonesi [399] and Knijnenburg and Willemsen [247] distinguish system- and user-centric evaluations and emphasize the different objectives of the adopted evaluation methods: system-centric evaluation methods evaluate the system, while user-centric evaluation methods target the user experience when interacting with the system.

Offline Evaluation

In research literature, the most frequently used experiment type for RS evaluation are so-called "offline evaluations".[5] An offline evaluation uses a pre-collected dataset that contains users' explicit feedback on items (e.g., ratings of items) or implicit feedback on items (e.g., the items purchased, viewed, or consumed) [174]. User behavior is then mimicked and simulated based on this historical data, no real users (and their interactions with the system) are involved in the actual experiments. For the experiments, parts of the rating information are removed (at random) from the given dataset's user-item matrix (so-called leave-n-out evaluation [106]) and, subsequently, the recommender algorithms are analyzed regarding their ability to[1] recommend (i.e., predict) the missing information [42, 44]—assessing whether the given recommender is apt to simulate user behavior to predict ratings that are reflected in the previously hidden data. Typically, offline evaluations are used to compare two or more RS algorithms (offline A/B testing [161]). Offline evaluations are meant to identify promising recommendation approaches by using metrics such as algorithmic accuracy and precision [42, 44, 247], and evaluating the predictive power of the approaches in regards to user preferences and opinions [154]—

[5] According to Jannach and Bauer [207], more than 92% of the 117 RS papers published at AAAI and IJCAI in 2018 and 2019 relied exclusively on offline experiments. At ACM RecSys 2018 and 2019, three of four papers only used offline evaluations. For the years 2006–2011, more than two-thirds of papers relied on offline experiments Jannach, Zanker, Ge, and Gröning [218].

Type	Description
Offline	Method: simulation of user behavior based on past interactions Task: defined by the researcher, purely algorithmic Repeatability: evaluation of an arbitrary number of experiments (e.g., algorithmic settings, models) possible at low cost Scale: large dataset, large number of users Insights: quantitative, narrow (focused on the predictive performance of algorithms)
User Study	Method: user observation in live or laboratory setting Task: defined by the researcher, carried out by the user Repeatability: expensive (recruitment of users) Scale: small cohort of users Insights: quantitative and/or qualitative (live user data, logging of user actions, eye tracking, questionnaires before/during/after task)
Online	Method: real-world user observation, online field experiment Task: self-selected by the user, carried out by the user Repeatability: expensive (requires full system and users) Scale: size of the cohort of users depending on evaluation system and user base Insights: quantitative and/or qualitative (live user data, logging of user actions, questionnaires before/during/after exposure to the system)

Table 11.2.: Overview of experiment types.

thus, the scope of evaluation objectives that can be evaluated with an offline evaluation is rather narrow [174] and focused on algorithmic tasks. It is, however, easy to repeat offline experiments as each evaluation run can be repeated any number of times using different recommender setups, algorithm parameters, datasets, users, etc., and also, at an arbitrary scale regarding the input dataset and the number of users evaluated.

Temporal aspects of data can be critical in the design of such an evaluation. Burke [66] suggests a "temporal leave-one-out approach", where the timestamps are considered in selecting which part of the data is used for training the model and which part for testing. Gunawardana, Shani, and Yogev [174] emphasize that selecting data based on timestamps allows for simulating what a recommender's predictions would have been if it had been running at the time when the data was available. Starting with no available prior data for computing predictions and stepping through user and interaction data in temporal order may be ideal in terms of simulating the system's behavior along the timeline; however, for large data sets, such an approach is computationally expensive [174].

While offline evaluations are widely used to obtain insights into the predictive performance of different recommendation algorithms, there are also disadvantages to offline evaluations. Given the described setup that relies on historic data, offline evaluation does not involve (current) real users. There is no interaction of users with the given (to be evaluated) RS algorithm in an actual system and the performance of the algorithm in a real-world scenario can not be assessed. Hence, the generalizability (external validity) of the findings obtained by offline experiments is limited, and frequently questioned [388]. For instance, a recent study [225] showed that offline experiments on historical data for a destination recommender system did show higher predictive accuracy than a subsequent user study. In another study [388], offline experiments underestimated the precision results of online evaluations.

Counterfactual learning methods [14, 449] overcome one of the key problems in offline evaluation; namely, that the dataset was logged from a real-world platform where a particular RS was active (i.e., logged policy) while the offline evaluation has the objective to evaluate another RS algorithm (i.e., target policy). With counterfactual learning methods, one can address the question of how well a new RS algorithm would have performed if it had been used instead of the policy that logged the historical data. This counterfactual approach also reduces the effect of selection biases (i.e., biases introduced into the data through the actions selected by the logging policy) [222].

User Study

A user study is conducted by recruiting a (small) set of human test subjects who perform several pre-defined tasks that require them to interact with the RS [174]. The goal here is to observe user interaction with the system and to distill real-time feedback on the system's performance and the user's perceived value of the system. This observation can either be conducted in a laboratory or live setting. Thereby, the user study may be conducted in a way to compare two or more systems in, for instance, an experimental setup (controlled experiment[6]); a user study may also focus on exploring a particular phenomenon without comparing specific RS approaches (exploratory study) [365]. The subject's interaction behavior with the system is recorded and based on these records, various quantitative measures may be computed (e.g., time to complete a task, click-through rate, recommendation acceptance). In addition, the setting of a user study allows for asking subjects closed or open-ended questions during, before, and after the task potentially also providing qualitative feedback [174]. Further, user studies allow for integrating various forms of measurements such as eye-tracking or think-aloud-testing [338]. Hence, user studies allow for the most comprehensive feedback compared to the other experiment types, enabling answers to the widest set of questions. Notably, user studies measure user experience at the time of recommendation.

It is important to note that user studies may lead to costs [44, 114]—both in user time and financial costs, often limiting the number of users being involved in the study or the number of different system dimensions and configurations that can be investigated and evaluated [154]. This also involves recruiting a set of participants that are willing to participate in the experiment. These participants should be representative of the actual

[6] Although an experimental setup may compare two or more variants of a RS, the term A/B testing is typically not used in the context of user studies.

users of the system and have access to a running recommender system. Furthermore, users who know that they are part of a study often tend to behave differently (called "Hawthorne effect" [272]). Generally, user studies need extensive preparation and planning as repeating is expensive. Besides, a wide range of sensors and detailed observations of user behavior need to be installed to make sure to not miss any vital information during the study as a potential rerun of the experiment may be expensive. These factors can be regarded as causes of the low adaption of user studies in the field of RS research [42, 44].

Online Evaluation

In online evaluations, the RS is deployed in a real-world, live setting [174]. In contrast to user studies, users are not presented with specific tasks, but use the system to perform self-selected real-world tasks (also referred to as "live user studies" [154]). Hence, online evaluations allow for the most realistic evaluation scenario as users are self-motivated and use the system in the most natural and realistic manner [252, 253]. Accordingly, online evaluations provide feedback on the system's performance for users with a real information need [174]. Similar to user studies, user behavior is logged and recorded and subsequently used to distill performance metrics such as recommendation accuracy. Typically, this also involves measuring the acceptance of recommendations using click-through rates (CTR).

While the real-world setting is an advantage of online evaluations [252, 253], this very setting limits this experiment type to collecting user behavior on the platform (e.g., purchases, clicks, dwell, time). When inferring user satisfaction from user behavior [42, 44], care has to be taken because user behavior (e.g., consumption activity) may also have different or additional causes such as integrated nudges [221], closing an app interpreted as negative feedback for an item [60], or biases due to interruptions or distractions [328].

We note that online evaluations require access to a RS and its implementation. Typically, online evaluations are carried out in the form of A/B testing [252] to compare the adapted system/algorithm to the original system. In so-called online field experiments [94], a small number of users are randomly assigned and exposed to different alternative RS configurations (instantiations) without their knowledge, and the users' interactions with the systems are recorded and analyzed. These instantiations may include different recommendation algorithms, and algorithm configurations, but also different interaction, presentation, or preference elicitation strategies.

Furthermore, online evaluations are performed for recommender systems that require a high amount of interaction with the user or where specifically the interaction with the user needs to be evaluated (e.g., critiquing systems [383], conversational recommender systems [98, 212], or novel interfaces and interaction strategies [59, 246]) that can not be simulated are often evaluated in online field experiments. Traditionally, this includes A/B testing [252].

11.3.4. Evaluation Design Space: Evaluation Aspects

In this section, we provide an overview of individual aspects that are to be considered in the evaluation design space. Many of these aspects are interwoven, and their characteristics might have interdependencies or may be mutually exclusive. For instance, synthetic datasets come—by definition—without any user involvement. Experiments with random assignment of user groups to treatments (e.g., different RS algorithms) may be implemented in user studies (in randomized control trials or laboratory experiments) or online evaluation (in online field experiments) alike. Furthermore, trade-offs between RS performance indicators have been observed; for instance, a trade-off between accuracy and diversity is frequently reported [230, 269], and diversity may not necessarily be perceived by users [197, 249] or differently across users [269]. Moreover, situational factors may influence user experience due to varying user needs or preferences [120, 385].

Consequently, frameworks are an effective means to organize this complexity. For instance, Knijnenburg, Willemsen, Gantner, Soncu, and Newell [249]'s *framework for the user-centric evaluation of recommender systems* models this complexity for studies addressing the user experience.

In the following, we describe the individual aspects of the evaluation design space.

Types of Data

The essential basis for the evaluation of recommender systems is data. The characteristics of data can be manifold and may depend on the type of data used for computing the actual recommendations, among other factors. In the following, we give a brief overview of the different characteristics of data that may be used when evaluating RS.

Implicit and Explicit Rating Data User ratings are usually collected by user behavior observations, which may, for instance, include records on the items that a user consumed, purchased, rated, viewed, or explored (e.g., pre-listening of songs), where the source may be an existing dataset or one that is collected for the respective study. When relying on the observation of user behavior when interacting with a RS, we typically distinguish between explicit and implicit feedback [199, 220]. Explicit feedback is provided directly by the user and the data unequivocally captures the user's perception of an item. Platforms that employ recommender systems frequently integrate mechanisms that allow users to explicitly express their interests in or preference for a specific item via rating scales (e.g., 5-star rating scale, likes, thumbs-up, or thumbs-down). The rating scales used for providing explicit feedback usually allow for expressing both, positive and negative preferences (e.g., a scale from "I like it a lot" to "I do not like it").

Implicit feedback, in contrast, is inferred from a user's observable and measurable behavior when interacting with a RS (e.g., purchases, clicks, dwell time). When relying on implicit feedback, evaluations presume that, for instance, a consumed item is a quality choice, while all other items are considered irrelevant [28]. Hence, implicit feedback is typically positive only (e.g., purchase, click), while the absence of such information does not imply that the user does not like an item (e.g., a user not having listened to a track does not imply that the user does not like the track). Some scenarios also allow for opportunities for negative implicit feedback such as, for instance, the skipping of songs. Furthermore, implicit feedback can be used to infer relative preferences (for example, if a user watched one movie ten times whereas other movies typically only once or play counts of songs for a music RS). Thus, implicit feedback may be mapped to a degree of preferences, thereby ranging on a continuous scale to its positive extremity [220]. When interpreting implicit feedback, the assumption is that specific behavior is an indication of quality, regardless of whether the behavior may have other causes; thus, for example, closing a music streaming app may be mistakenly interpreted as a skip (i.e., negative feedback) [60] or the behavior is influenced by interruptions or distractions [113].

Dimension	**Explicit Feedback**	**Implicit Feedback**
Accuracy	High	Low
Abundance	Low	High
Expressivity of user preference	Positive and negative	Positive
Measurement reference	Absolute	Relative

Table 11.3.: Characteristics of explicit and implicit feedback (adapted from Jawaheer, Szomszor, and Kostkova [220]).

Most of the research in RS has focused on either explicit or implicit data [220], while comparably few have combined these two heterogeneous types of feedback (e.g., [284, 293, 295]). Table 11.3 summarizes the characteristics of explicit and implicit feedback. Explicit feedback provides higher accuracy than implicit feedback inferred from behavior based on assumptions (e.g., the assumption that users only click on items they are interested in). Typically, when users navigate through a platform that employs a RS, an abundance of data about user behavior is logged. In contrast, users are reluctant to explicitly rate items [189, 239], which leads to comparably little explicit feedback data. Note that explicit feedback tends to concentrate on either side of the rating scale because users are more likely to express their preferences if they feel strongly in favor or against an item [20].

Although explicit and implicit feedback are heterogeneous types of feedback [220], research investigating the relations between implicit and explicit feedback for preference elicitation has shown that using implicit feedback is a viable alternative [345]. Still, implicit measures may reveal aspects that explicit measures do not [461]—particularly when user self-reports are not consistent with the actual user behavior. Integrating both, observation of actual user behavior and users' self-reports on intentions and perceptions, may deliver rich insights for which each approach in isolation would be insufficient.

Note that many evaluation designs presume that a consumed item is a viable option also in other contexts (e.g., another time, location, or activity) and consider item consumption as a generally valid positive implicit feedback. What the user indeed experiences, however, remains unclear. The validity of the feedback for other contexts depends on the design of the feedback mechanism. For instance, an item rated with five stars may be the user's lifetime favorite, but still not suitable for a certain occasion (e.g., a ballad for a workout, or a horror movie when watching with kids).

User, Item Information RS algorithms typically heavily rely on rating data for the computation of recommendations, where the computations are mostly based solely on the user-item matrix. However, these approaches have been shown to suffer from sparsity and also, the cold-start problem, where recommendations for new items or users cannot be computed accurately as there is not enough information on the user or item, respectively. Therefore, metadata on the user, items, or context can also be incorporated to further enhance recommendations (this information is often referred to as "side information") [140, 337]. For instance, keywords describing the item may be extracted from e.g., reviews on the item [18] or social ties between users can be extracted from relationships in social networks [303, 456]. Furthermore, when working toward business-oriented goals and metrics (cf. Section 11.3.4), data such as revenue information or click-through rates also have to be logged and analyzed [208]. In addition, context information is useful when users are expected to have different preferences in different contexts (e.g., watching a movie in a cinema or at home [397]).

Qualitative and Quantitative Data Besides collecting behavioral user data (e.g., implicit feedback logged during user interactions with the system), evaluations may also rely on qualitative or quantitative evidence where data is gathered directly from the user. Quantitative data collection methods are highly-structured instruments—such as scales, tests, surveys, or questionnaires—, which are typically standardized (e.g., same questions, same scales). This standardization facilitates validity and comparability across studies. Quantitative evidence allows for a deductive mode of analysis using statistical methods; answers may be compared and interrelated and allow for generalization to the population. Qualitative evidence is frequently deployed to understand the sample studied. Commonly used data collection methods include interviews, focus groups, and participant observations, where data is collected in the form of notes, videos, audio recordings, images, or text documents [151].

Natural and Synthetic Data Herlocker, Konstan, Terveen, and Riedl [190] distinguish between natural and synthetic datasets. While natural datasets capture user interactions with a RS or are directly derived from those, synthetic datasets are artificially created (e.g., [115, 502]). Natural datasets contain (historical) data that may capture previous interactions of users with a RS (e.g., user behavior such as clicks or likes), or data that may be associated with those (e.g., data that reflects users attitudes and feelings while interacting with a RS), or are derived from user interactions (e.g., turnover attributed to recommendations). In cases where a natural real-world dataset that would be sufficiently suitable for developing, training, and evaluating a RS is not available, a synthesized dataset may be used. In such cases, a synthesized dataset would allow for particularly modeling specific critical aspects that should be evaluated. For instance, a synthesized dataset may be created to reflect out-of-the-norm behavior. Herlocker, Konstan, Terveen, and Riedl [190] stress that a synthetic dataset should only be used in the early stages of developing a RS and that synthesized datasets cannot simulate and represent real user behavior. Yet, not only user-behavior-related data can be synthesized. For instance, Jannach and Adomavicius [205] use fictitious profit values to investigate profitability aspects of RS.

Data Collection

Data collection methods may be distinguished based on their focus on considering contemporary and historical events, where methods may rely on past events (e.g., existing datasets, data retrieved from social media) or investigate contemporary events (e.g., observations, laboratory experiments) [497]. In the following, we give an overview of data collection aspects.

User Involvement Evaluation methods may be distinguished with respect to user involvement. While offline studies do not require user interaction with a RS, user-centric evaluations need users to be involved, which is typically more expensive in terms of time and money [174, 399]—which is especially true for online evaluations with large user samples (cf. Section 11.3.3.

Randomized control trials are often considered the gold standard in behavioral science and related fields. In terms of RS evaluation, this means that users are recruited for the trial and randomly allocated to the RS to be evaluated (i.e., intervention) or to a standard RS (i.e., baseline) as the control. This procedure is also referred to as A/B-testing (e.g., [108, 252, 253]). Randomized group assignment minimizes selection bias, keeping the participant groups that encounter an intervention or the baseline as similar as possible. Presuming that the environment can control for all the remaining variables (i.e., keeping the variables constant), the different groups allow for comparing the proposed system to the baselines. For instance, randomized control trials that are grounded on prior knowledge (e.g., observations or theory) [478] and where the factors measured (and the instruments used for measuring these factors) are carefully selected may help determine whether an intervention was effective [83]; explaining presumed causal links in real-world interventions is often too complex for experimental methods.

While randomized control trials are conducted in laboratory settings, experiments in field settings are typically referred to as "social experiments". Thereby, the term social experiment covers research in a field setting where investigators treat whole groups of people in different ways [497]. In online environments, this is referred to as online field experiment [94]. In field settings, the investigator's control is only partly possible. Field settings have the advantage that outcomes are observed in a natural, real-world environment rather than in an artificial laboratory environment—in the field, people are expected to behave naturally. Overall, though, field experiments are always less controlled than laboratory experiments, and field experiments are more difficult to replicate [289]. For RS evaluation, an online field experiment [94] very often requires collaboration with a RS provider from industry, who is commercially oriented and may not be willing to engage in risky interventions that may cause losing users and/or revenues. However, e.g., for the 2017 RecSys Challenge[7], the best job recommendation approaches (determined by offline experiments) were also rolled out in XING's productive systems for online field experiments. Besides collaborating with industry, a number of online field experiments have been carried out using research systems (e.g., MovieLens) (e.g., [94, 519]). However, when carrying out a study with a research system, one also has to build a user community for it. Generally, this is often too great an investment just to carry out an experiment. This is why many researchers have argued for funding shared research infrastructure (in both Europe and the USA) including a system with actual users [255].

