Multimedia Tools and Applications (2021) 80:22509–22531
https://doi.org/10.1007/s11042-020-09890-7

User models for multi-context-aware music recommendation

Martin Pichl[1] · **Eva Zangerle**[1]

Received: 1 March 2019 / Revised: 24 June 2020 / Accepted: 16 September 2020 /
Published online: 1 October 2020

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.

Keywords Recommender systems · Context-aware recommender systems · Personalization · User modeling

1 Introduction

Over the last decade, people have increasingly started to use music streaming platforms providing millions of tracks [32]. Streaming platforms heavily rely on recommender systems to help users navigate through the provided collections and discover music they

✉ Eva Zangerle
eva.zangerle@uibk.ac.at

[1] Department of Computer Science, University of Innsbruck, Innsbruck, Austria

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 [27, 31] as people listen to different music during different activities [23]. Also, Cunningham et al. [13] 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 [42, 44]. 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 [43]. 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) [46] 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 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 3, we formulate the problem underlying our work. Section 4 presents the dataset utilized and in Section 5, we present the proposed multi-context user model and recommendation approach. Subsequently, we describe the experimental setup underlying our evaluation in Section 6 and present and discuss the obtained results in Section 7. Finally, we wrap up our work in Section 8.

[1] Please note that this manuscript is an extended version of [41], which was presented at the 2018 International Conference on Content-Based Multimedia Indexing (CBMI2018).

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 [42, 49, 54]. 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., [28]). 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) [28]). Several extensions to MF have been shown to work well (e.g., for implicit feedback data [18, 47] or for context-aware recommendations [5, 29]). 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) [50]. As a solution, hybrid systems combining collaborative filtering and content-based approaches have been proposed. Vall et al. [50] 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 [35] 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 [51] 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 [20] or by re-ranking strategies [17].

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 [1]. In the field of music recommender systems, users often seek music that suits their current context (i.e., occasion, event, or emotional states) [27, 31]. Kaminskas and Ricci [25] 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., [4, 16, 53]), the user's location (e.g., [11, 26]), or recommending music matching documents on the web a user reads at the moment [9]. Adomavicius and Tuzhilin [1] 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 [44], we showed that FM-based contextual modeling is able to outperform pre-filtering approaches.

Factorization Machines can be seen as an enhancement of CF [46]. 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 [46], 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 [7, 37] and shown to be useful for link prediction [7] or recommendations based on implicit feedback [52]. Inspired by the work of Rendle and Schmidt-Thieme [48], 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 [22]. More relevant for this work, FFMs have also been applied for music recommendation [10], 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.

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 (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.

$$f_R = User \times Track \rightarrow Rating \tag{1}$$

f_R can be learned by matrix factorization techniques such as SVD [30] as depicted in (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.

$$R = U \Sigma V^T \tag{2}$$

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

Prior research has shown that people listen to different music during different activities [23] and people create playlists that are intended for certain activities [13]. Hence, depending on different user contexts, different tracks need to be recommended. This

problem can be formulated as depicted in (3), where f_{CR} is a utility function assigning predicted ratings $\hat{r}_{ij}$ to user u for track i given user contexts c [1].

$$f_{CR} = User \times Track \times Contexts \rightarrow Rating \tag{3}$$

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.

4 Dataset

For our approach and the experiments conducted (cf. Sections 5 and 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 [43]. 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). 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., [2, 19, 36, 38, 43, 44, 55]). 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).

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 [43, 45]. Furthermore, we provide an interactive playlist explorer tool[3] that allows exploring the dataset and its acoustic characteristics in detail.

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

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

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 Fig. 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 4. Based on this dataset (shown in Fig. 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 Fig. 1), the input comprises the track id and the acoustic features for each track as provided by the Spotify API (cf. Section 4). This component computes the assignment of each track to an acoustic cluster. We describe this procedure in detail in Section 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 Fig. 1) computes the assignment of each track to a situational cluster. We detail this procedure in Section 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 3, we propose to utilize Factorization Machines (FM) [46] in a recommendation component ("Recommendation Component" in Fig. 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 5.3.

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 Fig. 1. In a first step, we aggregate the eight acoustic features obtained via the Spotify API (cf. Section 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) [33] 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

[4]In contrast to e.g., [14], we consider a playlist as an unordered set of tracks.

