Latent Feature Combination for Multi-Context Music Recommendation

Martin Pichl
Department of Computer Science
Universität Innsbruck, Austria
martin.pichl@uibk.ac.at

Eva Zangerle
Department of Computer Science
Universität Innsbruck, Austria
eva.zangerle@uibk.ac.at

***Abstract*—In recent years, music aficionados have increasingly been consuming music via public music streaming platforms. Due to the size of the collections provided, music recommender systems have become a vital component as these aim to provide recommendations that match the user's current context as, throughout the day, users listen to music in numerous different contexts and situations. In this paper, we propose a multi-context-aware track recommender system that *jointly* exploits information about the current situation and musical preferences of users. To jointly model users by their situational and musical preferences, we cluster users based on their situational features and similarly, cluster music tracks based on their content features. Our experiments show that by relying on Factorization Machines for the computation of recommendations, the proposed approach allows to successfully leverage interaction effects between listening histories, situational and track content information, substantially outperforming a set of baseline recommenders.**

***Index Terms*—Music Recommender Systems, Context-aware Recommender Systems**

I. Introduction

Over the last decade, people have increasingly switched from listening to their private music collections to using music streaming platforms providing millions of tracks [1]. To increase usability, streaming platforms heavily rely on recommender systems (RecSys) to help users navigate through the provided collections to discover music they like. However, whether or not a user likes a recommended track 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 essential for providing suitable personalized music recommendations [2], [3] as people listen to different music during different activities [4] and create playlists that are intended for certain activities [5].

Recently, music streaming platforms started to provide user playlists publicly which allows to quantitatively study listening and organizational habits. On Spotify[1], all user-created playlists are public by default and thus, can be crawled. To this end, Pichl et al. [6], [7] proposed an approach for clustering contextually similar playlists by extracting contextual information from the names of playlists, ultimately allowing for finding playlists that users created for similar purposes and situations. The authors propose to leverage these clusters as an additional feature for a factorization machine-based RecSys. Furthermore, they perform an analysis of the acoustic features (e.g., tempo or danceability) of the tracks contained in individual playlists [8] and find that there are five different groups (archetypes) of playlists, described by their audio characteristics.

However, thus far, information about the situational context of a user has *not* been linked with acoustic feature-based playlist archetypes. In this work, we are particularly interested in how contextual and audio characteristics may *jointly* be leveraged for track recommendations. Therefore, we propose to make use of factorization machines (FM) [9] as these allow for exploiting latent features and interactions between input variables and hence, are a suitable choice for this task. This makes FMs well suited for the task of multi-context-aware RecSys as we aim at exploiting interaction effects between contextual clusters extracted from the names of playlists and acoustic feature-based clusters based on audio characteristics. In this paper, we present a novel FM-based user model combining situational context with acoustic context and refer to this model as *multi-context user model*. In several experiments, we show that a RecSys leveraging this proposed model substantially outperforms context-agnostic baselines and, more importantly, a context-aware RecSys that relies on either context- or acoustic feature-based clusters individually.

The main contribution of this work is twofold: firstly, we leverage two types of contextual information for the computation of multi-context-aware track recommendations that allow 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).

II. Related Work

For the computation of recommendations, *user-based collaborative filtering* has been shown to work well in the field of music recommender systems [6], [10], [11]. User-based CF relies on a user-item matrix as input. This matrix, containing ratings of users for items, is exploited to group users based on their rating behavior and hence, to find similar users. Based on nearest neighbors, items for a given user are recommended

[1] http://www.spotify.com

978-1-5386-7021-7/18/$31.00

by choosing the items these neighbors rated favorably and that are new to the user. CF-based approaches utilizing matrix factorization (MF) techniques have been shown to yield better recommendation accuracies (e.g., [12]). Those approaches are also known as *latent factor models*, as factorizing the user-item matrix yields a latent representation of user-item ratings on a more abstract level (e.g., by applying Singular Value Decomposition (SVD) [12]). Several extensions to MF have been introduced (e.g., for implicit feedback data [13], [14] or for context-aware recommendations [15], [16]).