It is important to note that it is rarely feasible to repeat studies with user involvement for a substantially different set of algorithms and settings. System-centric (offline) evaluations are, in contrast, easily repeatable with varying algorithms [174, 190, 399]. However, offline evaluations have several weaknesses. For instance, data sparsity limits the coverage of items that can be evaluated. Also, the evaluation does not capture any explanations why a particular system or recommendation is preferred by a user (e.g., recommendation quality, aesthetics of the interface) [190]. Knijnenburg et al. [247, 249] propose a theoretical framework for user-centric evaluations that describes how users' personal interpretation of a system's critical features influences their experience and interaction with a system. In addition, Herlocker, Konstan, Terveen, and Riedl [190] describe various dimensions that may be used to further differentiate user study evaluations. Examples for user-centric evaluations can, for instance, be found in the following sources: [103, 124, 135, 398].

Overall, while system-centric methods without user involvement typically aim to evaluate the RS from an algorithmic perspective (e.g., in terms of accuracy of predictions), user involvement opens up possibilities for evaluating user experience [399].

[7] http://2017.recsyschallenge.com/

User Feedback Elicitation At the core of many recommender systems are user preference models. Building such models requires eliciting feedback from users, for which—at runtime—data is typically collected while users interact with the RS. For evaluation purposes, we can leverage a wider variety of methods for data collection. For instance, besides considering interaction logs, observation [237] may be used to elicit users' behavior. An alternative method is to ask users for their behavior or intentions in a particular scenario. Such self-reports may be directed to reports on what they have done in the past or what users intend to do in a certain context. However, self-reports may not be consistent with user behavior [85, 264, 461] because the link between an individual's attitude and behavior is generally not very strong [16]. Furthermore, the process of reporting on one's behavior may itself induce reflection and actual change of behavior, which is known as the question-behavior effect [439]. It is, thus, good practice to combine self-report data with other information or to apply adjustment methods because such an assessment considering several perspectives is more likely to provide an accurate picture [19].

For the elicitation of feedback on user experience, Pu, Chen, and Hu [364] propose an evaluation framework, called ResQue (Recommender systems' Quality of user experience) that aims to evaluate a comprehensive set of features of a RS: the system's usability, usefulness, interaction qualities, influence of these qualities on users' behavioral intentions, aspects influencing the adoption, etc.. ResQue provides specific questionnaire items and is, thus, considered highly operational. Knijnenburg, Willemsen, Gantner, Soncu, and Newell [249]'s framework for the user-centric evaluation of recommender systems takes a more abstract approach. It describes the structural relationships between the higher-level concepts without tying the concepts to specific questionnaire items. Therefore, it provides the flexibility to use and adapt the framework for various RS purposes and contextual settings and allows researchers to define and operationalize a set of specific, lower-level constructs. Both frameworks (i.e., Knijnenburg, Willemsen, Gantner, Soncu, and Newell [249] and Pu, Chen, and Hu [364]) may be integrated in user studies and online evaluations alike.

Existing Datasets One advantage of relying on existing datasets is that (offline) evaluations can be conducted early in a project. In comparison to soliciting and evaluating contemporary events, it is frequently "easier" and less expensive in terms of money and time to rely on historical data [174]. Also, by utilizing popular datasets (e.g., the MovieLens dataset [181]), results can be compared with similar research. However, such an evaluation is restricted to the past. For instance, the goal of a leave-n-out analysis [64] is to analyze to which extent recommender algorithms can reconstruct past user interactions. Hence, such an evaluation can only serve as a baseline evaluation measure because it only considers items that a user has already used in the past; assuming that unused items would not be used even if they were actually recommended [174]. Additional items that users might still consider useful are not considered in the evaluation because ratings for these items are not contained in the dataset [506]. This is also stressed by Gunawardana, Shani, and Yogev [174] by the following scenario: “For example, a user may not have used an item because she was unaware of its existence, but after the recommendation exposed that item the user can decide to select it. In this case, the number of false positives is overestimated.”

Another risk is that the dataset chosen might not be (sufficiently) representative—the more realistic and representative the dataset is for real user behavior, the more reliable the results of the offline experiments are [174]. In fact, the applicability of the findings gained in an evaluation based on a historic dataset is highly impacted by the “quality, volume and closeness of the evaluation dataset to the data which would be collected by the intended recommender system” [154].

Table 11.4 lists datasets widely used for evaluating recommender systems and their main characteristics such as the domain, size, rating type, and examples of papers that have utilized the dataset in the evaluation of their system. There are different MovieLens datasets, differing in the number of ratings contained (from 100K ratings in the ML100K dataset to 20M ratings in the ML20M dataset; we list ML1M and ML20M in the table). Alternatively, the yearly conducted RecSys-Challenge[8] also provides datasets from a yearly changing application domain and task (including job, music, or accommodation (hotel) recommendation).

Dataset	**Domain**	**Size**
MovieLens20M[9] [181]	Movie ratings	20,000,263 ratings; range [0.5,5]
MovieLens1M[10] [181]	Movie ratings	1,000,209 ratings; range [1,5]
BookCrossing[11] [524]	Book ratings	1,157,112 ratings; range [1,10]
Yelp[12]	Business ratings	8,021,122 ratings; range [0,5]
MovieTweetings[13] [125]	Movie ratings	871,272 ratings; range [0,10]

Table 11.4.: Widely used datasets for evaluating RS.

Data Quality and Biases

An important factor for RS evaluation is the quality of the data underlying the evaluations. This also includes potential biases that may be contained in the data used for the evaluation. Such biases may occur in the distributions of users, items, or ratings that are selected to be part of the evaluation dataset. As Gunawardana, Shani, and Yogev [174] note, a typical example of a bias that is introduced when assembling the evalua-

[8] http://www.recsyschallenge.com/

[9] available for download at https://grouplens.org/datasets/movielens/

[10] available for download at https://grouplens.org/datasets/movielens/

[11] available for download at http://www2.informatik.uni-freiburg.de/~cziegler/BX/

[12] available for download at https://www.yelp.com/dataset

[13] available for download at https://github.com/sidooms/MovieTweetings

tion dataset is excluding users or items with low rate counts from the dataset. Careful curation of datasets by e.g., using random sampling methods for limiting the size of the dataset to reduce the experimentation time is crucial to avoid such biases. Another aspect that may influence data biases is the collection method [174], where users do not provide feedback that is evenly distributed among items as, for instance, users tend to rate items that they particularly like or dislike. However, methods such as resampling or reweighting may be used for correcting such biases [444, 445].

Adomavicius and Zhang [12] investigated the characteristics of rating data and their impact on the overall recommendation performance. The characteristics they used for describing rating datasets are (i) overall rating density (i.e., the degree to which the user-item matrix is filled), (ii) rating frequency distribution (i.e., how ratings are distributed among items; rating data often exhibits a long-tail distribution [22, 344]), and (iii) the variance of rating values. In a set of experiments, the authors find that the recommendation performance is highly impacted by the structural characteristics of the dataset, where rating density and variance exhibit the highest impact.

Evaluation Metrics

There is an extensive number of facets of RS that may be considered when assessing the performance of a recommendation algorithm [173, 174]. Consequently, also the evaluation of RS relies on a diverse set of metrics, which we briefly summarize in the following. The presented metrics can be utilized for different experiment types, however, we note that due to the dominance of offline experiments, most of the presented metrics stem from offline settings.

In their early work on RS evaluation, Herlocker, Konstan, Terveen, and Riedl [190] differentiate metrics for quantifying predictive accuracy, classification accuracy, rank accuracy, and prediction-rating correlation. Along the same lines, Gunawardana and Shani [173] investigate accuracy evaluation metrics and distinguish metrics based on the underlying task (rating prediction, recommending good items, optimizing utility, recommending fixed recommendation lists). Said, Tikk, Stumpf, Shi, Larson, and Cremonesi [399] classify the available metrics into classification metrics, predictive metrics, coverage metrics, confidence metrics, and learning rate metrics. In contrast, Avazpour, Pitakrat, Grunske, and Grundy [30] provide a more detailed classification, distinguishing 15 classes of evaluation dimensions; these range, for instance, from correctness to coverage, utility, robustness, and novelty. Gunawardana, Shani, and Yogev [174] distinguish prediction accuracy (rating prediction accuracy, usage prediction, ranking measures), coverage, novelty, serendipity, diversity, and confidence[14]. Chen and Liu [90] review evaluation metrics from

[14] Gunawardana, Shani, and Yogev [174] list further aspects that need to be evaluated, such as trust and risk, which are typically assessed via questionnaires. We do not cover these aspects here and kindly refer the interested reader to the original manuscript.

four different perspectives (or rather, disciplines): machine learning (e.g., mean absolute error), information retrieval (e.g., recall or precision), human-computer interaction (e.g., diversity, trust, or novelty), and software engineering (e.g., robustness or scalability).

In the following, we discuss the most widely used categories of evaluation metrics. Table 11.5 gives an overview of these metrics, which we classify along the lines of previous classifications. For an extensive overview of evaluation metrics in the context of recommender systems, we refer to [90, 160, 173, 174, 190, 346, 425]. Several works [395, 454] have shown that the metrics implemented in different libraries for RS evaluation (Section 11.3.4) sometimes use the same name while measuring different things, which leads to different results given the same input. Similarly, Bellogın and Said [47] report that papers present different variations of metrics (e.g., normalized vs non-normalized; computed over the entire dataset or on user-basis and then averaged); and sometimes the details of the evaluation protocol are not reported in papers [47, 74]. Tamm, Damdinov, and Vasilev [454] conclude that the more complex a metric is, the more room there is for different interpretations of the metric, leading to different variations of metric implementations. As a result, this might lead to misinterpretations of results within an evaluation [454], and limits the comparability across evaluations [47, 74, 395, 454]. In line with previous works [47, 74], we urge for a more detailed description of evaluation protocols as this will strengthen reproducibility and improve accountability [47].

Fundamentally, we emphasize that it is important to evaluate a RS with a suite of metrics because a one-metric evaluation will—in most cases—be one-sided and cannot characterize the broad performance of a RS. When optimizing a RS for one metric, it is crucial to also evaluate whether this optimization sacrifices performance elsewhere in the process [160, 190]. For instance, it is doubtful whether a RS algorithm optimized for prediction accuracy while sacrificing performance in terms of diversity, novelty, or coverage is overall desirable. Similarly, a RS that performs equally across various user groups but for all groups with similarly low accuracy and low diversity will not likely reach a good user experience for any user. It is, thus, crucial to measure—and report—a set of complementary metrics. In many cases, it will be key to find a good balance across metrics.

Prediction accuracy refers to the extent to which the RS can predict user ratings [174, 190]. These include error metrics that quantify the error of the rating prediction performed by the RS (i.e., the difference between the predicted rating and the actual rating in a leave-n-out setting). The most widely used prediction accuracy metrics are mean absolute error and root mean squared error.

Usage prediction metrics can be seen as classification metrics that capture the rate of correct recommendations—in a setting where each recommendation can be classified as relevant or non-relevant [173, 174, 190]. This involves binarizing ratings such as, e.g., on a rating scale of 1–5 considering ratings of 1–3 as non-relevant and ratings of 4 and 5 as relevant. The most popular usage prediction metrics are recall, precision, and the

Category	**Metrics**	**References**
Prediction accuracy	Mean absolute error (MAE)	[190, 430]
	(Root) Mean squared error ((R)MSE)	[190, 430]
Usage prediction	Recall, precision, F-score	[99, 473]
	Receiver operating characteristic curve (ROC)	[451]
	Area under ROC curve (AUC)	[35]
Ranking	Normalized discounted cumulative gain (NDCG)	[219]
	Mean reciprocal rank (MRR)	[477]
Novelty	Item novelty	[76]
	Global long-tail novelty	[81, 230]
Diversity	Intra-list diversity	[524]
Coverage	Item coverage	[160, 190]
	User space coverage	[160, 174]
	Gini index	[174]
Serendipity	Unexpectedness	[190]
	Serendipity	[230, 326]
Fairness across users	Value unfairness	[496]
	Absolute unfairness	[496]
	Over/underestimation of fairness	[496]
Fairness across items	Pairwise fairness	[49]
	Disparate treatment ratio (DTR)	[435]
	Equal expected exposure	[119]
	Equity of amortized attention	[50]
	Disparate impact ratio (DIR)	[435]
	Viable-Λ test	[401]
Business-oriented	Click-through rate (CTR)	[111, 159, 165]
	Adoption and conversion rate	[111, 165]
	Sales and revenue	[91, 274]
Articles providing an overview of metrics: [90, 160, 173, 174, 190, 346, 425].		

Table 11.5.: Overview of evaluation metrics.

F-score, which combines recall and precision. Precision is the fraction of recommended items that are also relevant. In contrast, recall measures the fraction of relevant items that are indeed recommended. Often, this includes restricting relevant items to the k most relevant items, where the system's ability to identify the k most suitable items for a user is captured as opposed to evaluating all recommendations (often referred to as recall@k or precision@k, respectively) [190]. Alternatively, the receiver operating characteristic curve can also be used to measure usage prediction, where the true positive rate is plotted against the false positive rate for various recommendation list lengths k. These curves can also be aggregated into a single score by computing the area under the ROC curve (AUC).

Ranking metrics are used to quantify the quality of the ranking of recommendation candidates [173, 346]. Relevant recommendations that are ranked higher are scored higher, whereas relevant documents that are ranked lower are provided a discounted score. Typical ranking metrics include normalized discounted cumulative gain (NDCG) [219], or mean reciprocal rank (MRR) [477].

Diversity refers to the dissimilarity of the items recommended [76, 230, 269, 474], where low similarity values mean high diversity. Diversity is often measured by computing the intra-list diversity [437, 524] and thereby, aggregating the pairwise similarity of all items on the recommendation list. Here, similarity can be computed, e.g., by Jaccard or cosine similarity [230].

Novelty metrics aim at measuring to which extent recommended items are novel [76]. Item novelty [201, 523] refers to the fraction of recommended items that are indeed new to the user, whereas global long-tail novelty measures the global novelty of items—i.e. if an item is known by few users and hence, is in the long tail of the item popularity distribution [64, 81].

Serendipity describes how surprising recommendations are to a user and hence, is tightly related to novelty [230, 326]. However, as Gunawardana, Shani, and Yogev [174] note, recommending a movie staring an actor that the user has liked in the past might be novel, but not necessarily surprising to the user. The so-called unexpectedness measure compares the recommendations produced by a serendipitous recommender to the recommendations computed by a baseline [326]. Building on the unexpectedness measure, serendipity can be measured by the fraction of relevant and unexpected recommendations in the list [230] or the unexpectedness measure [6].

Coverage metrics describe the extent to which items are actually recommended [7, 160]. This includes catalog coverage (i.e., the fraction of all available items that can be recommended; often referred to as item space coverage) [399], user space coverage [174] (i.e., the fraction of items that are recommended to a user; often also referred to as prediction coverage [160]), or measuring the distribution of items chosen by users (e.g., by using the Gini index or Shannon entropy) [174]. Coverage metrics are also used to measure fairness because coverage captures the share of items or users that are served by the RS.

Fairness metrics concern both, fairness across users and across items. In both cases, fairness may be captured at the level of the individual or at group level. Individual fairness captures fairness (or unfairness) at the level of individual subjects [50] and implies that similar subjects (hence, similar users or similar items) are treated similarly [128]. Group fairness defines fairness on a group level and requires that salient subject groups (e.g., demographic groups) should be treated comparably [129]; in other words, group fairness is defined as the collective treatment received by all members of a group [50].

A major goal of group fairness is that protected attributes—for instance, demographic traits such as age, gender, or ethnicity—do not influence recommendation outcomes due to data bias or model inaccuracies and biases [50, 427].

Fairness across users is typically addressed at the group level. One way to address group fairness from the user perspective is to disaggregate the user-oriented metrics to measure and compare to which extent user groups are provided with lower-quality recommendations (e.g, [130, 131, 138, 202, 267, 315, 427]). Yao and Huang [496] propose three (un-)fairness metrics: value unfairness measures, whether groups of users receive constantly lower or higher predicted ratings compared to their true preference; absolute unfairness measures the absolute difference of the estimation error for groups, and under/overestimation of fairness measures inconsistency in the extent to which predictions under- or overestimate the true ratings.

Fairness across items addresses the fair representation of item groups [50] and it is addressed at group level and at the level of individual items, too. The goal of many metrics is to measure the exposure or attention [50, 435] an item group receives and assess the fairness of this distribution: in a ranked list of recommendations, lower ranks are assumed to get less exposure and, thus, less attention.[15] Beutel, Chen, Doshi, Qian, Wei, Wu, Heldt, Zhao, Hong, Chi, and Goodrow [49] propose the concept of pairwise fairness, which aims to measure whether items of one group are consistently ranked lower than those of another group. Other metrics put exposure across groups and relevance of items into relation. The disparate treatment ratio (DTR) [435] is a statistical parity metric that measures exposure across groups proportional to relevance. Diaz, Mitra, Ekstrand, Biega, and Carterette [119] consider the distribution over rankings instead of a single fixed ranking. The idea behind the principle of equal expected exposure is that "no item should receive more or less expected exposure than any other item of the same relevance grade" [119]. Biega, Gummadi, and Weikum [50] capture unfairness at the level of individual items; they propose the equity of amortized attention, which indicates whether the attention is distributed proportionally to relevance when amortized over a sequence of rankings. The disparate impact ratio (DIR) [435] goes further than exposure and considers the impact of exposure: DIR measures across items groups, whether items obtain proportional impact in terms of the click-through rate. The viable-Λ test [401] accounts for varying user attention patterns through parametrization in the measurement of group fairness across items.

Business-oriented metrics are used by service providers to assess the business value of recommendations [208]. While service providers naturally are interested in user-centered metrics as positive user experience impacts revenue, business-oriented metrics allow to directly measure click-through-rates [111, 159, 165, 236], adoption and conversion

[15] While many approaches assume logarithmic discounting of attention [435], also other approaches exist, too (for example, using a geometric distribution [50] or parametrizing varying attention patterns [401]).

rates [111, 165], and revenue [91, 274]. Click-through rates measure the number of clicks generated by recommendations, whereas adoption and conversion rates measure how many clicks actually lead to the consumption of recommended items. Therefore, adoption and conversion rates, and even more so, the sales and revenue generated by recommended items, more directly measure the generated business value of recommendations.