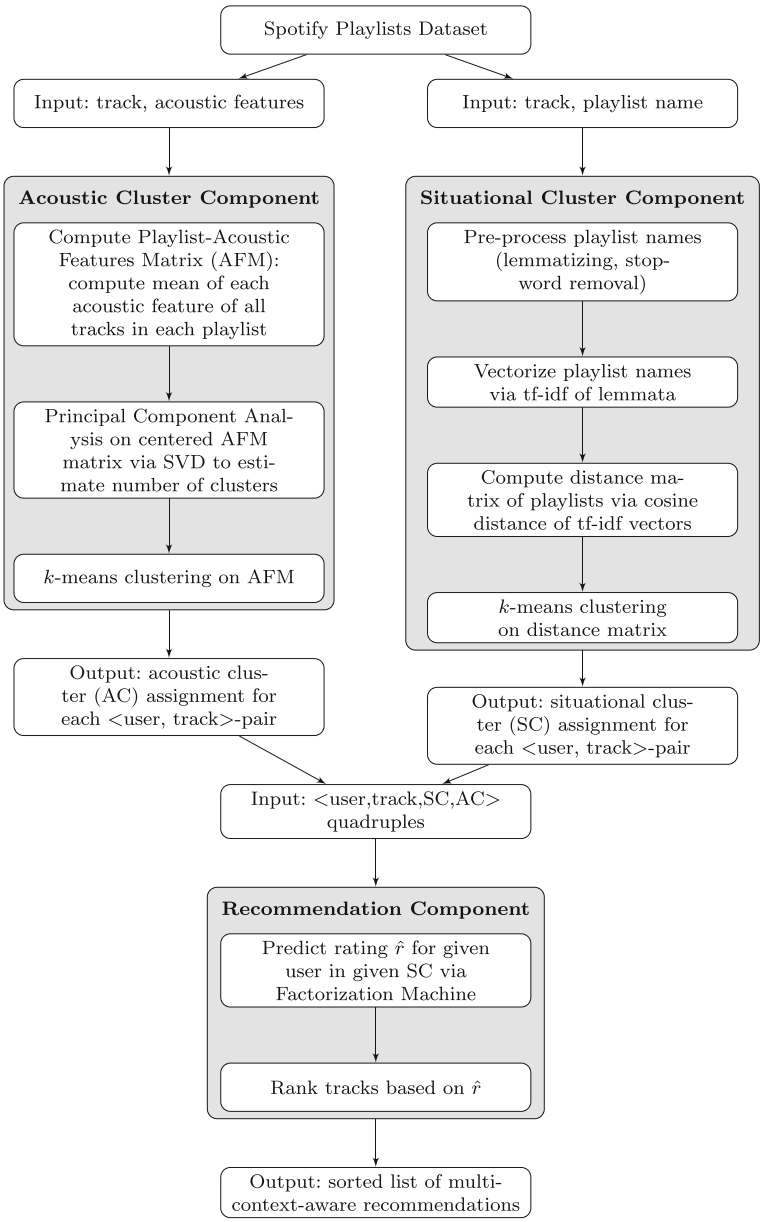

Fig. 1 Proposed framework for computing multi-context-aware recommendations

and refer to [43] 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) [40] via SVD [21].

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 Fig. 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 Fig. 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 [43] for further details on this approach and analyses of the resulting clusters.