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 [17]. 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 states) [2], [3]. As for the different types of contexts, Kaminskas and Ricci [18] 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 (e.g., [19], [20]) or the user's location (e.g., [21], [22]).

A recent enhancement of CF are factorization machines (FM) [9]. FMs combine the advantages of support vector machines (SVM) with factorization models. Factorization enables the FM to model all interactions between variables in linear time [9], 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).

In this work, we present a multi-context matrix factorization approach. We utilize SVD to represent the user context in a latent feature space and 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 a FM-based RecSys.

III. Problem Formulation

In the following, we formally define the problem tackled in this paper, namely the *context-aware track recommendation problem*. The basic input for such a RecSys is a user-item matrix R, which holds prior ratings for items by users. It consists of m rows (corresponding to the number of users) and n columns (corresponding to the number of tracks). The elements $r_{i,j}$ 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 1. The utility function f_R assigns predicted ratings $\hat{r}_{i,j}$ to unrated <user,track>-pairs. 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 as SVD [23] as depicted in Equation 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 $\hat{r}$ can be estimated using the dot product of the feature vector of the user $\vec{u_i}$ and the feature vector of the item $\vec{v_j}$: $\hat{r}_{u,i} = \vec{u_i} \cdot \vec{v_j}$.

Prior research has found that people listen to different music during different activities [4] and people create playlists that are intended for certain activities [5]. Hence, depending on different user contexts, different tracks need to be recommended. This problem can be formulated as depicted in Equation 3, where f_{CR} is a utiliy function assigning predicted ratings $\hat{r}_{i,j}$ to user u for track i given user contexts c [17].

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

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.

IV. Dataset

Throughout our experiments, we leverage a publicly available dataset containing Spotify playlists [8]. In a first step, we apply the proposed dimension reduction and clustering methods on the initial dataset to obtain the proposed acoustic feature and situational clusters. This results in a dataset containing <user,track,SC,AC,rating,acoustic features>-tuples. We also add the seven individual acoustic features for each track (AF) provided by the Spotify API[2], as we also aim to use these as a baseline approach in the course of our experiments. In a next step, we assign each track a rating value r as described in Section V. The rating indicates whether a certain user listened to a certain track in a certain situational cluster ($r = 1$) or not ($r = 0$). Please note that a user might listen to the same song in different situations, whereas a track always belongs to the same acoustic feature-based cluster. 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 in the playlist name for). On average, a user in the dataset listens to 770.19 tracks (SD=2,168.62, Median=264.50).

V. Proposed Multi-Context RecSys

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

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

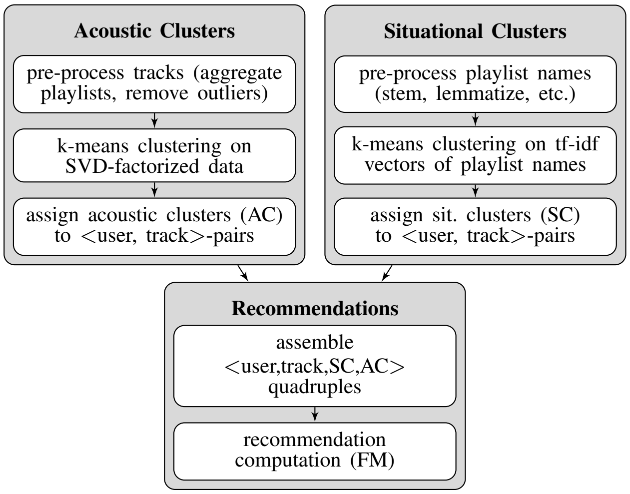

Fig. 1. Workflow for Computing Recommendations

between different contexts, between users and contexts and between tracks and contexts.

As input for the proposed approach, we require a dataset of playlists (i.e., sets of tracks) assembled by users as presented in Section IV. We assume that by adding a track to a playlist, the user expresses some preference for the track. For means of simplicity, we refer to such a user-track interaction as "a given user listened to a given track". Each playlist is tagged with a user-defined name that describes the content of the playlist. Based on such a dataset, we propose to compute two types of contextual information for the computation of context-aware track recommendations: (i) *playlist archetypes* and (ii) *situational clusters*, which we describe in the following.