Evaluation System

Involving users in evaluations requires a (usually graphical) user interface to allow users to interact with the system. In RS evaluation, different options are available concerning the extent to which the evaluated system is incorporated in a real-world or industry environment. This aspect is highly interwoven with the choice of whether to involve users in the evaluation. For an offline algorithmic evaluation, there is no need to provide a user interface, as no users are involved. However, measuring user experience requires the involvement of users and, hence, a user interface. Konstan and Riedl [256] distinguish three designs of systems for evaluation: (i) systems dedicated to experimental use, which may range from interfaces for purely experimental research to more sophisticated systems; (ii) collaborating with operators of real-world (industry) systems for online field (real-world) experiments; and (iii) developing and maintaining a research system and (large) user community for (long-term) evaluations.

Also, "bad" user interface design may bias the assessment of RSs because they affect the users' overall experience [101, 365]. Users may evaluate recommendations differently if they were presented by a improved user interface. Putting effort into a good (or neutral) user interface design is expensive. Maintenance costs for a dedicated research system are high, too. Likewise, acquiring a large set of users may be challenging. All these issues contribute to the low adoption of non-offline evaluations.

Generally, there are several RS evaluation frameworks. Most of these libraries are primarily for offline evaluations and hence, provide a set of recommender algorithms and an evaluation framework. These frameworks include, for instance, LensKit [133, 137], MyMediaLite [157], LibRec [175], Rival [395, 396], Surprise [200], or Elliot [25]. Recently, Beel, Collins, Kopp, Dietz, and Knoth [41] proposed a "living-lab" for online evaluations of scholarly recommender systems that can be used on top of a production recommender system and logs all user actions (clicks, purchases, etc.) to evaluate the algorithms' effectiveness in online evaluations and user studies.

11.4. Mapping a Fictitious Case to FEVR

In the following, we first present a fictitious case as an example evaluation. Then, we showcase how this scenario can be mapped to the FEVR framework (Section 11.4.1) and discuss the limitations of this evaluation configuration (Section 11.4.2).

The context of the example is as follows: in an academic setting, a group of researchers has developed a novel recommendation algorithm, termed *RecAlg*, that aims to improve the item diversity of music recommendations by incorporating audio and lyrics features of tracks, while also improving (or, at least, maintaining) prediction accuracy. The goal is to support users in finding likable music by providing personalized music recommendations.

11.4.1. Mapping to FEVR

With this example case, we revisit the major components of the proposed FEVR framework and discuss the design components regarding the evaluation of the RS. We provide a compact overview of the design components of the example case in Table 11.6. Note that the evaluation principles component draws from the other components and is discussed at the end of this section.

As for the *evaluation objectives*, the overall goal is to evaluate whether users are indeed able to find likable music when provided with recommendations computed by the novel RecAlg algorithm. The stakeholders addressed are naturally the users of the system (algorithm), but—as the proposed algorithm aims to improve the diversity of recommended tracks—artists could also benefit from this increased item diversity as a more diverse set of artists may now be represented in the set of recommended tracks. As for the properties evaluated, the researchers aim to evaluate the diversity of recommendations; particularly, the change in catalog coverage (and hence, the change to items in the long tail of the popularity curve). FEVR's evaluation design space (cf. Fig. 11.2 for a graphical overview of the core components) encompasses the main evaluation principles, which we discuss in the following. As *experiment type*, the group of researchers chooses to perform an offline evaluation to assess the basic algorithmic performance of RecAlg (that still needs to be confirmed in later user-centric evaluations to evaluate whether users do indeed also perceive the provided recommendations as more diverse and accurate).

The *evaluation aspects* that need to be considered in this evaluation encompass the data to be used, but also the evaluation system and the evaluation metrics applied. As this example is situated in a scientific setting, offline experiments can be performed using an existing evaluation framework. In this particular case, the Elliot [25] framework is chosen. As for the data used for the evaluation, the researchers rely on an extensive dataset of listening events, namely the LFM-2b dataset [410]. This dataset contains 2 billion listening events (i.e., a user has listened to a particular song) which represent implicit feedback as well as detailed side information on the music tracks contained. LFM-2b is

FEVR Component	**Brief Description**
Evaluation Objectives	
Overall Goal	To evaluate whether users are able to find likable music in the recommendations computed by the novel RecAlg algorithm
Stakeholders	Users of the system (algorithm) Artists may also benefit from an increased item diversity as a more diverse set of artists may be represented
Properties	Item diversity in the recommendations; catalog coverage
Evaluation Principles	
Hypothesis / Research Question	H_1: RecAlg provides users (on average) with more diverse recommendations with respect to the intra-list diversity while maintaining prediction accuracy compared to the baseline algorithm.
Control Variables	Follow accountability framework by Bellogín and Said [47] (for randomization in dataset splitting to prevent selection bias)
Generalization Power	Limited due to lack of user involvement and dataset biases
Reliability	Follow accountability framework by Bellogín and Said [47]
Experiment Type	Offline Evaluation with A/B-testing
Evaluation Aspects	
Types of Data	Implicit ratings (listening events), side information for music tracks
Data Collection	LFM-2b dataset [410]
Data Quality and Biases	Platform bias, popularity bias, skewed gender distribution, imbalanced country distribution.
Evaluation Metrics	Prediction accuracy with RMSE; intra-list diversity in terms of different unique artists
Evaluation System	Existing evaluation framework ELLIOT [25]

Table 11.6.: FEVR: Overview of Example Evaluation.

the most extensive and recent public dataset in the domain. In the experimental setup, users for the training, test, and validation sets are chosen randomly to avoid introducing biases in this step. The metrics employed directly reflect the goals of our evaluation: for quantifying RecAlg's prediction accuracy, the group of researchers rely on RMSE and for measuring the diversity of recommendation lists, they rely on intra-list diversity. As for the *evaluation principles*, the main hypothesis is that the novel RecAlg approach provides users with more diverse recommendations concerning the intra-list diversity while maintaining prediction accuracy. The generalization power of this evaluation is limited in the sense that it does not involve users, the dataset used encompasses biases, and the fact that implicit feedback data was used. For a further discussion on the limitations of this evaluation, please refer to Section 11.4.2. To ensure the reliability of the conducted experiments and to control for confounding factors, the group of researchers follows the accountability framework by Bellogıén and Said [47].

With this example evaluation scenario, we have illustrated how an evaluation configuration can be mapped to FEVR and demonstrated that FEVR can be used as a checklist for an evaluation configuration. However, we note that this described evaluation scenario is a very basic one and has limitations, which we discuss in the following (Section 11.4.2).

11.4.2. Limitations and Discussion

As for any kind of offline evaluation with a publicly available dataset, the generalization power is limited due to the inherent biases in the dataset. First and foremost, there is a platform bias and evaluation results would not generalize to other music streaming platforms or even to other application domains. This and other dataset biases (e.g., skewed gender distribution of users, imbalanced distribution across user countries) may be addressed by extending the evaluation by integrating further datasets. Comparing evaluation results across datasets provides the opportunity to reason across all results and, thereby, increases the generalizability of findings.

Furthermore, the presented example evaluation builds on assumptions which are—at least in the scenario presented—not grounded on prior knowledge and not justified with respective pointers to underlying theory or observations. Many assumptions concern a user's need for diversity and their perceived diversity. The evaluation setting is built on a set of assumptions including that users indeed enjoy or even want artist diversity in their playlists, that all users have similar diversity needs, that an individual's diversity need is constant (i.e., context-independent), and that individuals perceive the provided recommendations as diverse as the intra-list metric suggests. With a lack of literature on those topics, it is necessary to integrate additional methods in the evaluation to explore and clarify these assumptions (and to obtain a more comprehensive picture of the experiment's results). Frequently, this will require a mixed methods research approach [107] where quantitative and qualitative research methods are combined. A recent tutorial at the ACM SIGKDD Conference on Knowledge Discovery & Data Mining 2021 [442][16] demonstrates how a mixed-methods approach is used in real-world (industry) settings (specifically, the tutorial presents case studies from Spotify) to analyze and justify assumptions, develop business-oriented metrics, so that further evaluation steps are valid and reliable.

Finally, the results of the presented evaluation will give direction about the next evaluation steps. Hence, the evaluation results will inform whether the novel RecAlg algorithm

[16] The slides of the tutorial can be found at `https://github.com/kdd2021-mixedmethods`. A similar case study has already been presented at the tutorial on "Mixed methods for evaluating user satisfaction" at ACM RecSys 2018 [158].

achieves sufficient performance to be further evaluated in, for instance, a user study (e.g., by particularly considering diversity perception). Unsatisfactory results will suggest revisiting the algorithm and exploring further opportunities. Again, FEVR can serve as a checklist for the configuration of the next evaluation step.

11.5. Discussion, Conclusion, and Future Directions

The review of literature on RS evaluation shows that finding an adequate configuration for the comprehensive evaluation of a RS is a complex endeavor; the evaluation design space is rich, and finding an adequate configuration may be challenging. In this paper, we consolidate and systematically organize the dispersed knowledge on RS evaluation. With FEVR, we provide a basis, overview, and guidance for researchers as a profound source for orientation in evaluating RS.

Still, for RS to work in practice (i.e., in industry) as well as for the research community to advance, we have to engage in a more comprehensive evaluation of RS—an evaluation that embraces the entire RS and its context of use and does not only address single dimensions in isolation. Yet, to date, such a comprehensive evaluation approach is hardly adopted in RS research.

From a practical perspective, the reasons for the low adoption of comprehensive evaluation—and the excessive use of offline evaluation only—are manifold [77]: (a) identifying an adequate combination of evaluation designs and configurations (more broadly speaking, aspects that can and need to be addressed together) meeting the evaluation objectives may be a complex task (particularly for inexperienced researchers); (b) the costs for involving users in the evaluation process are high (compared to pure offline studies); (c) integrating results of multiple evaluation designs and configurations into an entire study is complex and drawing conclusions from components effectively across the entire study can be challenging; and (d) evaluations considering multiple methods require adequate skills in various (at least two) evaluation methods. Senior researchers tend to have a preference for one method [353] and apply methodologically what they are strong at, which also prevents young researchers from learning (and possibly adopting) additional methodical approaches. While these reasons for non-adoption are all plausible, we argue that the goal should be to use the most adequate evaluation setting for set evaluation objectives. In many cases, this will require an integration of multiple evaluation designs. This comes with several challenges:

- *Methodological issues.* Jannach, Mobasher, and Berkovsky [213] point to methodological issues and research practices in RS evaluation where novel recommender approaches are compared to weak (e.g., non-optimized) baselines [143, 144, 301]. Showing “phantom progress”, as Ludewig, Mauro, Latifi, and Jannach [301] term it, hamper the progress of research and is of little value for evaluating the recommender approach under investigation. Along this line comes the need for good evaluation protocols that are documented in papers with sufficient detail [47, 74] to strengthen reproducibility. Yet, using many different metric variants—even if properly documented—hinders the comparability across works. Accordingly, the development and establishment of standardized protocols is a core issue that the community needs to address for advancing the field.

- *Methodological competencies.* Employing a comprehensive RS evaluation requires researchers to build competencies in a set of methods as expertise in only one method is insufficient. Furthermore, consolidating these methods' results into an integrated picture of the system's quality and the perceived quality of the RS is another skill set that has to be developed.

- *Datasets.* A crucial task is to find or elicit datasets that are sufficiently representative of the use case that the RS is evaluated for. Reducing biases inherent in real-world data is considered one of the key challenges [56]. Furthermore, Jannach and colleagues [213, 215] call for the evaluation of RS' longitudinal effects. One of the challenges involved is to obtain a rich dataset over a long period of time in a comparable manner. With the fast-paced progress in RS research, RS approaches are continually being updated and fine-tuned and datasets embracing a longer period do likely encompass dynamics of having different RS approaches active at different times.

- *Multi-stakeholder RS.* Research on multi-stakeholder RS is currently still in its infancy. For evaluation, we can observe "a diversity of methodological approaches and little agreement on basic questions of evaluation" [1].

- *Conversational RS.* Although conversational RS seem to advance at an accelerated pace, no consensus on how to evaluate such systems has evolved yet [212]. For instance, conversational RS rely on natural language processing (NLP), and evaluating language models and generation models is itself an inherently complex task [212]. Using and evaluating such models in task-oriented systems such as conversational RS might be even more challenging [212].

- *Domain-specifics.* The quality of recommendations depends on the particular domain or application. For a news recommender, the recency of items is important. In the music domain, recommenders are often considered useful when they support discovery of the back catalog. In tourism, the geographical vicinity might be relevant. The evaluation configuration has to take such domain-specifics into account [207]. This requires deep domain knowledge (and data), which frequently requires collaborating with domain experts in academia and industry. Evaluation without domain expertise bears the risk of being based on wrong assumptions.

- *Multi-* evaluations.* Comprehensive evaluations encompassing the required multi-facettedness (e.g., multi-method, multi-metrics, multi-stakeholder) appears to be an adequate and necessary pathway for RS evaluation. The key issue is that we need to establish evaluations that are apt to characterize the broad performance of a RS, which can only be accomplished with thoughtful integration of multiple methods. This requires an evaluation culture where a suite of metrics is evaluated and reported, and where the needs of the multiple stakeholders of RS are considered. The hurdles of such evaluations, including involved costs, required skills, etc. are—undeniably—impediments we need to take up and overcome these challenges to advance recommender systems and the field of recommender systems at large. Yet, this seems to require a paradigm shift in our research community's evaluation efforts [207].

While FEVR framework provides a structured basis to adopt adequate evaluation configurations, we—as a community—have to move forward together: it is on us to adopt, apply, and establish suitable practices.

Acknowledgements

This research was funded in whole, or in part, by the Austrian Science Fund (FWF): P33526.

Bibliography

[1] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, and L. Pizzato. Multistakeholder Recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction*, 30(1):127–158, 2020. DOI: 10.1007/s11257-019-09256-1.

[2] H. Abdollahpouri and R. Burke. *Multistakeholder Recommender Systems*. In *Recommender Systems Handbook*. Springer, 2022, pages 647–677. DOI: 10.1007/978-1-0716-2197-4-17.

[3] H. Abdollahpouri and M. Mansoury. Multi-sided exposure bias in recommendation, 2020. DOI: 10.48550/ARXIV.2006.15772.

[4] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher. The Unfairness of Popularity Bias in Recommendation. In *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems*, volume 2440 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[5] I. Adaji, C. Sharmaine, S. Debrowney, K. Oyibo, and J. Vassileva. Personality Based Recipe Recommendation Using Recipe Network Graphs. In *Social Computing and Social Media. Technologies and Analytics*, pages 161–170. Springer, 2018.

[6] P. Adamopoulos and A. Tuzhilin. On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected. *ACM Transactions on Intelligent Systems and Technology*, 5(4):1–32, 2015. DOI: 10.1145/2559952.

[7] 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, 2005. DOI: 10.1109/tkde.2005.99.

[8] G. Adomavicius and K. YoungOk. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911, 2012. DOI: 10.1109/tkde.2011.15.

[9] G. Adomavicius, K. Bauman, A. Tuzhilin, and M. Unger. *Context-Aware Recommender Systems: From Foundations to Recent Developments*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 211–250. DOI: 10.1007/978-1-0716-2197-4-6.

[10] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-Aware Recommender Systems. *AI Magazine*, 32(3):67–80, 3, 2011. DOI: 10.1609/aimag.v32i3.2364.

[11] G. Adomavicius and A. Tuzhilin. *Context-Aware Recommender Systems*. In *Recommender Systems Handbook*. Springer, 1st edition, 2010, pages 217–253. DOI: 10.1007/978-0-387-85820-3-7.

[12] G. Adomavicius and J. Zhang. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems*, 3(1):1–17, 2012. DOI: 10.1145/2151163.2151166.

[13] T. W. Adorno. *Introduction to the Sociology of Music*. Continuum, 1988.

[14] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 5–14. ACM, 2019. DOI: 10.1145/3331184.3331202.

[15] C. C. Aggarwal. *Ensemble-Based and Hybrid Recommender Systems*. In *Recommender Systems*. Springer, 2016, pages 199–224. DOI: 10.1007/978-3-319-29659-3-6.

[16] I. Ajzen and M. Fishbein. Attitude-Behavior Relations: A Theoretical Analysis and Review of Empirical Research. *Psychological Bulletin*, 84(5):888–918, 1977. DOI: 10.1037/0033-2909.84.5.888.

[17] M. Aljukhadar, S. Senecal, and C.-E. Daoust. Using Recommendation Agents to Cope with Information Overload. *International Journal of Electronic Commerce*, 17(2):41–70, 2012. DOI: 10.2753/jec1086-4415170202.

[18] A. Almahairi, K. Kastner, K. Cho, and A. Courville. Learning Distributed Representations from Reviews for Collaborative Filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 147–154. ACM, 2015. DOI: 10.1145/2792838.2800192.

[19] A. Althubaiti. Information bias in health research: Definition, pitfalls, and adjustment methods. *Journal of Multidisciplinary Healthcare*, 9:211, 2016. DOI: 10.2147/jmdh.s104807.

[20] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver. Rate it Again: Increasing Recommendation Accuracy by User Re-Rating. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, RecSys '09, pages 247–258. ACM, 2009. DOI: 10.1145/1639714.1639744.

[21] J. S. Andersen. Using the Echo Nest's automatically extracted music features for a musicological purpose. In *4th International Workshop on Cognitive Information Processing*, CIP '14, pages 1–6. IEEE, 2014. DOI: 10.1109/cip.2014.6844510.

[22] C. Anderson. *The long tail: How endless choice is creating unlimited demand*. Random House, 2007.

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

[24] I. Andjelkovic, D. Parra, and J. O'Donovan. Moodplay. Interactive mood-based music discovery and recommendation. In *Proceedings of the Conference on User Modeling Adaptation and Personalization*, UMAP '16, pages 275–279. ACM, 2016. DOI: 10.1145/2930238.2930280.

[25] V. W. Anelli, A. Bellogıén, A. Ferrara, D. Malitesta, F. A. Merra, C. Pomo, F. M. Donini, and T. D. Noia. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 2405–2414. ACM, 2021. DOI: 10.1145/3404835.3463245.