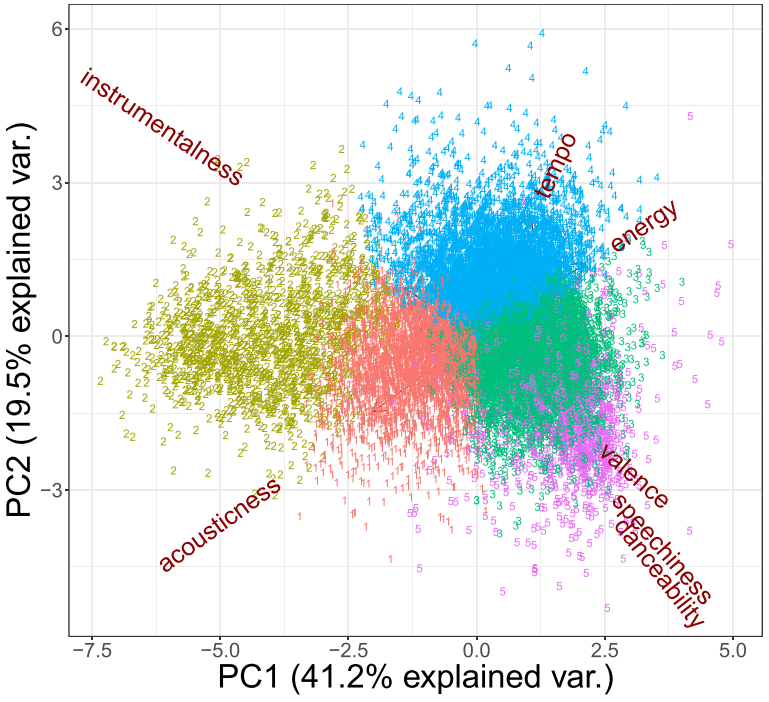

Fig. 2 Latent representation of playlist clusters

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 [42, 44], we mine for activities and other descriptors (seasons, events, etc.) in the names of playlists.

As depicted in Fig. 1, we firstly lemmatize all terms contained in playlist names using WordNet [39]. 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 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 [42] (cf. Section 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 [42, 44] 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.

5.3 Recommendation computation

The context extraction steps described in Sections 5.1 and 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 [46] 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

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

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 Fig. 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 [18]. 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 (4).

$$r_{ijsc} = \begin{cases} 1 & if\ u_i\ listened\ to\ t_j\ in\ SC_s\ and\ AC_c \\ 0 & otherwise \end{cases} \tag{4}$$

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: 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 ($\sum_{j=i+1}^{n} \langle \mathbf{v_i}, \mathbf{v_j} \rangle x_i x_j$). 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 $\langle \mathbf{v_i}, \mathbf{v_j} \rangle$ of low-dimensional vectors [46].

$$\hat{r}_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v_i}, \mathbf{v_j} \rangle x_i x_j \tag{5}$$

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 [15, 46].

Recently, higher-order Factorization Machines (HOFM) have been introduced, that allow for incorporating higher-order interaction effects [7, 37]. Aiming at further advancing the presented approach, we propose to also exploit 3-way interaction effects. A HOFM model is depicted in (6), where a further factor capturing 3-way interactions is added (in comparison

to 2-way Factorization Machines as depicted in (5)). Again, we rely on the Markov Chain Monte Carlo (MCMC) learning method.

$$\hat{r}_{HOFM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \langle \mathbf{v_i}, \mathbf{v_j} \rangle x_i x_j + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{l=j+1}^{m} \langle \mathbf{v_i}, \mathbf{v_j}, \mathbf{v_l} \rangle x_i x_j x_l \quad (6)$$

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.

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 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 1. This excerpt shows that user 872 has listened to track 250246 (belonging to acoustic feature-based cluster 4) in 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.

[6] https://github.com/andland/libFMexe
[7] https://github.com/cran/FactoRizationMachines

Table 1 Dataset fragment

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

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.

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 [34] (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 [44], 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 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.

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.

Table 2 Overview of evaluated models (top: baseline approaches, bottom: variations of the proposed multi-context approach).

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)	✓		✓	✓

7 Results and discussion

In the following we first discuss the results of the top-n recommendation task evaluation (Section 7.1), followed by the results of the rating prediction task (Section 7.2). Subsequently, we present the results of the evaluation of the impact of interaction effects (Section 7.3).

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 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 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.

Table 3 Top-R evaluation results (sorted by F_1, best results in bold)

Approach	Precision	Recall	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

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 [6]. 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 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 [3]. 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 [8]. Therefore, we evaluate the top-n performance of the proposed recommender system for a small number of n. Figure 3 depicts F_1 for $n = 1 \ldots 10$, 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