The additional context information allows to model *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 into a context-aware RecSys tackling the problem as stated in Section III, we propose a model based on factorization machines (FM) [9]. 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. An overview of the proposed approach is given in Figure 1, where the steps taken to extract contextual information that is leveraged in the recommendation computation are outlined.

A. Playlist Archetypes

The proposed approach relies on clusters of playlists (archetypes) that share similar acoustic features (e.g., the tempo of the tracks contained). In a first step, we process and aggregate seven standard acoustic features obtained via the Spotify API[3] as proposed by previous research [8], [24]. These content features are extracted and aggregated from the audio signal of a track and comprise: *danceability* (how suitable a track is for dancing), *energy* (perceived intensity and activity), *speechiness* (presence of spoken words in a track), *acousticness* (confidence whether track is acoustic), *instrumentalness* (prediction whether track contains no vocals), *tempo* (in beats per minute) and *valence* (musical positiveness conveyed). Next, we aggregate the acoustic features of each track per playlist using the arithmetic mean and remove the outliers as proposed by Pichl et al. [8]. The result of this aggregation step is a lower dimensional $m \times n$ matrix AFM, where each row represents a playlist and each column represents an acoustic feature. 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) [25] via SVD [26]. Based on the principal components (PCs) obtained by the conducted PCA, we explain differences in playlists and, more importantly, estimate the number of acoustic features clusters (ACs), by the explained variance of each PC (squared singular values s_i^2 (diagonal of Σ)). Having obtained the number of ACs $k = 5$ (the accumulated variance of the principal components is 85.64 and hence greater than the 80% threshold at this point) we compute the clusters by applying k-means on the dimension-reduced matrix AFM. The clustering assigns each playlist and hence, implicitly each track, to a playlist archetype that allows capturing a user's preferences towards certain types of music. We depict the result of this approach in Figure 2, where each playlist is represented by an integer that represents the cluster assignment. The clusters are marked by individual colors and numbers and are annotated with the according acoustic feature. We observe that playlists that are highly influenced by instrumental and acoustic features are separated from the remaining playlists by the first PC. 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 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 context-aware track recommendations.

B. 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 [6], [7], we mine for activities and other descriptors (seasons, events, etc.) in the names of playlists. As depicted in Figure 1, we firstly stem and lemmatize all playlist names. Next, we remove stop words and non-contextual terms (e.g., genre, artist and track names) as these do not provide any contextual information. For the resulting bags of lemmata describing each playlist, we compute the term frequency-inverse document frequency (tf-idf) for each bag of lemmata

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

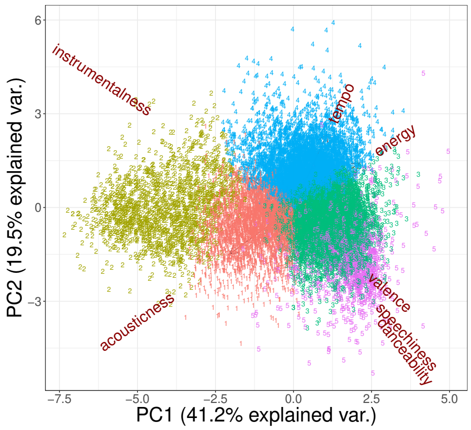

Fig. 2. Latent Representation of Playlist Clusters

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. As we evaluate our approach using the same dataset as [6] (cf. Section VI), we set the number of clusters to $k = 23$, as proposed in the original approach. This provides us with situational clusters capturing in which context a user listened to certain tracks that we incorporate in a RecSys in the next step.