[26] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering Points to Identify the Clustering Structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, volume 28 of number 2 in *SIGMOD '99*, pages 49–60. ACM, 1999. DOI: 10.1145/304182.304187.

[27] A. Ankolekar and T. Sandholm. Foxtrot: A Soundtrack for Where You Are. In *Proceedings of Interacting with Sound Workshop on Exploring Context-Aware, Local and Social Audio Applications*, IwS '11, pages 26–31. ACM, 2011. DOI: 10.1145/2019335.2019341.

[28] J. Arévalo, J. R. Duque, and M. Creatura. A Missing Information Loss function for implicit feedback datasets, 2018. DOI: 10.48550/ARXIV.1805.00121.

[29] C. Argueta, F. H. Calderon, and Y.-S. Chen. Multilingual Emotion Classifier using Unsupervised Pattern Extraction from Microblog Data. *Intelligent Data Analysis*, 20(6):1477–1502, 2016. DOI: 10.3233/ida-140267.

[30] I. Avazpour, T. Pitakrat, L. Grunske, and J. Grundy. *Dimensions and Metrics for Evaluating Recommendation Systems*. In *Recommendation Systems in Software Engineering*. Springer, 2013, pages 245–273. DOI: 10.1007/978-3-642-45135-5-10.

[31] T. Ayaki, H. Yanagimoto, and M. Yoshioka. Recommendation from access logs with ensemble learning. *Artificial Life and Robotics*, 22(2):163–167, 2017. DOI: 10.1007/s10015-016-0346-x.

[32] A. B. Al-Badareen, M. H. Selamat, J. Din, M. A. Jabar, and S. Turaev. Software quality evaluation: User's view. *International Journal of Applied Mathematics and Informatics*, 5(3):200–207, 2011.

[33] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger. InCarMusic: Context-aware Music Recommendations in a Car. In *International Conference on Electronic Commerce and Web Technologies*, volume 85 of *Lecture Notes in Business Information Processing*, pages 89–100. Springer, 2011. DOI: 10.1007/978-3-642-23014-1-8.

[34] 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 '11, pages 301–304. ACM, 2011. DOI: 10.1145/2043932.2043988.

[35] D. Bamber. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology*, 12(4):387–415, 1975. DOI: 10.1016/0022-2496(75)90001-2.

[36] M. D. Barone, J. Bansal, and M. H. Woolhouse. Acoustic Features Influence Musical Choices Across Multiple Genres. *Frontiers in Psychology*, 8, 2017. DOI: 10.3389/fpsyg.2017.00931.

[37] M. Barthet, D. Marston, C. Baume, G. Fazekas, and M. B. Sandler. Design and Evaluation of Semantic Mood Models for Music Recommendation using Editorial Tags. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, ISMIR '13, pages 421–426. ISMIR, 2013.

[38] C. Bauer. Allowing for Equal Opportunities for Artists in Music Recommendation. In *Proceedings of the 1st Workshop on Designing Human-Centric Music Information Research Systems (wsHCMIR 2019), Satellite Event to ISMIR 2019*, pages 16–18, 2019.

[39] C. Bauer and M. Schedl. Global and country-specific mainstreaminess measures: Definitions, analysis, and usage for improving personalized music recommendation systems. *PLoS ONE*, 14(6), 2019. DOI: 10.1371/journal.pone.0217389.

[40] C. Bauer and E. Zangerle. Leveraging multi-method evaluation for multi-stakeholder settings. In *Proceedings of the 1st Workshop on the Impact of Recommender Systems co-located with 13th ACM Conference on Recommender Systems*, volume 2462 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[41] J. Beel, A. Collins, O. Kopp, L. W. Dietz, and P. Knoth. Online Evaluations for Everyone: Mr. DLib's Living Lab for Scholarly Recommendations. In *ECIR 2019: Advances in Information Retrieval - 41st European Conference on IR Research, Proceedings, Part II*, ECIR '19, pages 213–219. Springer, 2019.

[42] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, and B. Gipp. A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 7–14. ACM, 2013. DOI: 10.1145/2532508.2532511.

[43] J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research Paper Recommender Systems: A Literature Survey. *International Journal on Digital Libraries*, 17(4):305–338, 2015. DOI: 10.1007/s00799-015-0156-0.

[44] J. Beel and S. Langer. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *International Conference on Theory and Practice of Digital Libraries*, TPDL '15, pages 153–168. Springer, 2015. DOI: 10.1007/978-3-319-24592-8-12.

[45] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitinger, and A. Nürnberger. Research Paper Recommender System Evaluation: a Quantitative Literature Survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 15–22. ACM, 2013. DOI: 10.1145/2532508.2532512.

[46] A. Bellogin, P. Castells, and I. Cantador. Precision-Oriented Evaluation of Recommender Systems. An algorithmic comparison. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 333–336. ACM, 2011. DOI: 10.1145/2043932.2043996.

[47] A. Bellogıén and A. Said. Improving accountability in recommender systems research through reproducibility. *User Modeling and User-Adapted Interaction*, 31:941–977, 2021. DOI: 10.1007/s11257-021-09302-x.

[48] R. v. d. Berg, T. N. Kipf, and M. Welling. Graph Convolutional Matrix Completion. In *KDD'18 Deep Learning Day*, 2018. DOI: 10.48550/ARXIV.1706.02263.

[49] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow. Fairness in Recommendation Ranking through Pairwise Comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pages 2212–2220. ACM, 2019. DOI: 10.1145/3292500.3330745.

[50] A. J. Biega, K. P. Gummadi, and G. Weikum. Equity of Attention: Amortizing Individual Fairness in Rankings. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pages 405–414. ACM, 2018. DOI: 10.1145/3209978.3210063.

[51] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006, pages 424–429. DOI: 10.1007/978-0-387-45528-0.

[52] M. Blondel, A. Fujino, N. Ueda, and M. Ishihata. Higher-Order Factorization Machines. In *Advances in Neural Information Processing Systems*, Neurips '16, pages 3351–3359. Curran Associates, 2016.

[53] E. Bodner, I. Iancu, A. Gilboa, A. Sarel, A. Mazor, and D. Amir. Finding words for emotions: The reactions of patients with major depressive disorder towards various musical excerpts. *The Arts in Psychotherapy*, 34(2):142–150, 2007. DOI: 10.1016/j.aip.2006.12.002.

[54] D. Boer and R. Fischer. Towards a holistic model of functions of music listening across cultures: A culturally decentred qualitative approach. *Psychology of Music*, 40(2):179–200, 2011. DOI: 10.1177/0305735610381885.

[55] D. Bogdanov, M. Haro, F. Fuhrmann, E. Gómez, and P. Herrera. Content-based Music Recommendation based on User Preference Examples. In *Workshop on Music Recommendation and Discovery (Womrad 2010) at ACM Recommender Systems 2010*, 2010.

[56] T. Bogers, M. Koolen, B. Mobasher, C. Petersen, and A. Tuzhilin. Report on the 4th Workshop on Recommendation in Complex Environments (ComplexRec 2020). *SIGIR Forum*, 54(2), 2020. DOI: 10.1145/3483382.3483397.

[57] 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 '10, pages 63–70. ACM, 2010. DOI: 10.1145/1864708.1864724.

[58] P. Bonhard, C. Harries, J. McCarthy, and M. A. Sasse. Accounting for Taste: Using Profile Similarity to Improve Recommender Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1057–1066. ACM, 2006. DOI: 10.1145/1124772.1124930.

[59] S. Bostandjiev, J. O'Donovan, and T. Höllerer. TasteWeights: a Visual Interactive Hybrid Recommender System. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 35–42. ACM, 2012. DOI: 10.1145/2365952.2365964.

[60] K. Bosteels, E. Pampalk, and E. E. Kerre. Evaluating and Analysing Dynamic Playlist Generation Heuristics Using Radio Logs and Fuzzy Set Theory. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, ISMIR '09, pages 351–356. ISMIR, 2009.

[61] L. Bottou. Stochastic gradient learning in neural networks. In *Proceedings of the 4th International Neuro-Nimes Conference*. EC2, 1991.

[62] M. Braunhofer, M. Kaminskas, and F. Ricci. Location-aware Music Recommendation. *International Journal of Multimedia Information Retrieval*, 2(1):31–44, 2013. DOI: 10.1007/s13735-012-0032-2.

[63] 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 '11, pages 253–256. ACM, 2011. DOI: 10.1145/2043932.2043977.

[64] 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'98, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[65] E. Brynjolfsson, Y. J. Hu, and M. D. Smith. From Niches to Riches: Anatomy of the Long Tail. *MIT Sloan Management Review*, 47(4):67–71, 2006.

[66] R. Burke. Evaluating the Dynamic Properties of Recommendation Algorithms. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 225–228. ACM, 2010. DOI: 10.1145/1864708.1864753.

[67] R. Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[68] R. Burke. Multisided fairness for recommendation, 2017. DOI: 10.48550/ARXIV.1707.00093.

[69] R. Burke, H. Abdollahpouri, B. Mobasher, and T. Gupta. Towards Multi-Stakeholder Utility Evaluation of Recommender Systems. In *UMAP 2016 Extended Proceedings: 1st Workshop on Surprise, Opposition, and Obstruction in Adaptive and Personalized Systems*, volume 1618 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[70] R. Burke, N. Sonboli, and A. Ordonez-Gauger. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 202–214. PMLR, 2018.

[71] 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 International Conference on Multimedia*, MULTIMEDIA '07. ACM, 2007. DOI: 10.1145/1291233.1291369.

[72] R. A. Calvo and S. D'Mello. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010. DOI: 10.1109/t-affc.2010.1.

[73] D. T. Campbell. Factors Relevant to the Validity of Experiments in Social Settings. *Psychological Bulletin*, 54(4):297–312, 1957. DOI: 10.1037/h0040950.

[74] R. Cañamares, P. Castells, and A. Moffat. Offline Evaluation Options for Recommender Systems. *Information Retrieval*, 23(4):387–410, 2020. DOI: 10.1007/s10791-020-09371-3.

[75] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, 96(4):668–696, 4, 2008. DOI: 10.1109/jproc.2008.916370.

[76] P. Castells, N. Hurley, and S. Vargas. *Novelty and Diversity in Recommender Systems*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 603–646. DOI: 10.1007/978-1-0716-2197-4-16.

[77] I. Celik, I. Torre, F. Koceva, C. Bauer, E. Zangerle, and B. Knijnenburg. UMAP 2018 Intelligent User-Adapted Interfaces: Design and Multi-Modal Evaluation (IUadaptMe) Workshop Chairs' Welcome & Organization. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, pages 137–139. ACM, 2018. DOI: 10.1145/3213586.3226202.

[78] Ò. Celma. *Music Recommendation and Discovery*. Springer, 2010. DOI: 10.1007/978-3-642-13287-2.

[79] Ò. Celma. *Music Recommendation and Discovery*. Springer, 1st edition, 2010. DOI: 10.1007/978-3-642-13287-2.

[80] Ò. Celma and P. Cano. From hits to niches? Or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*. ACM, 2008. DOI: 10.1145/1722149.1722154.

[81] Ò. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, RecSys '08, pages 179–186. ACM, 2008. DOI: 10.1145/1454008.1454038.

[82] Ò. Celma Herrada et al. *Music recommendation and discovery in the long tail.* PhD thesis, Universitat Pompeu Fabra, 2009.

[83] T. C. Chalmers, H. Smith, B. Blackburn, B. Silverman, B. Schroeder, D. Reitman, and A. Ambroz. A method for assessing the quality of a randomized control trial. *Controlled Clinical Trials*, 2(1):31–49, 1981. DOI: 10.1016/0197-2456(81)90056-8.

[84] T. Chamorro-Premuzic and A. Furnham. Personality and music: Can traits explain how people use music in everyday life? *British Journal of Psychology*, 98(2):175–185, 2007. DOI: 10.1348/000712606x111177.

[85] Y.-L. Chao and S.-P. Lam. Measuring Responsible Environmental Behavior: Self-reported and Other-Reported Measures and Their Differences in Testing a Behavioral Model. *Environment and Behavior*, 43(1):53–71, 2009. DOI: 10.1177/0013916509350849.

[86] C. C. Chen, S.-Y. Shih, and M. Lee. Who should you follow? Combining learning to rank with social influence for informative friend recommendation. *Decision Support Systems*, 90:33–45, 2016. DOI: 10.1016/j.dss.2016.06.017.

[87] C.-M. Chen, M.-F. Tsai, J.-Y. Liu, and Y.-H. Yang. Music Recommendation Based on Multiple Contextual Similarity Information. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1 of *WI-IAT '13*, pages 65–72. IEEE, 2013. DOI: 10.1109/wi-iat.2013.10.

[88] 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 '13, pages 649–652. ACM, 2013. DOI: 10.1145/2502081.2502170.

[89] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He. Bias and Debias in Recommender System: A Survey and Future Directions, 2020. DOI: 10.48550/ARXIV.2010.03240.

[90] M. Chen and P. Liu. Performance Evaluation of Recommender Systems. *International Journal of Performability Engineering*, 13(8):1246–1256, 2017. DOI: 10.23940/ijpe.17.08.p7.12461256.

[91] P.-Y. Chen, Y.-C. Chou, and R. J. Kauffman. Community-Based Recommender Systems: Analyzing Business Models from a Systems Operator's Perspective. In *42nd Hawaii International Conference on System Sciences*, HICSS '09, pages 1–10. IEEE, 2009. DOI: 10.1109/hicss.2009.117.

[92] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist Prediction via Metric Embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 12 of *KDD '12*, pages 349–354. ACM, 2012. DOI: 10.1145/2339530.2339643.

[93] T. Chen and C. Guestrin. XGBoost. A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM, 2016. DOI: 10.1145/2939672.2939785.

[94] Y. Chen and J. Konstan. Online Field Experiments: A Selective Survey of Methods. *Journal of the Economic Science Association*, 1(1):29–42, 2015. DOI: 10.1007/s40881-015-0005-3.

[95] R. Cheng and B. Tang. A Music Recommendation System Based on Acoustic Features and User Personalities. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD '16, pages 203–213. Springer, 2016.

[96] Z. Cheng and J. Shen. Just-for-Me: An Adaptive Personalization System for Location-Aware Social Music Recommendation. In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 1267–1268. ACM, 2014. DOI: 10.1145/2578726.2578751.

[97] P. Chordia, M. Godfrey, and A. Rae. Extending Content-Based Recommendation: The Case of Indian Classical Music. In *Proceedings of the 8th International Society for Music Information Retrieval Conference*, ISMIR '08, pages 571–576. ISMIR, 2008.

[98] K. Christakopoulou, F. Radlinski, and K. Hofmann. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 815–824. ACM, 2016. DOI: 10.1145/2939672.2939746.

[99] C. W. Cleverdon, J. Mills, and E. M. Keen. Factors determining the performance of indexing systems, (Volume 1: Design). Technical report, College of Aeronautics, 1966.

[100] D. A. Cobb-Clark and S. Schurer. The stability of big-five personality traits. *Economic Letters*, 115(1):11–15, 2012. DOI: 10.1016/j.econlet.2011.11.015.

[101] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl. Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 585–592. ACM, 2003. DOI: 10.1145/642611.642713.

[102] S. P. Crain, K. Zhou, S.-H. Yang, and H. Zha. *Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond*. In *Mining Text Data*. Springer, 2012, pages 129–161. DOI: 10.1007/978-1-4614-3223-4-5.

[103] P. Cremonesi, F. Garzotto, S. Negro, A. V. Papadopoulos, and R. Turrin. Looking for "Good" Recommendations: A Comparative Evaluation of Recommender Systems. In *Human-Computer Interaction*, INTERACT '11, pages 152–168. Springer, 2011. DOI: 10.1007/978-3-642-23765-2-11.

[104] P. Cremonesi and D. Jannach. Progress in Recommender Systems Research: Crisis? What Crisis? *AI Magazine*, 42(3):43–54, 2021. DOI: 10.1609/aimag.v42i3.18145.

[105] P. Cremonesi, Y. Koren, and R. Turrin. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 39–46. ACM, 2010. DOI: 10.1145/1864708.1864721.

[106] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci. An Evaluation Methodology for Collaborative Recommender Systems. In *International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, AXMEDIS '08, pages 224–231. IEEE, 2008. DOI: 10.1109/axmedis.2008.13.

[107] J. W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2nd edition, 2003.

[108] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. Seven Pitfalls to Avoid when Running Controlled Experiments on the Web. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1105–1114. ACM, 2009. DOI: 10.1145/1557019.1557139.

[109] 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 '06. ISMIR, 2006.

[110] M. F. Dacrema, S. Boglio, P. Cremonesi, and D. Jannach. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Transactions on Information Systems*, 39(2):1–49, 2021. DOI: 10.1145/3434185.

[111] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The YouTube Video Recommendation System. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 293–296. ACM, 2010. DOI: 10.1145/1864708.1864770.

[112] R. A. Davis, K.-S. Lii, and D. N. Politis. *Remarks on Some Nonparametric Estimates of a Density Function*. In *Selected Works of Murray Rosenblatt*. Springer, 2011, pages 95–100. DOI: 10.1007/978-1-4419-8339-8-13.

[113] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro. *Semantics-Aware Content-Based Recommender Systems.* In *Recommender Systems Handbook*. Springer, 2nd edition, 2015, pages 119–159. DOI: 10.1007/978-1-4899-7637-6-4.

[114] Z. Dehghani Champiri, A. Asemi, and S. Siti Salwah Binti. Meta-analysis of evaluation methods and metrics used in context-aware scholarly recommender systems. *Knowledge and Information Systems*, 61(2):1147–1178, 2019. DOI: 10.1007/s10115-018-1324-5.

[115] M. del Carmen-Rodríguez-Hernández, S. Ilarri, R. Hermoso, and R. Trillo-Lado. DataGenCARS: A Generator of Synthetic Data for the Evaluation of Context-Aware Recommendation Systems. *Pervasive and Mobile Computing*, 38:516–541, 2017. DOI: 10.1016/j.pmcj.2016.09.020.

[116] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana. Content-Based Video Recommendation System Based on Stylistic Visual Features. *Journal on Data Semantics*, 5(2):99–113, 2016. DOI: 10.1007/s13740-016-0060-9.

[117] J. J. Deng, C. H. C. Leung, A. Milani, and L. Chen. Emotional States Associated with Music. Classification, prediction of changes, and consideration in recommendation. *ACM Transactions on Interactive Intelligent Systems*, 5(1):1–36, 2015. DOI: 10.1145/2723575.