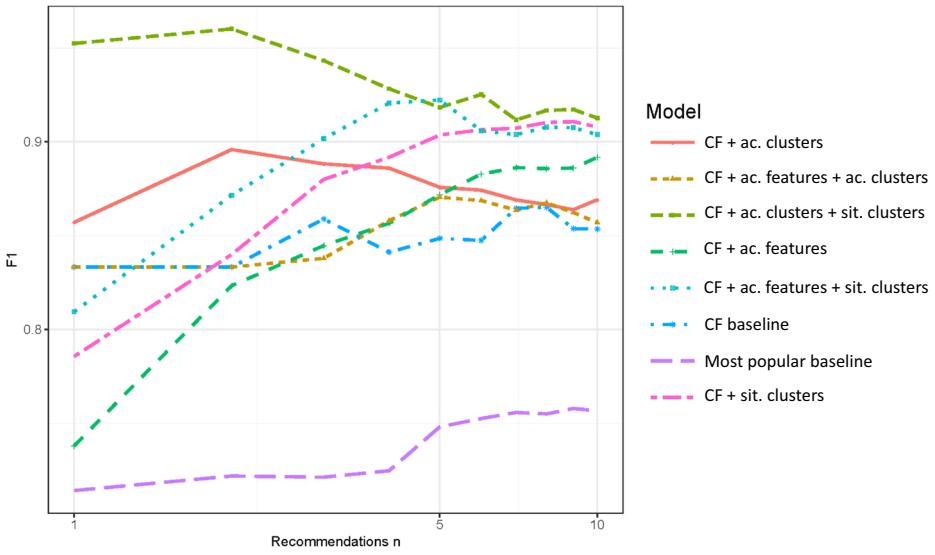

Fig. 3 F_1 for $n = 1 \ldots 10$ recommendations

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., [4, 5]). 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 7.3.

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.

Table 4 depicts the results of the rating prediction measures computed over the test set in Table 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 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

Table 4 Rating prediction evaluation results (sorted by RMSE, best results in bold).

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

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.

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. 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 in Fig. 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

Table 5 Impact of interaction effects: top-R evaluation, where the AC+SC model incorporates CF + acoustic clusters + situational clusters.

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

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.

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 Fig. 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 6) to conduct a fair comparison among the three approaches