C. Recommendation Computation

The previous steps provide us with information about (i) a user's preference for playlist archetypes and (ii) the situational context in which a user listens to certain tracks. This information is extracted in the form of user-cluster assignments. We now combine these clusters and the listening history of users to compute track recommendations. Particularly, we propose to utilize FMs [9] 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 assigning the corresponding contextual clusters to each user-track pair, now forming <user,track,AC,SC>-tuples. By adding a fifth column $rating$ to the dataset, we derive the input matrix R for our rating prediction problem to be solved by the FM: for each unique <user,track,AC,SC>-tuple, the $rating$ r_{ijsc} is 1 if a user i has listened to a track j in situational context s in a playlist belonging to archetype c. Our dataset does not contain any implicit feedback by users (i.e., play counts, skipping behavior or session durations). Therefore, we cannot estimate any preference towards an item as i.e., proposed by [13]. However, adding a track to a playlist is a clear preference for the song for us. However, as no weighting is possible, we assign binary preferences: for each <user,track,AC,SC>-tuple, for which we cannot obtain a rating for, we assume the rating to be $r = 0$ (as proposed by [27]). In our dataset (as presented in Section VI), we observe that the class distribution of relevant and irrelevant tracks is highly imbalanced as naturally, users only listen to a small fraction of the songs available. Therefore, we rely on oversampling in order to achieve a 1:1 ratio between relevant and irrelevant tracks on which we train and test our classifiers to avoid bias towards negative values.

Based on this data, 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 moreover 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 4, where we show that the proposed 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 as logistic regression do, FMs rely on factorization to model the interaction as the inner product of low dimensional vectors ($\langle \vec{v_i}, \vec{v_j} \rangle$) [9].

$$\hat{r}_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j \tag{4}$$

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 [9]. For the top-n recommendations task, we consider all tracks with $\hat{r} < 0.5$ and hence, all tracks with a predicted rating below 0.5 as irrelevant. 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 n recommendations.

VI. Evaluation

A. Evaluation Methodology

We assess the performance of recommendation models in a 5-fold cross evaluation with random sampling, where we compute a predicted rating $\hat{r}$ for all tracks in the 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 averaged over all folds. For assessing the performance of the rating prediction task for the different recommendation models, we compute the *root mean squared error* (RMSE). For assessing the performance of different recommendation models on the top-n recommendations, 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 *precision*, *recall* and F_1 measures.

B. Evaluated Recommender Systems

To assess the effects of incorporating different contextual information encoded as clusters into a recommendation system,

TABLE I
OVERVIEW OF EVALUATED MODELS AND RATING PREDICTION EVALUATION RESULTS

Model	CF	AF	AC	SC	RMSE
R					0.5
MP					0.71
CF	✓				0.75
AF	✓	✓			0.44
SC	✓			✓	0.72
AC	✓		✓		0.57
AF+AC	✓	✓	✓		0.47
AF+SC	✓	✓		✓	**0.40**
AC+SC	✓		✓	✓	**0.40**

we propose to evaluate a theoretical random baseline, three baseline approaches and a set of different extended models.

Besides the random baseline, we employ a set of three baseline methods: (i) a non-model-based approach that recommends the most popular tracks (MP) of each situational cluster; (ii) a 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 [28]. Please note that here, we use the individual acoustic features of all tracks and do not rely on acoustic feature clusters in this model.

Table I gives an overview of all evaluated models and the according input data. We derive a set of extended models utilizing the situational clusters mined from the playlist names (SC) and playlist context derived from acoustic feature clusters (AC). 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 both *AF+AC*. Finally, we evaluate a multi-context-aware model that additionally combines both clusters (*AC+SC*) and a model incorporating the situational clusters (SC) mined from the playlist names, the *AF+SC* model.

C. Results and Discussion

In the following, we firstly discuss the results of the top-n evaluation followed by the results of the rating prediction task.

1) Top-n Recommendations: 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 [29]. Therefore, we evaluate the top-n performance of the proposed RecSys for a small number of n. Figure 3 depicts recall, precision and F_1 for $n = 1 \ldots 10$. 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-score of 0.89 by 3.70%. We observe that all model-based approaches outperform the MP-baseline. 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.