[118] D. Deutsch. *Psychology of Music*. Academic Press, 3rd edition, 2013.

[119] F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. Evaluating Stochastic Rankings with Expected Exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM 2020, pages 275–284. ACM, 2020. DOI: 10.1145/3340531.3411962.

[120] P. R. Dickson. Person-Situation: Segmentation's Missing Link. *Journal of Marketing*, 46(4):56, 1982. DOI: 10.2307/1251362.

[121] E. Diener. Subjective well-being: The science of happiness and a proposal for a national index. *American Psychologist*, 55(1):34–43, 2000. DOI: 10.1037/0003-066x.55.1.34.

[122] K. Dinnissen and C. Bauer. Fairness in Music Recommender Systems: A Stakeholder-centered Mini Review. *Frontiers in Big Data*, 5, 2022. DOI: 10.3389/fdata.2022.913608.

[123] J. Donaldson. A Hybrid Social-Acoustic Recommendation System for Popular Music. In *Proceedings of the 1st ACM Conference on Recommender Systems*, RecSys '07, pages 187–190. ACM, 2007. DOI: 10.1145/1297231.1297271.

[124] S. Dooms, T. De Pessemier, and L. Martens. A User-Centric Evaluation of Recommender Algorithms for an Event Recommendation System. In *Proceedings of the Workshop on Human Decision Making in Recommender Systems (DecisionsRecSys'11) and User-Centric Evaluation of Recommender Systems and Their Interfaces-2 (UCERSTI 2)*, volume 811 of *CEUR Workshop Proceedings*, pages 67–73. CEUR-WS.org, 2011.

[125] S. Dooms, T. De Pessemier, and L. Martens. MovieTweetings: A Movie Rating Dataset Collected From Twitter. In *Workshop on Crowdsourcing and Human Computation for Recommender Systems*, CrowdRec '13, 2013.

[126] J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, 2005. DOI: 10.1002/aris.1440370108.

[127] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The Yahoo! Music Dataset and KDD-Cup'11. In *Proceedings of KDD Cup 2011 Competition*, volume 18, pages 3–18. JMLR, 2012.

[128] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226. ACM, 2012. DOI: 10.1145/2090236.2090255.

[129] C. Dwork and C. Ilvento. Group Fairness under Composition. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, FAT*2018. ACM, 2018.

[130] M. D. Ekstrand, R. Burke, and F. Diaz. Fairness and discrimination in recommendation and retrieval. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 576–577. ACM, 2019. DOI: 10.1145/3298689.3346964.

[131] M. D. Ekstrand and V. Mahant. Sturgeon and the Cool Kids: Problems with Random Decoys for top-N Recommender Evaluation. In *Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference*, FLAIRS '17, pages 639–644. AAAI, 2017.

[132] M. D. Ekstrand. Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011. DOI: 10.1561/1100000009.

[133] M. D. Ekstrand. LensKit for Python: Next-Generation Software for Recommender Systems Experiments. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 2999–3006. ACM, 2020. DOI: 10.1145/3340531.3412778.

[134] M. D. Ekstrand, A. Das, R. Burke, and F. Diaz. *Fairness in Recommender Systems.* In *Recommender Systems Handbook.* Springer, 3rd edition, 2022, pages 679–707. DOI: 10.1007/978-1-0716-2197-4-18.

[135] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan. User Perception of Differences in Recommender Algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 161–168. ACM, 2014. DOI: 10.1145/2645710.2645737.

[136] M. D. Ekstrand and D. Kluver. Exploring author gender in book rating and recommendation. *User Modeling and User-Adapted Interaction*, 31(3):377–420, 2021. DOI: 10.1007/s11257-020-09284-2.

[137] M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 133–140. ACM, 2011. DOI: 10.1145/2043932.2043958.

[138] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 172–186. PMLR, 2018.

[139] A. M. Elkahky, Y. Song, and X. He. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 278–288. ACM, 2015. DOI: 10.1145/2736277.2741667.

[140] Y. Fang and L. Si. Matrix Co-factorization for Recommendation with Rich Side Information and Implicit Feedback. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 65–69. ACM, 2011. DOI: 10.1145/2039320.2039330.

[141] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič. *Evaluating Group Recommender Systems*. In *SpringerBriefs in Electrical and Computer Engineering*. Springer, 2018. Chapter Evaluating Group Recommender Systems, pages 59–71. DOI: 10.1007/978-3-319-75067-5-3.

[142] I. Fernández-Tobıéas, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction*, 26(2-3):221–255, 2016. DOI: 10.1007/s11257-016-9172-z.

[143] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, and D. Jannach. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Transactions on Information Systems*, 39(2):1–49, 2021. DOI: 10.1145/3434185.

[144] M. Ferrari Dacrema, P. Cremonesi, and D. Jannach. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 101–109. ACM, 2019. DOI: 10.1145/3298689.3347058.

[145] A. Ferraro, X. Serra, and C. Bauer. Break the Loop: Gender Imbalance in Music Recommenders. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, CHIIR '21, pages 249–254. ACM, 2021. DOI: 10.1145/3406522.3446033.

[146] B. Ferwerda and M. Schedl. Enhancing Music Recommender Systems with Personality Information and Emotional States: A Proposal. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization*, volume 1181 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

[147] B. Ferwerda and M. Schedl. Investigating the Relationship Between Diversity in Music Consumption Behavior and Cultural Dimensions: A Cross-Country Analysis. In F. Cena, M. C. Desmarais, and D. Dicheva, editors, *Late-breaking Results, Posters, Demos, Doctoral Consortium and Workshops: Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalisation*, volume 1618 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[148] B. Ferwerda, M. Schedl, and M. Tkalčič. Personality & Emotional States: Understanding Users' Music Listening Needs. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015)*, volume 1388 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[149] B. Ferwerda, M. Tkalčič, and M. Schedl. Personality Traits and Music Genres. What do people prefer to listen to? In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 285–288. ACM, 2017. DOI: 10.1145/3079628.3079693.

[150] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist Generation Using Start and End Songs. In *Proceedings of the 9th International Society for Music Information Retrieval Conference*, ISMIR '08. ISMIR, 2008.

[151] U. Flick. *An Introduction to Qualitative Research*. SAGE Publications, 2014.

[152] C. Freudenthaler, L. Schmidt-Thieme, and S. Rendle. Bayesian Factorization Machines. In *Proceedings of NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011.

[153] B. J. Frey and D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, 2007. DOI: 10.1126/science.1136800.

[154] J. Freyne and S. Berkovsky. *Evaluating Recommender Systems for Supportive Technologies*. In *Human-Computer Interaction Series*. Springer, 2013, pages 195–217. DOI: 10.1007/978-1-4471-4778-7-8.

[155] M. Funk, A. Rozinat, E. Karapanos, A. Alves de Medeiros, and A. Koca. In situ evaluation of recommender systems: Framework and instrumentation. *International Journal of Human-Computer Studies*, 68(8):525–547, 2010. DOI: 10.1016/j.ijhcs.2010.01.002.

[156] S. Funk. Try this at home. *http://sifter. org/ simon/journal/2006*, 2006.

[157] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: a Free Recommender System Library. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 305–308. ACM, 2011. DOI: 10.1145/2043932.2043989.

[158] J. Garcia-Gathright, C. Hosey, B. S. Thomas, B. Carterette, and F. Diaz. Mixed Methods for Evaluating User Satisfaction. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 541–542. ACM, 2018. DOI: 10.1145/3240323.3241622.

[159] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. Offline and online evaluation of news recommender systems at swissinfo.ch. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 169–176. ACM, 2014. DOI: 10.1145/2645710.2645745.

[160] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 257–260. ACM, 2010. DOI: 10.1145/1864708.1864761.

[161] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. Offline A/B Testing for Recommender Systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 198–206. ACM, 2018. DOI: 10.1145/3159652.3159687.

[162] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the Long Tail. In *Proceedings of the 3rd ACM International Conference on Web search and data mining*, WSDM '10, pages 201–210. ACM, 2010. DOI: 10.1145/1718487.1718513.

[163] L. R. Goldberg. The structure of phenotypic personality traits. *American Psychologist*, 48(1):26–34, 1993. DOI: 10.1037/0003-066x.48.1.26.

[164] E. Gómez, C. Shui Zhang, L. Boratto, M. Salamó, and G. Ramos. Enabling cross-continent provider fairness in educational recommender systems. *Future Generation Computer Systems*, 127:435–447, 2022. DOI: 10.1016/j.future.2021.08.025.

[165] C. A. Gomez-Uribe and N. Hunt. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, 6(4):1–19, 2016. DOI: 10.1145/2843948.

[166] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. Comparing and Combining Sentiment Analysis Methods. In *Proceedings of the 1st ACM Conference on Online Social Networks*, COSN '13, pages 27–38. ACM, 2013. DOI: 10.1145/2512938.2512951.

[167] B. Gong, M. Kaya, and N. Tintarev. *Contextual Personalized Re-Ranking of Music Recommendations through Audio Features.* Master's thesis, TU Delft, 2020.

[168] S. N. Goodman, D. Fanelli, and J. P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12, 2016. DOI: 10.1126/scitranslmed.aaf5027.

[169] F. Gouyon, G. Widmer, X. Serra, and A. Flexer. Acoustic Cues to Beat Induction: A Machine Learning Perspective. *Internal Audit Handbook*, 24(2):177–188, 2006. DOI: 10.1525/mp.2006.24.2.177.

[170] A. G. Greenwald, A. R. Pratkanis, M. R. Leippe, and M. H. Baumgardner. Under what conditions does theory obstruct research progress? *Psychological Review*, 93(2):216–229, 1986. DOI: 10.1037/0033-295x.93.2.216.

[171] J. Gross. *Emotion Regulation: Conceptual and Empirical Foundations.* In *Handbook of emotion regulation.* The Guilford Press, 2nd edition, 2007, pages 1–19.

[172] A. Grover and J. Leskovec. node2vec. Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864. ACM, 2016. DOI: 10.1145/2939672.2939754.

[173] A. Gunawardana and G. Shani. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.

[174] A. Gunawardana, G. Shani, and S. Yogev. *Evaluating Recommender Systems.* In *Recommender Systems Handbook.* Springer, 3rd edition, 2022, pages 547–601. DOI: 10.1007/978-1-0716-2197-4-15.

[175] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith. LibRec: A Java Library for Recommender Systems. In *UMAP 2015 Extended Proceedings*, volume 1388 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2015.

[176] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1725–1731. AAAI, 2017. DOI: 10.5555/3172077.3172127.

[177] R. Haas and V. Brandes. *Music that works.* Springer, 2009. DOI: 10.1007/978-3-211-75121-3.

[178] 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, 2009. DOI: 10.1007/s11042-009-0332-6.

[179] D. J. Hargreaves and A. C. North. The Functions of Music in Everyday Life: Redefining the Social in Music Psychology. *Psychology of Music*, 27(1):71–83, 1999. DOI: 10.1177/0305735699271007.

[180] N. Hariri, B. Mobasher, and R. Burke. Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 131–138. ACM, 2012. DOI: 10.1145/2365952.2365979.

[181] F. M. Harper and J. A. Konstan. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 2016. DOI: 10.1145/2827872.

[182] D. Hauger and M. Schedl. Exploring Geospatial Music Listening Patterns in Microblog Data. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 133–146. Springer, 2012.

[183] 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 Society for Music Information Retrieval Conference*, ISMIR '13. ISMIR, 2013.

[184] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 639–648. ACM, 2020.

[185] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182. ACM, 2017. DOI: 10.1145/3038912.3052569.

[186] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182. ACM, 2017. DOI: 10.1145/3038912.3052569.

[187] J. F. Helliwell, R. Layard, and J. Sachs. *World Happiness Report*. Sustainable Development Solutions Network, 2016.

[188] J. Herlocker, J. A. Konstan, and J. Riedl. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*, 5(4):287–310, 2002. DOI: 10.1023/a:1020443909834.

[189] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer supported cooperative work*, CSCW '00, pages 241–250. ACM, 2000. DOI: 10.1145/358916.358995.

[190] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004. DOI: 10.1145/963770.963772.

[191] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks, 2015. DOI: 10.48550/ARXIV.1511.06939.

[192] G. Hofstede, G. J. Hofstede, and M. Minkov. *Cultures and Organizations: Software of the Mind*, volume 2. McGraw-Hill, 1991.

[193] G. Hofstede, G. J. Hofstede, and M. Minkov. *Cultures and Organizations: Software of the mind*, volume 3rd, revised. McGraw-Hill, 2010.

[194] G. H. Hofstede. *Culture's Consequences. Comparing Values, Behaviors, Institutes and Organizations across Nations.* Macat Library, 2018, page 475. DOI: 10.4324/9781912128334.

[195] N. P. C. Hong. Reproducibility Badging and Definitions: A Recommended Practice of the National Information Standards Organization, 2021. DOI: 10.3789/niso-rp-31-2021.

[196] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177. ACM, 2004. DOI: 10.1145/1014052.1014073.

[197] R. Hu and P. Pu. Helping Users Perceive Recommendation Diversity. In *Proceedings of the Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), at the 5th ACM International Conference on Recommender Systems (RecSys 2011)*, CEUR Workshop Proceedings, pages 43–50. CEUR-WS.org, 2011.

[198] Y. Hu and M. Ogihara. NextOne Player: A Music Recommendation System Based on User Behavior. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ISMIR '11. ISMIR, 2011.

[199] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272. IEEE, 2008. DOI: 10.1109/icdm.2008.22.

[200] N. Hug. Surprise, a Python library for recommender systems. http://surpriselib.com, 2017.

[201] N. Hurley and M. Zhang. Novelty and Diversity in Top-N Recommendation—Analysis and Evaluation. *ACM Transactions on Internet Technology*, 10(4):1–30, 2011. DOI: 10.1145/1944339.1944341.

[202] B. Hutchinson and M. Mitchell. 50 Years of Test (Un)fairness: Lessons for Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT*'19, pages 49–58. ACM, 2019. DOI: 10.1145/3287560.3287600.

[203] C. Hutto and E. Gilbert. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8 of number 1, pages 216–225, 2014.

[204] P. Jaccard. The Distribution Of The Flora In The Alpine Zone.1. *New Phytologist*, 11(2):37–50, 1912. DOI: 10.1111/j.1469-8137.1912.tb05611.x.

[205] D. Jannach and G. Adomavicius. Price and Profit Awareness in Recommender Systems. In *1st International Workshop on Value-Aware and Multistakeholder Recommendation*, VAMS '17. arXiv, 2017. DOI: 10.48550/arXiv.1707.08029.

[206] D. Jannach and G. Adomavicius. Recommendations with a Purpose. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 7–10. ACM, 2016. DOI: 10.1145/2959100.2959186.

[207] D. Jannach and C. Bauer. Escaping the McNamara Fallacy: Towards more Impactful Recommender Systems Research. *AI Magazine*, 41(4):79–95, 2020. DOI: 10.1609/aimag.v41i4.5312.

[208] D. Jannach and M. Jugovac. Measuring the Business Value of Recommender Systems. *ACM Transactions on Management Information Systems*, 10(4), 2019. DOI: 10.1145/3370082.

[209] D. Jannach, I. Kamehkhosh, and G. Bonnin. Analyzing the Characteristics of Shared Playlists for Music Recommendation. In *Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) co-located with the 8th ACM Conference on Recommender Systems (RecSys 2014)*, volume 1271 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

[210] D. Jannach, L. Lerche, and I. Kamehkhosh. Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 187–194. ACM, 2015. DOI: 10.1145/2792838.2800182.

[211] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015. DOI: 10.1007/s11257-015-9165-3.

[212] D. Jannach, A. Manzoor, W. Cai, and L. Chen. A Survey on Conversational Recommender Systems. *ACM Computing Surveys*, 54(5), 2021. DOI: 10.1145/3453154.

[213] D. Jannach, B. Mobasher, and S. Berkovsky. Research directions in session-based and sequential recommendation: A preface to the special issue. *User Modeling and User-Adapted Interaction*, 30(4):609–616, 2020. DOI: 10.1007/s11257-020-09274-4.

[214] D. Jannach, P. Pu, F. Ricci, and M. Zanker. Recommender Systems: Trends and Frontiers. *AI Magazine*, 43(2):145–150, 2022. DOI: 10.1002/aaai.12050.

[215] D. Jannach, O. S. Shalom, and J. A. Konstan. Towards More Impactful Recommender Systems Research. In *Proceedings of the 1st Workshop on the Impact of Recommender Systems*, volume 2462 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[216] D. Jannach and M. Zanker. *Value and Impact of Recommender Systems*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 519–546. DOI: 10.1007/978-1-0716-2197-4-14.

[217] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender systems: An Introduction*. Cambridge University Press, 2010.

[218] D. Jannach, M. Zanker, M. Ge, and M. Gröning. Recommender Systems in Computer Science and Information Systems – A Landscape of Research. In *Proceedings of the International Conference on Electronic Commerce and Web Technologies*, EC-Web '12, pages 76–87. Springer, 2012. DOI: 10.1007/978-3-642-32273-0-7.

[219] K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. DOI: 10.1145/582415.582418.

[220] G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender*, HetRec '10, pages 47–51. ACM, 2010. DOI: 10.1145/1869446.1869453.

[221] M. Jesse and D. Jannach. Digital nudging with recommender systems: Survey and future directions. *Computers in Human Behavior Reports*, 3, 2021. DOI: 10.1016/j.chbr.2020.100052.

[222] T. Joachims, B. London, Y. Su, A. Swaminathan, and L. Wang. Recommendations as Treatments: *AI Magazine*, 42(3):19–30, 2021. DOI: 10.1609/aaai.12014.

[223] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986. DOI: 10.1007/978-1-4757-1904-8.

[224] 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 '16, pages 43–50. ACM, 2016. DOI: 10.1145/2959100.2959134.

[225] S.-G. Jung, J. Salminen, S. A. Chowdhury, D. Ramirez Robillos, and B. J. Jansen. Things Change: Comparing Results Using Historical Data and User Testing for Evaluating a Recommendation Task. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20. ACM, 2020. DOI: 10.1145/3334480.3382945.

[226] P. N. Juslin and J. A. Sloboda. Handbook of Music and Emotion. *Theory and Research*, 2001.