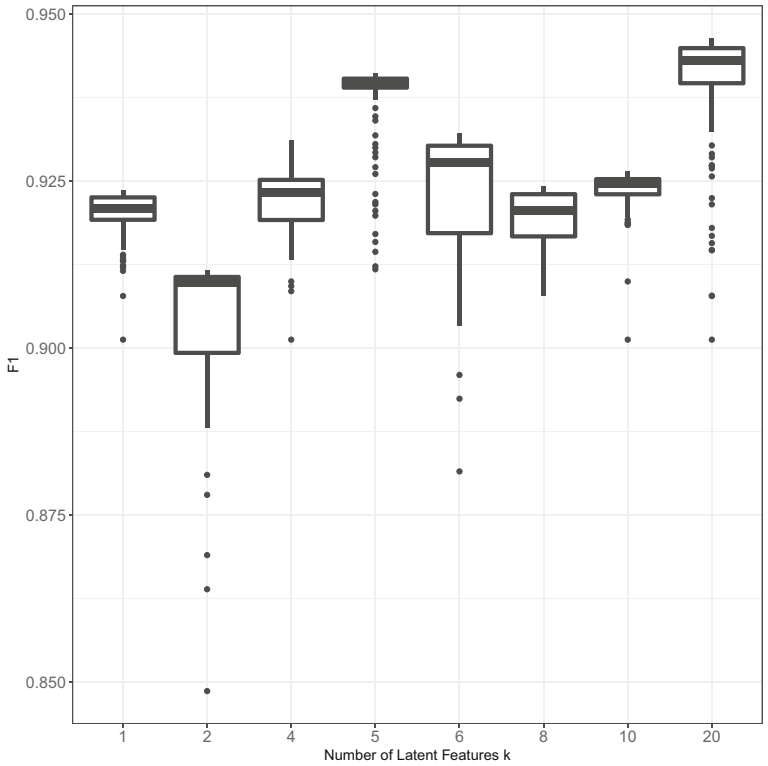

Fig. 4 $F_1@R$ for different numbers of latent features k

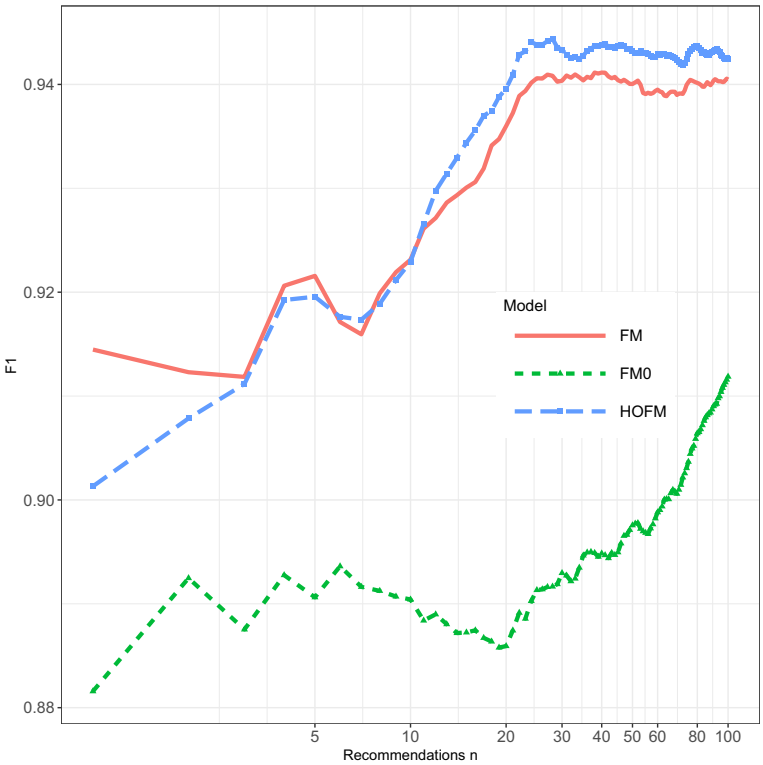

Fig. 5 F_1 for FM0 (no interactions), FM (2-way interactions) and HOFM (3-way interactions) for the AC+SC model for $n = 0...100$ (x-axis log-scaled)

(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 [52].

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 [53] or culture [56]. Also, the extraction of situational information from the names of playlists may also benefit from utilizing factorization models [12]. From an evaluation perspective, we also aim to investigate beyond-accuracy metrics [24] in future work to look into how contextual factors might affect aspects such as diversity of recommendation lists or novelty.

Funding Open access funding provided by University of Innsbruck and Medical University of Innsbruck.

References

1. Adomavicius G, Tuzhilin A (2010) Context-aware recommender systems. In: Ricci F., Rokach L., Shapira B., Kantor P. B. (eds) Recommender Systems Handbook, 1st edn., chap. 7. Springer, New York, pp 217–253
2. Andersen JS (2014) Using the Echo Nest's automatically extracted music features for a musicological purpose. In: 2014 4th International workshop on cognitive information processing (CIP), pp 1–6
3. Anderson C (2006) The Long Tail: Why the future of business is selling less of more hyperion
4. Baltrunas L, Kaminskas M, Ludwig B, Moling O, Ricci F, Aydin A, Lueke KH, Schwaiger R (2011) InCarMusic: Context-aware music recommendations in a car. In: E-Commerce and Web Technologies, LNBIP, vol 85. Springer, pp 89–100
5. Baltrunas L, Ludwig B, Ricci F (2011) Matrix factorization techniques for context aware recommendation. In: Proceedings Fifth ACM Conference on Recommender Systems (RecSys 2011), pp 301–304
6. Bellogin A, Castells P, Cantador I (2011) Precision-oriented evaluation of recommender systems: An algorithmic comparison. In: Proceedings of the fifth ACM conference on recommender systems. ACM, pp 333–336
7. Blondel M, Fujino A, Ueda N, Ishihata M (2016) Higher-order factorization machines. In: Advances in Neural Information Processing Systems, vol 29. Curran Associates Inc, pp 3351–3359
8. Bollen D, Knijnenburg BP, Willemsen MC, Graus M (2010) Understanding choice overload in recommender systems. In: Proceedings 4th ACM Conference on Recommender Systems (RecSys 2010), pp 63–70
9. Cai R, Zhang C, Chong W, Lei Z, Ma WY (2007) MusicSense: Contextual music recommendation using emotional allocation modeling. In: Proceedings 15th ACM International Conference on Multimedia (MM 2007)
10. Chen C, Tsai M, Liu J, Yang Y (2013) Music recommendation based on multiple contextual similarity information. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol 1, pp 65–72. https://doi.org/10.1109/WI-IAT.2013.10
11. Cheng Z, Shen J (2014) Just-for-Me: An Adaptive personalization system for location-aware social music recommendation. In: Proceedings 2014 ACM International Conference on Multimedia Retrieval (ICMR 2014)
12. Crain SP, Zhou K, Yang SH, Zha H (2012) Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. Springer, pp. 129–161 US. https://doi.org/10.1007/978-1-4614-3223-4_5