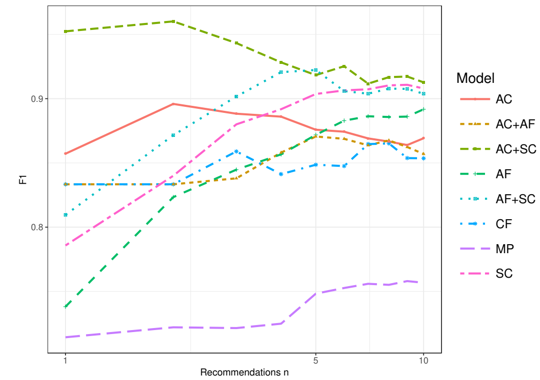

Fig. 3. F_1-Evaluation for $n = 1 \ldots 10$

Generally, we observe that models incorporating acoustic features along with situational clusters provide the best performance independently of the number of recommendations n. Besides that, we show that for a small number of recommendations n ($n \leq 10$), incorporating situational clusters is highly important and outweighs acoustic features. Moreover, the AC model leveraging acoustic clusters performs better that the AF model leveraging all acoustic features for small numbers of n. The long tail includes several popular tracks with high play counts, but many more tracks with low play counts (i.e., niche music). ACs group users who enjoy to listen to similar music, which is sufficient for small n. However, to recommend tracks from the long tail, recommending popular tracks does not suffice. For this scenario, a RecSys needs to accurately model the user's preferences by incorporating audio features (AF) of the tracks in the listening history of each user. Our experiments show that additionally incorporating the situational context (SC) improves the recommendation accuracy for both, short and long lists of recommendations. 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., [16], [20]).

2) Evaluation of the Predicted Ratings: We also evaluated a rating prediction task. The FM-component in our RecSys computes a predicted rating $\hat{r}$, i.e., the probability of a user listening to a certain track in a certain situational cluster. Please note that this naturally also impacts the ranking of items as we rank the items based on the predicted rating $\hat{r}$ and consider all items with a predicted rating $\hat{r} < 0.5$ as irrelevant and sort the remaining items in descending order by $\hat{r}$.

We provide the results of the rating prediction measures computed over the whole test set in Table I. Our results show that the AC+SC and the AF+SC models achieve the lowest error rates across all error measures. Hence, 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 are highly beneficial for context-aware track recommendations.

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 model-based FM approaches estimate $\hat{r}$, the probability whether a given user has listened to a given track in a given situational cluster. Ultimately, for non-relevant and correctly classified tracks in the test set, the error is 0 for the most popular approach, whereas there is an error for the model-based approaches (although the track is correctly classified). This is, as all tracks with a predicted rating $\hat{r} < 0.5$ are classified as irrelevant which yields a true positive for the classification-based measures, but the rating prediction measures indicate an error in the range between 0 and 0.5.

3) Estimating the Interaction Effects: Finally, we are also interested in estimating the impact of interaction effects on the recommendation quality. Therefore, we compare the performance of a FM that does not exploit any interaction effects and a FM that leverages interaction effects (as utilized in the previous experiments) based on the best user model detected (AC+SC). Our experiments show that adding interaction effects allows for a 17.41% higher F_1-score (0.88 vs. 0.75) and a lower RMSE (0.41 vs. 0.67). This again strengthens our hypothesis that those effects are beneficial in such a scenario.

VII. Conclusion and Future Work

We presented a multi-context-aware RecSys, jointly exploiting (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. In an offline evaluation, we show that (i) the integration of situational context improves the precision of music RecSys and that (ii) acoustic features and thereby, a user's musical taste, are particularly suitable to retrieve tracks a user likes from the long tail. We show that interaction effects between situational context and musical preferences provides the most accurate recommendations and simultaneously covers the long tail well. As for future work, we believe that the use of FMs allows for easily extending our current approach with further notions of context (e.g., emotion or culture).

References

[1] 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 *Proc. 17th Intl. Society for Music Information Retrieval Conf. (ISMIR 2016)*, 2016.

[2] J.-Y. Kim and N. J. Belkin, "Categories of music description and search terms and phrases used by non-music experts." in *Proc. 3rd Intl. Society for Music Information Retrieval Conf. (ISMIR 2002)*, 2002, pp. 209–214.