[227] P. N. Juslin and P. Laukka. Expression, Perception, and Induction of Musical Emotions: A Review and a Questionnaire Study of Everyday Listening. *Journal of New Music Research*, 33(3):217–238, 2004. DOI: 10.1080/0929821042000317813.

[228] M. Kamalzadeh, D. Baur, and T. Möller. A Survey on Music Listening and Management Behaviours. In *Proceedings of the 13th International Symposium on Music Information Retrieval*, ISMIR '12. ISMIR, 2012.

[229] I. Kamehkhosh and D. Jannach. User Perception of Next-Track Music Recommendations. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 113–121. ACM, 2017. DOI: 10.1145/3079628.3079668.

[230] M. Kaminskas and D. Bridge. Diversity, Serendipity, Novelty, and Coverage. A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems*, 7(1):1–42, 2017. DOI: 10.1145/2926720.

[231] M. Kaminskas, I. Fernández-Tobıéas, F. Ricci, and I. Cantador. Knowledge-based Music Retrieval for Places of Interest. In *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, MIRUM '12, pages 19–24. ACM, 2012. DOI: 10.1145/2390848.2390854.

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

[233] M. Kaminskas and F. Ricci. Location-Adapted Music Recommendation Using Tags. In *User Modeling, Adaption and Personalization*, UMAP '11, pages 183–194. Springer, 2011. DOI: 10.1007/978-3-642-22362-4-16.

[234] 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 '13, pages 17–24. ACM, 2013. DOI: 10.1145/2507157.2507180.

[235] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: n-Dimensional Tensor Factorization for Context-Aware Collaborative Filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 79–86. ACM, 2010. DOI: 10.1145/1864708.1864727.

[236] J. Katukuri, T. Könik, R. Mukherjee, and S. Kolay. Recommending Similar Items in Large-scale Online Marketplaces. In *IEEE International Conference on Big Data*, Big Data '14, pages 868–876. IEEE, 2014. DOI: 10.1109/bigdata.2014.7004317.

[237] B. B. Kawulich. Participant observation as a data collection method. *Forum: Qualitative Sozialforschung/Forum: Qualitative Social Research*, 6(2), 2005. DOI: 10.17169/fqs-6.2.466.

[238] M. G. Kendall. A New Measure Of Rank Correlation. *Biometrika*, 30(1-2):81–93, 1938. DOI: 10.1093/biomet/30.1-2.81.

[239] S. Khusro, Z. Ali, and I. Ullah. Recommender Systems: Issues, Challenges, and Research Opportunities. In *Information Science and Applications 2016*, ICISA '16, pages 1179–1189. Springer, 2016. DOI: 10.1007/978-981-10-0557-2-112.

[240] Y. Kim, L. M. Aiello, and D. Quercia. PepMusic: Motivational qualities of songs for daily activities. *EPJ Data Science*, 9(1):13, 2020. DOI: 10.1140/epjds/s13688-020-0221-9.

[241] 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 Society for Music Information Retrieval Conference*, ISMIR '02, pages 209–214. ISMIR, 2002.

[242] P. Knees and M. Schedl. A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(1):1–21, 2013. DOI: 10.1145/2542205.2542206.

[243] P. Knees and M. Schedl. Music Retrieval and Recommendation. A tutorial overview. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 1133–1136. ACM, 2015. DOI: 10.1145/2766462.2767880.

[244] P. Knees and M. Schedl. *Music Similarity and Retrieval*. Springer, 2016. DOI: 10.1007/978-3-662-49722-7.

[245] B. Knijnenburg, L. Meesters, P. Marrow, and D. Bouwhuis. User-Centric Evaluation Framework for Multimedia Recommender Systems. In *International Conference on User Centric Media*, volume 40 of *UCMEDIA '09*, pages 366–369. Springer, 2010. DOI: 10.1007/978-3-642-12630-7-47.

[246] B. P. Knijnenburg, N. J. Reijmer, and M. C. Willemsen. Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 141–148. ACM, 2011. DOI: 10.1145/2043932.2043960.

[247] B. P. Knijnenburg and M. C. Willemsen. *Evaluating Recommender Systems with User Experiments*. In *Recommender Systems Handbook*. Springer, 2nd edition, 2015, pages 309–352. DOI: 10.1007/978-1-4899-7637-6-9.

[248] B. P. Knijnenburg and M. C. Willemsen. Understanding the Effect of Adaptive Preference Elicitation Methods on User Satisfaction of a Recommender System. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, RecSys '09, pages 381–384. ACM, 2009. DOI: 10.1145/1639714.1639793.

[249] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction*, 22(4–5):441–504, 2012. DOI: 10.1007/s11257-011-9118-4.

[250] B. P. Knijnenburg, M. C. Willemsen, and A. Kobsa. A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 321–324. ACM, 2011. DOI: 10.1145/2043932.2043993.

[251] A. Kobsa. User modeling: Recent work, prospects and hazards. *Human Factors in Information Technology*, 10:111, 1993.

[252] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1168–1176. ACM, 2013. DOI: 10.1145/2487575.2488217.

[253] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled experiments on the web: Survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2008. DOI: 10.1007/s10618-008-0114-1.

[254] J. Konstan and L. Terveen. Human-Centered Recommender Systems: Origins, Advances, Challenges, and Opportunities. *AI Magazine*, 42(3):31–42, 2021. DOI: 10.1609/aimag.v42i3.18142.

[255] J. A. Konstan, R. Burke, and E. C. Malthouse. Towards an Experimental News User Community as Infrastructure for Recommendation Research. In *Proceedings of the 9th International Workshop on News Recommendation and Analytics*, volume 3143 of *CEUR Workshop Proceedings*, pages 43–46. CEUR-WS.org, 2021.

[256] J. A. Konstan and J. Riedl. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1–2):101–123, 2012. DOI: 10.1007/s11257-011-9112-x.

[257] S. L. Koole. The psychology of emotion regulation: An integrative review. *Cognition Emotion*, 23(1):4–41, 2009. DOI: 10.1080/02699930802619031.

[258] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 447–456. ACM, 2009. DOI: 10.1145/1557019.1557072.

[259] Y. Koren. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. *ACM Transactions on Knowledge Discovery from Data*, 4(1):1–24, 2010. DOI: 10.1145/1644873.1644874.

[260] Y. Koren. Factorization Meets the Neighborhood. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434. ACM, 2008. DOI: 10.1145/1401890.1401944.

[261] Y. Koren and R. Bell. *Advances in Collaborative Filtering*. In *Recommender Systems Handbook*. Springer, 1st edition, 2010, pages 145–186. DOI: 10.1007/978-0-387-85820-3-5.

[262] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, 2009. DOI: 10.1109/mc.2009.263.

[263] Y. Koren, S. Rendle, and R. Bell. *Advances in Collaborative Filtering*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 91–142. DOI: 10.1007/978-1-0716-2197-4-3.

[264] C. Kormos and R. Gifford. The validity of self-report measures of proenvironmental behavior: A meta-analytic review. *Journal of Environmental Psychology*, 40:359–371, 2014. DOI: 10.1016/j.jenvp.2014.09.003.

[265] E. Kouloumpis, T. Wilson, and J. Moore. Twitter Sentiment Analysis: The Good, the Bad and the OMG! In *Proceedings of the International AAAI Conference Weblogs and Social Media*, ICWSM '11, pages 538–541. AAAI, 2011.

[266] D. Kowald, S. Kopeinik, and E. Lex. The TagRec Framework as a Toolkit for the Development of Tag-Based Recommender Systems. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 23–28. ACM, 2017. DOI: 10.1145/3099023.3099069.

[267] D. Kowald, P. Müllner, E. Zangerle, C. Bauer, M. Schedl, and E. Lex. Support the Underground: Characteristics of Beyond-Mainstream Music Listeners. *EPJ Data Science*, 10(1):1–26, 2021. DOI: 10.1140/epjds/s13688-021-00268-9.

[268] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, 1964. DOI: 10.1007/bf02289694.

[269] M. Kunaver and T. Požrl. Diversity in recommender systems—A survey. *Knowledge-based Systems*, 123:154–162, 2017. DOI: 10.1016/j.knosys.2017.02.009.

[270] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600. ACM, 2010. DOI: 10.1145/1772690.1772751.

[271] D. Lamprecht, M. Strohmaier, and D. Helic. A method for evaluating discoverability and navigability of recommendation algorithms. *Computational Social Networks*, 4(1):9, 2017. DOI: 10.1186/s40649-017-0045-3.

[272] H. A. Landsberger. *Hawthorne Revisited: Management and the Worker, Its Critics, and Developments in Human Relations in Industry*. Cornell University Press, 1958.

[273] A. Laplante. Improving Music Recommender Systems: What Can We Learn from Research on Music Tastes? In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ISMIR '14. ISMIR, 2014.

[274] D. Lee and K. Hosanagar. Impact of Recommender Systems on Sales Volume and Diversity. In *Proceedings of the International Conference on Information Systems*, ICIS '14. AIS, 2014.

[275] J. H. Lee and J. S. Downie. Survey of Music Information Needs, Uses, and Seeking Behaviours: Preliminary Findings. In *Proceedings of the 5th International Society for Music Information Retrieval Conference*, ISMIR '04. ISMIR, 2004.

[276] 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 Society for Music Information Retrieval Conference*, ISMIR '16. ISMIR, 2016.

[277] K. Lee and K. Lee. My Head is Your Tail: Applying Link Analysis on Long-Tailed Music Listening Behavior for Music Recommendation. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 213–220. ACM, 2011. DOI: 10.1145/2043932.2043971.

[278] K. Lerman and R. Ghosh. Information Contagion: an Empirical Study of the Spread of News on Digg and Twitter Social Networks. In *Proceedings of 4th International Conference on Weblogs and Social Media*, ICWSM '10, pages 90–97. AAAI, 2010.

[279] O. Lesota, A. Melchiorre, N. Rekabsaz, S. Brandl, D. Kowald, E. Lex, and M. Schedl. Analyzing Item Popularity Bias of Music Recommender Systems: Are Different Genders Equally Affected? In *Proceedings of the 15th ACM Conference on Recommender Systems*, volume 12036 of *Lecture Notes in Computer Science*, pages 35–42. ACM, 2021. DOI: 10.1145/3460231.3478843.

[280] M. Levy and M. Sandler. Learning Latent Semantic Models for Music from Social Tags. *Journal of New Music Research*, 37(2):137–150, 2008. DOI: 10.1080/09298210802479292.

[281] E. Lex, D. Kowald, and M. Schedl. Modeling Popularity and Temporal Drift of Music Genre Preferences. *Transactions of the International Society for Music Information Retrieval*, 3(1):17–30, 2020. DOI: 10.5334/tismir.39.

[282] 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. DOI: 10.1016/j.jesp.2013.03.013.

[283] L. Li, W. Chu, J. Langford, and R. E. Schapire. A Contextual-Bandit Approach toPersonalized News Article Recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670. ACM, 2010. DOI: 10.1145/1772690.1772758.

[284] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, pages 1754–1763. ACM, 2018. DOI: 10.1145/3219819.3220023.

[285] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 689–698. ACM, 2018. DOI: 10.1145/3178876.3186150.

[286] Y. Liang and M. C. Willemsen. The role of preference consistency, defaults and musical expertise in users' exploration behavior in a genre exploration recommender. In *Fifteenth ACM Conference on Recommender Systems*, pages 230–240. ACM, 2021. DOI: 10.1145/3460231.3474253.

[287] A. Liaw and M. Wiener. Classification and Regression by randomForest. *R news*, 2(3):18–22, 2002.

[288] Y.-P. Lin, Y.-H. Yang, and T.-P. Jung. Fusion of electroencephalographic dynamics and musical contents for estimating emotional responses in music listening. *Frontiers in Neuroscience*, 8(94):1–14, 2014. DOI: 10.3389/fnins.2014.00094.

[289] C. Liu, H. S. Alavi, E. Costanza, S. Zhai, W. Mackay, and W. Moncur. Rigor, Relevance and Impact: The Tensions and Trade-Offs Between Research in the Lab and in the Wild. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19. ACM, 2019. DOI: 10.1145/3290607.3311744.

[290] J.-Y. Liu, S.-Y. Liu, and Y.-H. Yang. LJ2M dataset: Toward better understanding of music listening behavior and user mood. In *IEEE International Conference on Multimedia and Expo*, ICME '14, pages 1–6. IEEE, 2014. DOI: 10.1109/icme.2014.6890172.

[291] M. Liu, X. Hu, and M. Schedl. Artist Preferences and Cultural, Socio-Economic Distances Across Countries: A Big Data Perspective. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ISMIR '17, pages 103–111. ISMIR, 2017.

[292] M. Liu, X. Hu, and M. Schedl. The relation of culture, socio-economics, and friendship to music preferences: A large-scale, cross-country study. *PLoS ONE*, 13(12), 2018. DOI: 10.1371/journal.pone.0208186.

[293] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang. Unifying Explicit and Implicit Feedback for Collaborative Filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1445–1448. ACM, 2010. DOI: 10.1145/1871437.1871643.

[294] N.-H. Liu. Comparison of content-based music recommendation using different distance estimation methods. *Applied Intelligence*, 38(2):160–174, 2012. DOI: 10.1007/s10489-012-0363-y.

[295] S. Liu, X. Tu, and R. Li. Unifying Explicit and Implicit Feedback for Top-N Recommendation. In *IEEE 2nd International Conference on Big Data Analysis*, ICBDA '17, pages 35–39. IEEE, 2017. DOI: 10.1109/icbda.2017.8078860.

[296] B. Logan. Content-based Playlist Generation: Exploratory Experiments. In *Proceedings of the 3rd International Symposium on Music Information Retrieval*, ISMIR '02, pages 295–296. ISMIR, 2002.

[297] B. Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *Proceedings of the 1st International Symposium on Music Information Retrieval*, ISMIR '00, pages 1–11. ISMIR, 2000.

[298] A. J. Lonsdale and A. C. North. Why do we listen to music? A uses and gratifications analysis. Music uses and gratifications. *British Journal of Psychology*, 102(1):108–134, 2011. DOI: 10.1348/000712610x506831.

[299] P. Lops, M. de Gemmis, and G. Semeraro. *Content-based Recommender Systems: State of the Art and Trends*. In *Recommender Systems Handbook*. Springer, 1st edition, 2010, pages 73–105. DOI: 10.1007/978-0-387-85820-3-3.

[300] F. Lu and N. Tintarev. A Diversity Adjusting Strategy with Personality for Music Recommendation. In *Proceedings of the 5th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2018, co-located with ACM Conference on Recommender Systems (RecSys 2018)*, volume 2225 of *CEUR Workshop Proceedings*, pages 7–14. CEUR-WS.org, 2018.

[301] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 462–466. ACM, 2019. DOI: 10.1145/3298689.3347041.

[302] X. Luo, M. Zhou, Y. Xia, and Q. Zhu. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014. DOI: 10.1109/tii.2014.2308433.

[303] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: Social Recommendation using Probabilistic Matrix Factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Mining*, CIKM '08, pages 931–940. ACM, 2008. DOI: 10.1145/1458082.1458205.

[304] C. Matt, T. Hess, and C. Weiß. A factual and perceptional framework for assessing diversity effects of online recommender systems. *Internet Research*, 29(6):1526–1550, 2019. DOI: 10.1108/intr-06-2018-0274.

[305] R. R. McCrae and O. P. John. An Introduction to the Five-Factor Model and Its Applications. *Journal of Personality*, 60(2):175–215, 2, 1992. DOI: 10.1111/j.1467-6494.1992.tb00970.x.

[306] 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. DOI: 10.1109/tasl.2012.2199109.

[307] B. McFee and G. R. Lanckriet. Hypergraph Models of Playlist Dialects. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ISMIR '12, pages 343–348. ISMIR, 2012.

[308] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, and R. Mehrotra. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In RecSys '18, pages 31–39. ACM, 2018. DOI: 10.1145/3240323.3240354.

[309] L. McInnes and J. Healy. Accelerated Hierarchical Density Based Clustering. In *IEEE International Conference on Data Mining Workshops*, ICDMW '17, pages 33–42. IEEE, 2017. DOI: 10.1109/icdmw.2017.12.

[310] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017. DOI: 10.21105/joss.00205.

[311] L. McInnes, J. Healy, N. Saul, and L. Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861, 2018. DOI: 10.21105/joss.00861.

[312] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, Inc., 2000. DOI: 10.1002/0471721182.

[313] S. M. McNee, J. Riedl, and J. A. Konstan. Making Recommendations Better: An Analytic Model for HumanRecommender Interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1103–1108. ACM, 2006. DOI: 10.1145/1125451.1125660.

[314] M. McVicar, T. Freeman, and T. De Bie. Mining the Correlation between Lyrical and Audio Features and the Emergence of Mood. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ISMIR '11, pages 783–788. ISMIR, 2011.

[315] R. Mehrotra, A. Anderson, F. Diaz, A. Sharma, H. Wallach, and E. Yilmaz. Auditing Search Engines for Differential Satisfaction Across Demographics. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 626–633. ACM, 2017. DOI: 10.1145/3041021.3054197.

[316] A. B. Melchiorre, E. Zangerle, and M. Schedl. Personality Bias of Music Recommendation Algorithms. In *Fourteenth ACM Conference on Recommender Systems*, RecSys '20, pages 533–538. ACM, 2020. DOI: 10.1145/3383313.3412223.

[317] A. N. Mikhail Trofimov. tffm: Tensorflow implementation of an arbitrary order Factorization Machine. https://github.com/geffy/tffm, 2016.

[318] M. Millecamp, N. N. Htun, Y. Jin, and K. Verbert. Controlling Spotify Recommendations. Effects of personal characteristics on music recommender user interfaces. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, pages 101–109. ACM, 2018. DOI: 10.1145/3209219.3209223.

[319] G. A. Miller. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998. DOI: 10.7551/mitpress/7287.001.0001.

[320] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956. DOI: 10.1037/h0043158.

[321] R. Miotto, L. Barrington, and G. Lanckriet. Improving Auto-tagging by Modeling Semantic Co-occurrences. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, ISMIR '10. ISMIR, 2010.

[322] A. Mnih and G. Hinton. A Scalable Hierarchical Distributed Language Model. In *Proceedings of the International Conference Neural Information Processing Systems*, NIPS'08, pages 1081–1088. Curran Associates, 2008.

[323] J. L. Moore, S. Chen, D. Turnbull, and T. Joachims. Taste Over Time: The Temporal Dynamics of User Preferences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, ISMIR '13, pages 401–406. ISMIR, 2013.