13. Cunningham SJ, Bainbridge D, Falconer A (2006) More of an art than a science: Supporting the creation of playlists and mixes. In: Proceedings 7th International Symposium on Music Information Retrieval (ISMIR 2006)
14. Flexer A, Schnitzer D, Gasser M, Widmer G (2008) Playlist generation using start and end songs. In: Inproceedings International Symposium on Music Information Retrieval (ISMIR 2008)
15. Freudenthaler C, Schmidt-Thieme L, Rendle S (2011) Bayesian factorization machines. In: Proceedings of NIPS workshop on sparse representation and low-rank approximation
16. Han BJ, Rho S, Jun S, Hwang E (2010) Music emotion classification and context-based music recommendation. Multimed Tools Appl 47(3):433–460
17. Hariri N, Mobasher B, Burke R (2012) Context-aware music recommendation based on latent topic sequential patterns. In: Proceedings of the Sixth ACM conference on recommender systems, RecSys '12. ACM, pp 131–138. https://doi.org/10.1145/2365952.2365979
18. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings 2008 Eighth IEEE International Conference on Data Mining (ICDM 2008), pp 263–272
19. Jannach D, Kamehkhosh I, Bonnin G (2014) Analyzing the characteristics of shared playlists for music recommendation. In: RSWeb@ RecSys
20. Jannach D, Lerche L, Kamehkhosh I (2015) Beyond 'hitting the hits": Generating coherent music playlist continuations with the right tracks. In: Proceedings of the 9th ACM conference on recommender systems, pp 187–194
21. Jolliffe I (1986) Principal component analysis. Springer
22. Juan Y, Zhuang Y, Chin WS, Lin CJ (2016) Field-aware factorization machines for CTR prediction. In: Proceedings 10th ACM Conference on Recommender Systems (RecSys 2016), pp 43–50
23. Kamalzadeh M, Baur D, Möller T (2012) A survey on music listening and management behaviours. In: Proceedings 13th International Symposium on Music Information Retrieval (ISMIR 2012)
24. Kaminskas M, Bridge D (2016) Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Trans Interact Intell Syst 7(1):2:1–2:42. https://doi.org/10.1145/2926720
25. Kaminskas M, Ricci F (2012) Contextual music information retrieval and recommendation: State art and challenges. Comput Sci Rev 6(2):89–119
26. Kaminskas M, Ricci F, Schedl M (2013) Location-aware music recommendation using auto-tagging and hybrid matching. In: Proceedings 7th ACM Conference on Recommender Systems (RecSys 2013), pp 17–24
27. Kim JY, Belkin NJ (2002) Categories of music description and search terms and phrases used by non-music experts. In: Proceedings 3rd International Society for Music Information Retrieval Conference (ISMIR 2002), pp 209–214
28. Koren Y (2008) Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proceedings 14th ACM International conference on knowledge discovery and data mining (KDD 2008), pp 426–434
29. Koren Y (2009) Collaborative Filtering with Temporal Dynamics. In: Proceedings 15th ACM International conference on knowledge discovery and data mining (KDD 2009), pp 447–456
30. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems computer 42(8)
31. Lee JH, Downie JS (2004) Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In: Proceedings 5th International Society for Music Information Retrieval Conference (ISMIR 2004)
32. Lee JH, Kim YS, Hubbles C (2016) A look at the cloud from both sides now: an analysis of cloud music service usage. In: Proceedings 17th International Society for Music Information Retrieval Conference (ISMIR 2016)
33. Leys C, Ley C, Klein O, Bernard P, Licata L (2013) Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. J Exp Soc Psychol 49(4):764–766
34. McFee B, Barrington L, Lanckriet G (2012) Learning content similarity for music recommendation. IEEE Transactions on Audio, Speech, and Language Processing 20(8):2207–2218
35. McFee B, Lanckriet GR (2012) Hypergraph models of playlist dialects. In: ISMIR, vol 12. Citeseer, pp 343–348
36. McVicar M, Freeman T, De Bie T (2011) Mining the correlation between lyrical and audio features and the emergence of mood. In: Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011), pp 783–788
37. Mikhail Trofimov AN (2016) tffm: Tensorflow implementation of an arbitrary order factorization machine https://github.com/geffy/tffm