[3] J. H. Lee and J. S. Downie, "Survey of music information needs, uses, and seeking behaviours: Preliminary findings." in *Proc. 5th Intl. Society for Music Information Retrieval Conf. (ISMIR 2004)*, 2004.

[4] M. Kamalzadeh, D. Baur, and T. Möller, "A survey on music listening and management behaviours," in *Proc. 13th Intl. Symposium on Music Information Retrieval (ISMIR 2012)*, 2012.

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

[6] M. Pichl, E. Zangerle, and G. Specht, "Towards a context-aware music recommendation approach: What is hidden in the playlist name?" in *15th IEEE Intl. Conf. on Data Mining Workshops*, 2015, pp. 1360–1365.

[7] ——, "Improving context-aware music recommender systems: Beyond the pre-filtering approach," in *Proc. 2017 ACM on Intl. Conf. on Multimedia Retrieval (ICMR 2017)*, 2017, pp. 201–208.

[8] ——, "Understanding Playlist Creation on Music Streaming Platforms," in *Proc. IEEE Symposium on Multimedia (ISM 2016)*, 2016.

[9] S. Rendle, "Factorization machines with libFM," *ACM Intelligent Systems and Technology*, vol. 3, no. 3, pp. 57:1–57:22, May 2012.

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

[11] M. Schedl, A. Vall, and K. Farrahi, "User Geospatial Context for Music Recommendation in Microblogs," in *Proc. 37th Intl. ACM Conf. on Research and Development in Information Retrieval (SIGIR 2014)*, 2014.

[12] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM Intl. Conf. on Knowledge Discovery and Data Mining (KDD 2008)*, 2008, pp. 426–434.

[13] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 2008 Eighth IEEE Intl. Conf. on Data Mining (ICDM 2008)*, 2008, pp. 263–272.

[14] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. on Uncertainty in Artificial Intelligence (UAI 2009)*, pp. 452–461.

[15] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. 15th ACM Intl. Conf. on Knowledge Discovery and Data Mining (KDD 2009*, 2009, pp. 447–456.

[16] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. Fifth ACM Conf. on Recommender Systems (RecSys 2011)*, 2011, pp. 301–304.

[17] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, 1st ed., F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York, NY, USA: Springer-Verlag New York, Inc., 2010, ch. 7, pp. 217–253.

[18] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State art and challenges," *Computer Science Review*, vol. 6, no. 2, pp. 89–119, 2012.

[19] B.-j. Han, S. Rho, S. Jun, and E. Hwang, "Music emotion classification and context-based music recommendation," *Multimedia Tools and Applications*, vol. 47, no. 3, pp. 433–460, 2010.

[20] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lke, and R. Schwaiger, "Incarmusic: Context-aware music recommendations in a car," in *E-Commerce and Web Technologies*, ser. LNBIP. Springer, 2011, vol. 85, pp. 89–100.

[21] M. Kaminskas, F. Ricci, and M. Schedl, "Location-aware music recommendation using auto-tagging and hybrid matching," in *Proc. 7th ACM Conf. on Recommender Systems (RecSys 2013)*, 2013, pp. 17–24.

[22] Z. Cheng and J. Shen, "Just-for-Me: An Adaptive Personalization System for Location-Aware Social Music Recommendation," in *Proc. 2014 ACM Intl. Conf. on Multimedia Retrieval (ICMR 2014)*, 2014.

[23] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[24] 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 2017, Bucharest, Romania*. ACM, 2017, pp. 201–208. [Online]. Available: https://www.evazangerle.at/wp-content/uploads/2017/06/acm-icmr-2017.pdf

[25] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

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

[27] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Proc. 2008 Eighth IEEE Intl. Conf. on Data Mining (ICDM 2008)*, 2008, pp. 502–511.

[28] B. McFee, L. Barrington, and G. Lanckriet, "Learning content similarity for music recommendation," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 8, pp. 2207–2218, 2012.

[29] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus, "Understanding choice overload in recommender systems," in *In Proc. 4th ACM Conf. on Recommender Systems (RecSys 2010)*, 2010, pp. 63–70.