[324] J. L. Moore, T. Joachims, and D. Turnbull. Taste Space Versus the World: An Embedding Analysis of Listening Habits and Geography. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ISMIR '14, pages 439–444. ISMIR, 2014.

[325] M. Moscati, E. Parada-Cabaleiro, Y. Deldjoo, E. Zangerle, and M. Schedl. Music4All-Onion – A Large-Scale Multi-Faceted Content-Centric Music Recommendation Dataset. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pages 4339–4343. ACM, 2022. DOI: 10.1145/3511808.3557656.

[326] T. Murakami, K. Mori, and R. Orihara. Metrics for Evaluating the Serendipity of Recommendation Lists. In *Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence*, JSAI'07, pages 40–46. Springer, 2008. DOI: 10.1007/978-3-540-78197-4-5.

[327] F. Murtagh and P. Legendre. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*, 31(3):274–295, 2014. DOI: 10.1007/s00357-014-9161-z.

[328] C. Musto, M. d. Gemmis, P. Lops, F. Narducci, and G. Semeraro. *Semantics and Content-Based Recommendations*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 251–298. DOI: 10.1007/978-1-0716-2197-4-7.

[329] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, and V. S. T. Wilson. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320. ACL, 2013.

[330] O. Nalmpantis and C. Tjortjis. The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender Systems. In *Engineering Applications of Neural Networks*, pages 498–507. Springer, 2017.

[331] A. Neviarouskaya, H. Prendinger, and M. Ishizuka. SentiFul: A Lexicon for Sentiment Analysis. *IEEE Transactions on Affective Computing*, 2(1):22–36, 2011. DOI: 10.1109/t-affc.2011.1.

[332] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

[333] T. V. Nguyen, A. Karatzoglou, and L. Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 63–72. ACM, 2014. DOI: 10.1145/2600428.2609623.

[334] F. Å. Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs, 2011. DOI: 10.48550/ARXIV.1103.2903.

[335] A. N. Nikolakopoulos, X. Ning, C. Desrosiers, and G. Karypis. *Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 39–89. DOI: 10.1007/978-1-0716-2197-4-2.

[336] X. Ning and G. Karypis. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *IEEE 11th International Conference on Data Mining*, ICDM '11, pages 497–506. IEEE, 2011. DOI: 10.1109/icdm.2011.134.

[337] X. Ning and G. Karypis. Sparse Linear Methods with Side Information for Top-n Recommendations. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 155–162. ACM, 2012. DOI: 10.1145/2365952.2365983.

[338] M. Nørgaard and K. Hornbæk. What do Usability Evaluators do in Practice? An Explorative Study of Think-Aloud Testing. In *Proceedings of the 6th ACM Conference on Designing Interactive systems*, DIS '06, pages 209–218. ACM, 2006. DOI: 10.1145/1142405.1142439.

[339] 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 '12, pages 154–168. Springer, 2012.

[340] A. Pacuk, P. Sankowski, K. Węgrzycki, A. Witkowski, and P. Wygocki. RecSys Challenge 2016. Job recommendations based on preselection of offers and gradient boosting. In *Proceedings of the Recommender Systems Challenge*, 10:1–10:4. ACM, 2016. DOI: 10.1145/2987538.2987544.

[341] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-Class Collaborative Filtering. In *Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 502–511. IEEE, 2008. DOI: 10.1109/icdm.2008.16.

[342] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008. DOI: 10.1561/1500000011.

[343] S.-T. Park and D. M. Pennock. Applying Collaborative Filtering Techniques to Movie Search for Better Ranking and Browsing. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 550–559. ACM, 2007. DOI: 10.1145/1281192.1281252.

[344] Y.-J. Park and A. Tuzhilin. The Long Tail of Recommender Systems and How to Leverage It. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 11–18. ACM, 2008. DOI: 10.1145/1454008.1454012.

[345] D. Parra and X. Amatriain. Walk the Talk: Analyzing the Relation between Implicit and Explicit Feedback for Preference Elicitation. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*, UMAP '11, pages 255–268. Springer, 2011. DOI: 10.1007/978-3-642-22362-4-22.

[346] D. Parra and S. Sahebi. Recommender Systems: Sources of Knowledge and Evaluation Metrics. In *Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*, pages 149–175. Springer, 2013. DOI: 10.1007/978-3-642-33326-2-7.

[347] B. Paudel, F. Christoffel, C. Newell, and A. Bernstein. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Transactions on Interactive Intelligent Systems*, 7(1), 2017. DOI: 10.1145/2955101.

[348] M. J. Pazzani and D. Billsus. *Content-Based Recommendation Systems*. In *The Adaptive Web*. Springer, 2007, pages 325–341. DOI: 10.1007/978-3-540-72079-9-10.

[349] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. DOI: 10.1080/14786440109462720.

[350] E. J. Pedhazur and L. P. Schmelkin. *Measurement, Design, and Analysis*. Psychology Press, 2013. DOI: 10.4324/9780203726389.

[351] D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware Data Mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 560–568. ACM, 2008. DOI: 10.1145/1401890.1401959.

[352] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77, 2007. DOI: 10.2753/mis0742-1222240302.

[353] G. C. A. Peng and F. Annansingh. *Experiences in Applying Mixed-Methods Approach in Information Systems Research.* In *Information Systems Research and Exploring Social Artifacts.* IGI Global, 2013. Chapter 14, pages 266–293. DOI: 10.4018/978-1-4666-2491-7.ch014.

[354] C. S. Pereira, J. Teixeira, P. Figueiredo, J. Xavier, S. L. Castro, and E. Brattico. Music and Emotions in the Brain: Familiarity Matters. *PLoS ONE*, 6(11), 2011. DOI: 10.1371/journal.pone.0027241.

[355] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710. ACM, 2014. DOI: 10.1145/2623330.2623732.

[356] M. Pichl and E. Zangerle. Latent Feature Combination for Multi-Context Music Recommendation. In *International Conference on Content-Based Multimedia Indexing*, CBMI '18, pages 1–6. IEEE, 2018. DOI: 10.1109/cbmi.2018.8516495.

[357] M. Pichl and E. Zangerle. User models for multi-context-aware music recommendation. *Multimedia Tools and Applications*, 80(15):22509–22531, 2021. DOI: 10.1007/s11042-020-09890-7.

[358] M. Pichl, E. Zangerle, and G. Specht. #Nowplaying on #spotify: leveraging spotify information on twitter for artist recommendations. In *Current Trends in Web Engineering, 15th International Conference, ICWE 2015 Workshops (Revised Selected Papers)*, pages 163–174. Springer, 2015. DOI: 10.1007/978-3-319-24800-4-14.

[359] M. Pichl, E. Zangerle, and G. Specht. Improving Context-Aware Music Recommender Systems: Beyond the Pre-filtering Approach. In *Proceedings of the 2017 ACM International Conference on Multimedia Retrieval*, ICMR '17, pages 201–208. ACM, 2017. DOI: 10.1145/3078971.3078980.

[360] 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*, 8(4):44–59, 2017. DOI: 10.4018/ijmdem.2017100103.

[361] M. Pichl, E. Zangerle, and G. Specht. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name? In *IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1360–1365. IEEE, 2015. DOI: 10.1109/icdmw.2015.145.

[362] M. Pichl, E. Zangerle, and G. Specht. Understanding Playlist Creation on Music Streaming Platforms. In *IEEE International Symposium on Multimedia*, ISM '16, pages 475–480. IEEE, 2016. DOI: 10.1109/ism.2016.0107.

[363] M. Pichl, E. Zangerle, G. Specht, and M. Schedl. Mining Culture-Specific Music Listening Behavior from Social Media Data. In *IEEE International Symposium on Multimedia*, ISM '17, pages 208–215. IEEE, 2017. DOI: 10.1109/ism.2017.35.

[364] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 157–164. ACM, 2011. DOI: 10.1145/2043932.2043962.

[365] P. Pu, L. Chen, and R. Hu. Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22(4–5):317–355, 2012. DOI: 10.1007/s11257-011-9115-7.

[366] M. Punkanen, T. Eerola, and J. Erkkilä. Biased emotional recognition in depression: Perception of emotions in music by depressed patients. *Journal of Affective Disorders*, 130(1-2):118–126, 2011. DOI: 10.1016/j.jad.2010.10.034.

[367] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-Aware Recommender Systems. *ACM Computing Surveys*, 51(4), 2018. DOI: 10.1145/3190616.

[368] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*, RecSys '17, pages 130–137. ACM, 2017. DOI: 10.1145/3109859.3109896.

[369] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2007. DOI: 10.1007/978-0-387-22750-4.

[370] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to Know You: Learning New User Preferences in Recommender Systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 127–134. ACM, 2002. DOI: 10.1145/502716.502737.

[371] S. Rendle. Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–22, 2012. DOI: 10.1145/2168752.2168771.

[372] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461. AUAI Press, 2009.

[373] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*, SIGIR '11, pages 635–644. ACM, 2011. DOI: 10.1145/2009916.2010002.

[374] 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 '10, pages 81–90. ACM, 2010. DOI: 10.1145/1718487.1718498.

[375] S. Rendle, L. Zhang, and Y. Koren. On the difficulty of evaluating baselines: A study on recommender systems, 2019. DOI: 10.48550/ARXIV.1905.01395.

[376] P. J. Rentfrow and S. D. Gosling. The content and validity of music-genre stereotypes among college students. *Psychology of Music*, 35(2):306–326, 2007. DOI: 10.1177/0305735607070382.

[377] P. J. Rentfrow and S. D. Gosling. The Do Re Mi's of Everyday Life: The Structure and Personality Correlates of Music Preferences. *Journal of Personality and Social Psychology*, 84(6):1236–1256, 2003. DOI: 10.1037/0022-3514.84.6.1236.

[378] P. Resnick and H. R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997. DOI: 10.1145/245108.245121.

[379] M. Reusens, W. Lemahieu, B. Baesens, and L. Sels. Evaluating recommendation and search in the labor market. *Knowledge-based Systems*, 152:62–69, 2018. DOI: 10.1016/j.knosys.2018.04.007.

[380] D. Reynolds. Gaussian Mixture Models. *Encyclopedia of Biometrics*, 741:827–832, 2015.

[381] 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 '09, pages 713–716. ACM, 2009. DOI: 10.1145/1631272.1631395.

[382] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. A. Gonçalves, and F. Benevenuto. Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29, 2016.

[383] F. Ricci and Q. N. Nguyen. Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. *IEEE Intelligent Systems*, 22(3):22–29, 2007. DOI: 10.1109/mis.2007.43.

[384] F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems: Techniques, Applications, and Challenges*. In *Recommender Systems Handbook*. Springer, 3rd edition, 2022, pages 1–35. DOI: 10.1007/978-1-0716-2197-4-1.

[385] C. Richthammer and G. Pernul. Situation awareness for recommender systems. *Electronic Commerce Research*, 20(4):783–806, 2018. DOI: 10.1007/s10660-018-9321-z.

[386] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and Passivity in Social Media. In *Proceedings of the 20th International Conference cCompanion on World Wide Web*, pages 113–114. ACM, 2011. DOI: 10.1145/1963192.1963250.

[387] R. L. Rosa, D. Z. Rodriguez, and G. Bressan. Music recommendation system based on user's sentiments extracted from social networks. *IEEE Transactions on Consumer Electronics*, 61(3):359–367, 2015. DOI: 10.1109/tce.2015.7298296.

[388] M. Rossetti, F. Stella, and M. Zanker. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 31–34. ACM, 2016. DOI: 10.1145/2959100.2959176.

[389] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000. DOI: 10.1126/science.290.5500.2323.

[390] D. Russo, P. Ciancarini, T. Falasconi, and M. Tomasi. A Meta-Model for Information Systems Quality: A Mixed Study of the Financial Sector. *ACM Transactions on Management Information Systems*, 9(3), 2018. DOI: 10.1145/3230713.

[391] S. Saarikallio. Music as emotional self-regulation throughout adulthood. *Psychology of Music*, 39(3):307–327, 2010. DOI: 10.1177/0305735610374894.

[392] S. Saarikallio, S. Nieminen, and E. Brattico. Affective reactions to musical stimuli reflect emotional use of music in everyday life. *Music and Science*, 17(1):27–39, 2012. DOI: 10.1177/1029864912462381.

[393] S. H. Saarikallio. Music in Mood Regulation: Initial Scale Development. *Music and Science*, 12(2):291–309, 2008. DOI: 10.1177/102986490801200206.

[394] M. E. Sachs, A. Damasio, and A. Habibi. The pleasures of sad music: A systematic review. *Frontiers in Human Neuroscience*, 9(404):1–12, 2015. DOI: 10.3389/fnhum.2015.00404.

[395] A. Said and A. Bellogıén. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 129–136. ACM, 2014. DOI: 10.1145/2645710.2645746.

[396] A. Said and A. Bellogıén. RiVal — A Toolkit to Foster Reproducibility in Recommender System Evaluation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 371–372. ACM, 2014. DOI: 10.1145/2645710.2645712.

[397] A. Said, E. W. De Luca, and S. Albayrak. Inferring Contextual User Profiles—Improving Recommender Performance. In *Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems*, volume 791 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[398] A. Said, B. Fields, B. J. Jain, and S. Albayrak. User-Centric Evaluation of a K-Furthest Neighbor Collaborative Filtering Recommender Algorithm. In *Proceedings of the 2013 Conference on Computer supported cooperative work*, CSCW '13, pages 1399–1408. ACM, 2013. DOI: 10.1145/2441776.2441933.

[399] A. Said, D. Tikk, K. Stumpf, Y. Shi, M. Larson, and P. Cremonesi. Recommender Systems Evaluation: A 3D Benchmark. In *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE*, volume 910 of *CEUR Workshop Proceedings*, pages 21–23. CEUR-WS.org, 2012.

[400] S. L. Salzberg. On Comparing Classifiers: Pitfalls To Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997. DOI: 10.1023/a:1009752403260.

[401] P. Sapiezynski, W. Zeng, R. Robertson, A. Mislove, and C. Wilson. Quantifying the Impact of User Attentionon Fair Group Representation in Ranked Lists. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW 2019 Companion, pages 553–562. ACM, 2019. DOI: 10.1145/3308560.3317595.

[402] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295. ACM, 2001. DOI: 10.1145/371920.372071.

[403] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. *Collaborative Filtering Recommender Systems*. In *The Adaptive Web*. Springer, 2007, pages 291–324. DOI: 10.1007/978-3-540-72079-9-9.

[404] T. Schäfer, P. Sedlmeier, C. Städtler, and D. Huron. The psychological functions of music listening. *Frontiers in Psychology*, 4(511):1–34, 2013. DOI: 10.3389/fpsyg.2013.00511.

[405] M. Schedl. Deep Learning in Music Recommendation Systems. *Frontiers in Applied Mathematics and Statistics*, 5:44, 2019. DOI: 10.3389/fams.2019.00044.

[406] 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. DOI: 10.1007/s13735-017-0118-y.

[407] M. Schedl. Leveraging Microblogs for Spatiotemporal Music Information Retrieval. In *European Conference on Information Retrieval*, ECIR '13, pages 796–799. Springer, 2013. DOI: 10.1007/978-3-642-36973-5-87.

[408] M. Schedl. The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, pages 103–110. ACM, 2016. DOI: 10.1145/2911996.2912004.

[409] M. Schedl and C. Bauer. Distance-and Rank-based Music Mainstreaminess Measurement. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 364–367. ACM, 2017.

[410] M. Schedl, S. Brandl, O. Lesota, E. Parada-Cabaleiro, D. Penz, and N. Rekabsaz. LFM-2b: A Dataset of Enriched Music Listening Events for Recommender Systems Research and Fairness Analysis. In *Proceedings of the 7th ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '22, pages 337–341. ACM, 2022. DOI: 10.1145/3498366.3505791.

[411] M. Schedl, G. Breitschopf, and B. Ionescu. Mobile Music Genius. Reggae at the beach, metal on a friday night? In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 507–510. ACM, 2014. DOI: 10.1145/2578726.2582612.

[412] M. Schedl and A. Flexer. Putting the User in the Center of Music Information Retrieval. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ISMIR '12, pages 385–390. ISMIR, 2012.

[413] 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. DOI: 10.1007/s10844-013-0247-6.

[414] 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, 2014. DOI: 10.1561/1500000042.

[415] M. Schedl, D. Hauger, K. Farrahi, and M. Tkalčič. On the Influence of User Characteristics on Music Recommendation Algorithms. In *Advances in Information Retrieval - 37th European Conference on IR Research, ECIR 2015, Proceedings*, volume 9022 of *Lecture Notes in Computer Science*, pages 339–345. Springer, 2015. DOI: 10.1007/978-3-319-16354-3-37.

[416] M. Schedl and P. Knees. Personalization in Multimodal Music Retrieval. In *Adaptive Multimedia Retrieval. Large-Scale Multimedia Retrieval and Evaluation*, pages 58–71. Springer, 2013.

[417] M. Schedl, P. Knees, and F. Gouyon. New Paths in Music Recommender Systems Research. In *Proceedings of the 11th ACM Conference on Recommender Systems*, RecSys '17, pages 392–393. ACM, 2017. DOI: 10.1145/3109859.3109934.

[418] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas. *Music Recommender Systems*. In *Recommender Systems Handbook*. Springer, 2nd edition, 2015, pages 453–492. DOI: 10.1007/978-1-4899-7637-6-13.

[419] M. Schedl, F. Lemmerich, B. Ferwerda, M. Skowron, and P. Knees. Indicators of Country Similarity in Terms of Music Taste, Cultural, and Socio-economic Factors. In *IEEE International Symposium on Multimedia*, ISM '17, pages 308–311. IEEE, 2017. DOI: 10.1109/ism.2017.55.

[420] M. Schedl and D. Schnitzer. Hybrid Retrieval Approaches to Geospatial Music Recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 793–796. ACM, 2013. DOI: 10.1145/2484028.2484146.

[421] M. Schedl and D. Schnitzer. Location-Aware Music Artist Recommendation. In *Proceedings of the 20th International Conference on MultiMedia Modeling*, MMM '14, pages 205–213. Springer, 2014.

[422] M. Schedl, A. Vall, and K. Farrahi. User Geospatial Context for Music Recommendation in Microblogs. In *Proceedings of the 37th International ACM SIGIR Conference on Research & development in information retrieval*, SIGIR '14, pages 987–990. ACM, 2014. DOI: 10.1145/2600428.2609491.