38. Millecamp M, Htun NN, Jin Y, Verbert K (2018) Controlling spotify recommendations: effects of personal characteristics on music recommender user interfaces. In: Proceedings of the 26th Conference on user modeling, adaptation and personalization, pp 101–109
39. Miller GA (1998) WordNet: An electronic lexical database. MIT Press, Cambridge
40. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572
41. Pichl M, Zangerle E (2018) Latent feature combination for multi-context music recommendation. In: 2018 International conference on content-based multimedia indexing, CBMI 2018. IEEE, pp 1–6.
42. Pichl M, Zangerle E, Specht G (2015) Towards a context-aware music recommendation approach: What is hidden in the playlist name? In: 15th IEEE International conference on data mining workshops, pp 1360–1365
43. Pichl M, Zangerle E, Specht G (2016) Understanding playlist creation on music streaming platforms. In: 2016 IEEE International Symposium on Multimedia (ISM). IEEE, pp 475–480
44. Pichl M, Zangerle E, Specht G (2017) Improving context-aware music recommender systems: Beyond the pre-filtering approach. In: Proceedings of the 2017 ACM on International conference on multimedia retrieval, ICMR 2017. ACM, pp 201–208. https://doi.org/10.1145/3078971.3078980
45. Pichl M, Zangerle E, Specht G (2017) Understanding user-curated playlists on Spotify: A machine learning approach. Int J Multimed Data Eng Manag IJMDEM 8(4):44–59. https://doi.org/10.4018/IJMDEM.2017100103
46. Rendle S (2012) Factorization machines with libFM. ACM Intell Sys Technol 3(3):57:1–57:22
47. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), pp 452–461
48. Rendle S, Schmidt-Thieme L (2010) Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings Third ACM International Conference on Web Search and Data Mining (WSDM 2010), pp 81–90
49. Schedl M, Vall A, Farrahi K (2014) User geospatial context for music recommendation in microblogs. In: Proceedings 37th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2014)
50. Vall A, Dorfer M, Eghbal-Zadeh H, Schedl M, Burjorjee K, Widmer G (2019) Feature-combination hybrid recommender systems for automated music playlist continuation. User Model User-Adap Inter 29(2):527–572
51. Van den Oord A, Dieleman S, Schrauwen B (2013) Deep content-based music recommendation. In: Advances in neural information processing systems, pp 2643–2651
52. Wu G, Swaminathan V, Mitra S, Kumar R (2017) Digital content recommendation system using implicit feedback data. In: 2017 IEEE International Conference on Big Data (Big Data), pp 2766–2771. https://doi.org/10.1109/BigData.2017.8258242
53. Zangerle E, Chen C, Tsai M, Yang Y (2018) Leveraging affective hashtags for ranking music recommendations. IEEE Trans Affect Comput, pp 1–1. https://doi.org/10.1109/TAFFC.2018.2846596
54. Zangerle E, Gassler W, Specht G (2012) Exploiting Twitter's collective knowledge for music recommendations. In: Proceedings 2nd workshop on making sense of microposts (#MSM2012)
55. Zangerle E, Pichl M (2018) Content-based user models: Modeling the many faces of musical preference. In: 19th International society for music information retrieval conference
56. Zangerle E, Pichl M, Schedl M (2018) Culture-aware music recommendation. In: Proceedings of the 26th conference on user modeling, adaptation and personalization, UMAP '18. ACM, New York, pp 357–358, https://doi.org/10.1145/3209219.3209258

Martin Pichl has been a research assistant and data scientist at the University of Innsbruck in the research group Databases and Information Systems (DBIS), a subdivision of the Department of Computer Science between 2014 and 2018. He earned his master's degree in Information Systems at the University of Innsbruck in 2014 and his PhD in Computer Science at the University of Innsbruck in 2018. His main fields of expertise are information retrieval and recommender systems. In his PhD project, he investigated user-centric music recommender systems. For this, he focused on incorporating a user's context during music consumption into recommender systems to provide better and more suitable music recommendations in a given context. For his work, he received the best student paper award at the International Conference on Content-Based Multimedia Indexing 2018.

Eva Zangerle is a postdoctoral researcher at the University of Innsbruck at the research group for Databases and Information Systems (Department of Computer Science). She earned her master's degree in Computer Science at the University of Innsbruck and subsequently pursued her Ph.D. from the University of Innsbruck in the field of recommender systems for collaborative social media platforms. Her main research interests are within the fields of social media analysis, recommender systems, and information retrieval. Over the last years, she has combined these three fields of research and investigated context-aware music recommender systems based on data retrieved from social media platforms aiming to exploit new sources of information for recommender systems. She was awarded a Postdoctoral Fellowship for Overseas Researchers from the Japan Society for the Promotion of Science allowing her to make a short-term research stay at the Ritsumeikan University in Kyoto.