[423] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7(2):95–116, 2018. DOI: 10.1007/s13735-018-0154-2.

[424] 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):582–593, 2002. DOI: 10.1037/0022-3514.82.4.582.

[425] G. Schröder, M. Thiele, and W. Lehner. Setting goals and choosing metrics for recommender system evaluations. In *Joint proceedings of the '11 Workshop on Human Decision Making in Recommender Systems (DecisionsRecSys'11) and User-Centric Evaluation of Recommender Systems and Their Interfaces-2*, volume 811 of *CEUR Workshop Proceedings*, pages 78–85. CEUR-WS.org, 2011.

[426] E. Schubert. The influence of emotion, locus of emotion and familiarity upon preference in music. *Psychology of Music*, 35(3):499–515, 2007. DOI: 10.1177/0305735607072657.

[427] A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, and J. Vertesi. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT*'19, pages 59–68. ACM, 2019. DOI: 10.1145/3287560.3287598.

[428] A. Severyn and A. Moschitti. Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 959–962. ACM, 2015. DOI: 10.1145/2766462.2767830.

[429] B. Shao, D. Wang, T. Li, and M. Ogihara. Music Recommendation Based on Acoustic Features and User Access Patterns. *IEEE/ACM Transactions on Speech and Audio Processing*, 17(8):1602–1611, 2009. DOI: 10.1109/tasl.2009.2020893.

[430] U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217. ACM, 1995. DOI: 10.1145/223904.223931.

[431] S. J. Sheather. Density Estimation. *Statistical Science*, 19(4):588–597, 2004. DOI: 10.1214/088342304000000297.

[432] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. DOI: 10.1109/34.868688.

[433] Y. Shi, M. Larson, and A. Hanjalic. Collaborative Filtering beyond the User-Item Matrix. A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):1–45, 2014. DOI: 10.1145/2556270.

[434] Y. Shi, M. Larson, and A. Hanjalic. Mining Contextual Movie Similarity with Matrix Factorization for Context-Aware Recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 34–40. ACM, 2010. DOI: 10.1145/1869652.1869658.

[435] A. Singh and T. Joachims. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 2219–2228. ACM, 2018. DOI: 10.1145/3219819.3220088.

[436] J. Smedslund. Why Psychology Cannot be an Empirical Science. *Integrative Psychological and Behavioral Science*, 50(2):185–195, 2015. DOI: 10.1007/s12124-015-9339-x.

[437] B. Smyth and P. McClave. Similarity vs. Diversity. In *International Conference on Case-Based Reasoning*, ICCBR '01, pages 347–361. Springer, 2001. DOI: 10.1007/3-540-44593-5-25.

[438] N. Sonboli, R. Burke, M. Ekstrand, and R. Mehrotra. The multisided complexity of fairness in recommender systems. *AI Magazine*, 43(2):164–176, 2022. DOI: 10.1002/aaai.12054.

[439] E. R. Spangenberg, I. Kareklas, B. Devezer, and D. E. Sprott. A Meta-Analytic Synthesis of the Question-Behavior Effect. *Journal of Consumer Psychology*, 26(3):441–458, 2016. DOI: 10.1016/j.jcps.2015.12.004.

[440] K. Sparck Jones. A Statistical Interpretation Of Term Specificity And Its Application In Retrieval. *Journal of Documentation*, 28(1):11–21, 1972. DOI: 10.1108/eb026526.

[441] S. Srinivasan, S. Bhattacharya, and R. Chakraborty. Segmenting web-domains and hashtags using length specific models. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1113–1122. ACM, 2012. DOI: 10.1145/2396761.2398410.

[442] B. St. Thomas, P. Chandar, C. Hosey, and F. Diaz. Mixed Method Development of Evaluation Metrics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 4070–4071. ACM, 2021. DOI: 10.1145/3447548.3470802.

[443] H. Steck. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference on*, WWW '19, pages 3251–3257. ACM, 2019. DOI: 10.1145/3308558.3313710.

[444] H. Steck. Evaluation of Recommendations: Rating-prediction and Ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 213–220. ACM, 2013. DOI: 10.1145/2507157.2507160.

[445] H. Steck. Item Popularity and Recommendation Accuracy. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 125–132. ACM, 2011. DOI: 10.1145/2043932.2043957.

[446] E. Stern. *Evaluation Research Methods*. SAGE Publications, 2005. DOI: 10.4135/9781446261606.

[447] X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009, 2009. DOI: 10.1155/2009/421425.

[448] Z. Sun, D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, and C. Geng. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 23–32. ACM, 2020. DOI: 10.1145/3383313.3412489.

[449] A. Swaminathan and T. Joachims. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research*, 16(52):1731–1755, 2015.

[450] A. Swartz. MusicBrainz: A Semantic Web Service. *IEEE Intelligent Systems*, 17(1):76–77, 2002. DOI: 10.1109/5254.988466.

[451] J. A. Swets. Information Retrieval Systems: Statistical Decision Theory May Provide a Measure of Effectiveness Better than Measures Proposed to Date. *Science*, 141(3577):245–250, 1963. DOI: 10.1126/science.141.3577.245.

[452] M. Tamir. Don't worry, be happy? Neuroticism, trait-consistent affect regulation, and performance. *Journal of Personality and Social Psychology*, 89(3):449–461, 2005. DOI: 10.1037/0022-3514.89.3.449.

[453] M. Tamir. The Maturing Field of Emotion Regulation. *Emotion Review*, 3(1):3–7, 2011. DOI: 10.1177/1754073910388685.

[454] Y.-M. Tamm, R. Damdinov, and A. Vasilev. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently? In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys 2021, pages 708–713. ACM, 2021. DOI: 10.1145/3460231.3478848.

[455] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1067–1077. ACM, 2015. DOI: 10.1145/2736277.2741093.

[456] J. Tang, C. Aggarwal, and H. Liu. Recommendations in Signed Social Networks. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 31–40. ACM, 2016. DOI: 10.1145/2872427.2882971.

[457] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000. DOI: 10.1126/science.290.5500.2319.

[458] J. Teorell, M. Samanni, S. Holmberg, and B. Rothstein. The Quality of Government Basic Dataset made from The QoG Standard Dataset. *University of Gothenburg: The Quality of Government Institute*, 2012.

[459] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment Strength Detection in Short Informal Text. *Journal of the Association for Information Science and Technology*, 61(12):2544–2558, 2010. DOI: 10.1002/asi.21416.

[460] N. Tintarev, M. Dennis, and J. Masthoff. Adapting Recommendation Diversity to Openness to Experience: A Study of Human Behaviour. In *User Modeling, Adaptation, and Personalization*, UMAP '13, pages 190–202. Springer, 2013.

[461] N. Tintarev and J. Masthoff. *Designing and Evaluating Explanations for Recommender Systems*. In *Recommender Systems Handbook*. Springer, 2nd edition, 2010, pages 479–510. DOI: 10.1007/978-0-387-85820-3-15.

[462] M. E. Tipping and C. M. Bishop. Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation*, 11(2):443–482, 1999. DOI: 10.1162/089976699300016728.

[463] M. Tkalčič and L. Chen. *Personality and Recommender Systems*. In *Recommender Systems Handbook*. Springer, 2015, pages 715–739. DOI: 10.1007/978-1-4899-7637-6-21.

[464] M. Tkalčič, A. Košir, and J. Tasič. Affective recommender systems: The role of emotions in recommender systems. In *Proceedings of the RecSys Workshop on Human Decision Making in Recommender Systems*, pages 9–13, 2011.

[465] N. K. Tran. Classification and Learning-to-rank Approaches for Cross-Device Matching at CIKM Cup 2016, 2016. DOI: 10.48550/ARXIV.1612.07117.

[466] O. Tsur and A. Rappoport. What's in a Hashtag? Content based Prediction of the Spread of Ideas in Microblogging Communities. In *Proceedings of the 5th ACM International Conference on Web search and data mining*, WSDM '12, pages 643–652. ACM, 2012. DOI: 10.1145/2124295.2124320.

[467] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, 2008. DOI: 10.1109/tasl.2007.913750.

[468] Twitter: Twitter Filter API https://dev.twitter.com/docs/api/1.1
/post/statuses/filter.

[469] A. Vall, M. Dorfer, H. Eghbal-zadeh, M. Schedl, K. Burjorjee, and G. Widmer. Feature-combination hybrid recommender systems for automated music playlist continuation. *User Modeling and User-Adapted Interaction*, 29(2):527–572, 2019. DOI: 10.1007/s11257-018-9215-8.

[470] A. Van Den Oord, S. Dieleman, and B. Schrauwen. Deep Content-Based Music Recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS '13, pages 2643–2651. Curran Associates, 2013. DOI: 10.5555/2999792.2999907.

[471] L. van der Maaten and G. Hinton. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55, 2011. DOI: 10.1007/s10994-011-5273-4.

[472] A. van Goethem and J. Sloboda. The functions of music for affect regulation. *Music and Science*, 15(2):208–228, 2011. DOI: 10.1177/1029864911401174.

[473] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[474] S. Vargas and P. Castells. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 109–116. ACM, 2011. DOI: 10.1145/2043932.2043955.

[475] G. Vigliensoni and I. Fujinaga. The Music Listening Histories Dataset. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ISMIR '17, pages 96–102. ISMIR, 2017.

[476] E. M. Voorhees. The Philosophy of Information Retrieval Evaluation. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 355–370. Springer, 2002. DOI: 10.1007/3-540-45691-0-34.

[477] E. M. Voorhees. The TREC-8 question answering track report, 2000.

[478] E.-J. Wagenmakers, R. Wetzels, D. Borsboom, H. L. J. van der Maas, and R. A. Kievit. An Agenda for Purely Confirmatory Research. *Perspectives on Psychological Science*, 7(6):632–638, 2012. DOI: 10.1177/1745691612463078.

[479] J.-C. Wang, Y.-H. Yang, H.-M. Wang, and S.-K. Jeng. The Acoustic Emotion Gaussians Model for Emotion-based Music Annotation and Retrieval. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 89–98. ACM, 2012. DOI: 10.1145/2393347.2393367.

[480] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang. Community-Based Weighted Graph Model for Valence-Arousal Prediction of Affective Words. *IEEE/ACM Transactions on Speech and Audio Processing*, 24(11):1957–1968, 2016. DOI: 10.1109/taslp.2016.2594287.

[481] X. Wang, D. Rosenblum, and Y. Wang. Context-Aware Mobile Music Recommendation for Daily Activities. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 99–108. ACM, 2012. DOI: 10.1145/2393347.2393368.

[482] 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 '14, pages 627–636. ACM, 2014. DOI: 10.1145/2647868.2654940.

[483] Y. Wang, W. Ma, M. Zhang, Y. Liu, and S. Ma. A Survey on the Fairness of Recommender Systems. *ACM Transactions on Information Systems*, 2022. DOI: 10.1145/3547333.

[484] D. Watson and R. Mandryk. An In-Situ Study of Real-Life Listening Context. In *Proceedings of Sound and Music Computing Conference*, SMC '12, pages 11–16. SMC, 2012.

[485] C. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005. DOI: 10.3354/cr030079.

[486] G. Wu, V. Swaminathan, S. Mitra, and R. Kumar. Digital content recommendation system using implicit feedback data. In *IEEE International Conference on Big Data*, Big Data '17, pages 2766–2771. IEEE, 2017. DOI: 10.1109/bigdata.2017.8258242.

[487] Q. Wu, H. Wang, L. Hong, and Y. Shi. Returning is Believing: Optimizing Long-term User Engagement in Recommender Systems. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1927–1936. ACM, 2017. DOI: 10.1145/3132847.3133025.

[488] B. Xiao and I. Benbasat. E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. *Management Information Systems Quarterly*, 31(1):137, 2007. DOI: 10.2307/25148784.

[489] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander. A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases. In *Proceedings of the 14th International Conference on Data Engineering*, volume 96 of number 34 in *ICDE '96*, pages 226–231. IEEE, 1996. DOI: 10.1109/icde.1998.655795.

[490] H.-C. Yang and Z.-R. Huang. Mining personality traits from social messages for game recommender systems. *Knowledge-based Systems*, 165:157–168, 2019. DOI: 10.1016/j.knosys.2018.11.025.

[491] Y.-H. Yang and H. H. Chen. Machine Recognition of Music Emotion. A review. *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–30, 2012. DOI: 10.1145/2168752.2168754.

[492] Y.-H. Yang and H. H. Chen. *Music Emotion Recognition*. CRC Press, 2011. DOI: 10.1201/b10731.

[493] Y.-H. Yang and J.-Y. Liu. Quantitative Study of Music Listening Behavior in a Social and Affective Context. *IEEE Transactions on Multimedia*, 15(6):1304–1315, 2013. DOI: 10.1109/tmm.2013.2265078.

[494] Y.-H. Yang and Y.-C. Teng. Quantitative Study of Music Listening Behavior in a Smartphone Context. *ACM Transactions on Interactive Intelligent Systems*, 5(3):1–30, 2015. DOI: 10.1145/2738220.

[495] L. Yang, T. Sun, M. Zhang, and Q. Mei. We Know What You #Tag: Does the Dual Role Affect Hashtag Adoption? In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 261–270. ACM, 2012. DOI: 10.1145/2187836.2187872.

[496] S. Yao and B. Huang. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS '17, pages 2925–2934. Curran Associates, 2017. DOI: 10.5555/3294996.3295052.

[497] R. K. Yin. *Case Study Research: Design and Methods*. SAGE Publications, 5th revised edition, 1989.

[498] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 974–983. ACM, 2018. DOI: 10.1145/3219819.3219890.

[499] S. Yoo and K. Lee. A Data-driven Approach to Identifying Music Listener Groups based on Users' Playrate Distributions of Listening Events. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 77–81. ACM, 2017. DOI: 10.1145/3099023.3099075.

[500] W. York. Voices from hell–the dark, not-so-dulcet Cookie Monster vocals of extreme metal. *The San Francisco Bay Guardian*:14–20, 2004.

[501] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences. In *Proceedings of the 7th International Society for Music Information Retrieval Conference*, ISMIR '06. ISMIR, 2006.

[502] H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon. Scalable Coordinate Descent Approaches to Parallel Matrix Factorization for Recommender Systems. In *IEEE 12th International Conference on Data Mining*, ICDM '12, pages 765–774. IEEE, 2012. DOI: 10.1109/icdm.2012.168.

[503] E. Zangerle. Culture-Aware Music Recommendation Dataset. DOI: 10.5281/zenodo.3477842.

[504] E. Zangerle and C. Bauer. Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 2022. ISSN: 0360-0300. DOI: 10.1145/3556536.

[505] E. Zangerle, C.-M. Chen, M.-F. Tsai, and Y.-H. Yang. Leveraging Affective Hashtags for Ranking Music Recommendations. *IEEE Transactions on Affective Computing*, 12(1):78–91, 2021. DOI: 10.1109/taffc.2018.2846596.

[506] 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 '16. ACM, 2016. DOI: 10.1145/2957792.2957804.

[507] 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): Big things come in small packages*, volume 838 of *CEUR Workshop Proceedings*, pages 14–17. CEUR-WS.org, 2012.

[508] E. Zangerle and M. Pichl. Content-based User Models: Modeling the Many Faces of Musical Preference. In *Proceedings of the 19th International Society for Music Information Retrieval Conference 2018 (ISMIR 2018)*, ISMIR '18, pages 709–716. ISMIR, 2018. DOI: 10.5281/zenodo.1492515.

[509] E. Zangerle, M. Pichl, W. Gassler, and G. Specht. #Nowplaying music dataset. Extracting listening behavior from twitter. In *Proceedings of the 1st International Workshop on Internet-Scale Multimedia Management*, pages 21–26. ACM, 2014. DOI: 10.1145/2661714.2661719.

[510] 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 '14, pages 21–26. ACM, 2014. DOI: 10.1145/2661714.2661719.

[511] E. Zangerle, M. Pichl, and M. Schedl. Culture-Aware Music Recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, pages 357–358. ACM, 2018. DOI: 10.1145/3209219.3209258.

[512] E. Zangerle, M. Pichl, and M. Schedl. User Models for Culture-Aware Music Recommendation: Fusing Acoustic and Cultural Cues. *Transactions of the International Society for Music Information Retrieval*, 3(1):1–16, 2020. DOI: 10.5334/tismir.37.

[513] M. Zanker, L. Rook, and D. Jannach. Measuring the Impact of Online Personalisation: Past, Present and Future. *International Journal of Human-Computer Studies*, 131:160–168, 2019. DOI: 10.1016/j.ijhcs.2019.06.006.

[514] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009. DOI: 10.1109/tpami.2008.52.

[515] M. Zentner, D. Grandjean, and K. R. Scherer. Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion*, 8(4):494–521, 2008. DOI: 10.1037/1528-3542.8.4.494.

[516] B. Zhang, J. Shen, Q. Xiang, and Y. Wang. CompositeMap: A Novel Framework for Music Similarity Measure. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 403–410. ACM, 2009. DOI: 10.1145/1571941.1572011.

[517] S. Zhang, Y. Tay, L. Yao, A. Sun, and C. Zhang. *Deep Learning for Recommender Systems*. In *Recommender Systems Handbook*. Springer, 2022, pages 173–210. DOI: 10.1007/978-1-0716-2197-4-5.

[518] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, 52(1), 2019. DOI: 10.1145/3285029.

[519] Q. Zhao, G. Adomavicius, F. M. Harper, M. Willemsen, and J. A. Konstan. Toward Better Interactions in Recommender Systems: Cycling and Serpentining Approaches for Top-N Item Lists. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 1444–1453. ACM, 2017. DOI: 10.1145/2998181.2998211.

[520] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 167–176. ACM, 2018. DOI: 10.1145/3178876.3185994.

[521] L. Zheng, V. Noroozi, and P. S. Yu. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 425–434. ACM, 2017. DOI: 10.1145/3018661.3018665.

[522] Y. Zheng, B. Mobasher, and R. D. Burke. The Role of Emotions in Context-aware Recommendation. In *Proceedings of the RecSys Workshop on Human Decision Making in Recommender Systems*, pages 21–28, 2013.

[523] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakening, and Y.-C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010. DOI: 10.1073/pnas.1000488107.

[524] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32. ACM, 2005. DOI: 10.1145/1060745.1060754.