Personalized Next-Track Music Recommendation with Multi-dimensional Long-Term Preference Signals

Iman Kamehkhosh
TU Dortmund, 44227
Dortmund, Germany
iman.kamehkhosh@
tu-dortmund.de

Dietmar Jannach
TU Dortmund, 44227
Dortmund, Germany
dietmar.jannach@
tu-dortmund.de

Lukas Lerche
TU Dortmund, 44227
Dortmund, Germany
lukas.lerche@
tu-dortmund.de

ABSTRACT

The automated generation of playlists given a user's most recent listening history is a common feature of modern music streaming platforms. In the research literature, a number of algorithmic proposals for this "next-track recommendation" problem have been made in recent years. However, nearly all of them are based on the user's most recent listening history, context, or location but do not consider the users' long-term listening preferences or social network.

In this work, we explore the value of long-term preferences for personalizing the playlist generation process and evaluate different strategies of applying multi-dimensional user-specific preference signals. The results of an empirical evaluation on five different datasets show that although the short-term listening history should generally govern the next-track selection process, long-term preferences can measurably help to increase the personalization quality.

Keywords

Music Recommendation; Personalization; Evaluation

1. INTRODUCTION

Due to the massive volume of online music, relatively large personal music collections, and the dominant proportion of long tail items in the music domain [20], music recommender systems have gained popularity over time. Nowadays, music recommendation is a typical feature of web platforms and player applications like iTunes, Deezer or Spotify.

Generally, music recommender systems can be applied to recommend different music-related items such as albums, artists, or concerts. In this work, we focus on a specific type of music recommendation: given a history of recent tracks played by a user, our goal is to recommend a (personalized) list of tracks to be played next. We refer to such recommendations as "next-track recommendations", which can both be used to generate playlist continuations and to implement virtually endless radio stations given a number of seed tracks.

Proceedings of the 1st Workshop on Multi-dimensional Information Fusion for User Modeling and Personalization (IFUP 2016), collocated with ACM UMAP 2016, 16/07/2016, Halifax, Canada. Copyright held by the authors.

In recent years, a number of next-track recommendation algorithms have been proposed, among them in particular context-aware or location-aware ones or approaches that use musical features, meta-data, or social tags [9, 15, 25]. However, even though *long-term* user preferences intuitively play a key role in personalizing the recommendations, nearly all the playlist generation approaches reviewed in the recent surveys in [7, 18] only consider the user's *short-term* listening history.

In this work, we therefore aim to explore the value of long-term preferences for personalizing the next-track recommendation process. More precisely, instead of basing the recommendations solely on the sequence of seed tracks from the users' recent listening histories, as done, e.g., in [7], [12], or [15], we incorporate *multi-dimensional* signals extracted from the users' long-term preferences and their social network into the recommendation process.

In particular, we look at a user's long-term preferences to identify and recommend (a) tracks that the user liked in the past (*track repetition*), (b) tracks of artists that the user liked in the past (*favorite artists*), (c) tracks that are semantically similar to past liked ones (*topic similarity*), and (d) tracks that are often played together with those that the user likes (*track co-occurrence*). In addition, we look for (e) tracks and artists that the user's *social friends* liked and consider them for recommendation. We use then a multi-faceted scoring scheme to incorporate this heterogeneous information in the recommendation process.

Our focus is to optimize an accuracy criterion and we use the track *hit rate* (recall) to assess to which extent our algorithms are capable of selecting tracks that would also be chosen by the users. Our results show that several of the considered personalization signals (e.g., track co-occurrence patterns in the users' long-term preferences or favorite track repetitions) can help to increase the hit rate. Since accuracy is not the only quality criterion for music recommenders, we will also discuss the effect of different personalization strategies on the *diversity* of the resulting recommendations and the *coherence* of the recommendations with the user's last played tracks.

2. PERSONALIZED NEXT-TRACK RECOMMENDATION TECHNIQUES

In this section, we present the details of the proposed personalization approaches for next-track music recommendation. A general assumption of our approaches is that in typical applications the tracks that were most recently played or most recently added to a playlist should primarily govern the selection of the immediate next tracks. Personalization can then be helpful to further adjust the ranking of the tracks based on the user's long-term preferences or other user-specific signals. Recommending tracks of one of the user's most favorite rock artists can, for example, generally be a good strategy because many users tend to repeatedly listen to their favorite artists. If the most recently played tracks belong however to the genre "rap", recommending rock hits – even though the user might generally like each of them – might lead to a poor recommendation quality and user experience.

2.1 General Scoring Scheme and Terminology

In our work, we use a general scoring scheme that combines a *baseline (main) algorithm* – which focuses on the short-term profile – with *personalization components*. Given a short-term listening history h (a sequence of tracks), our goal is to determine a relevance score for each possible next track t^* using different personalization signals.

Similar to [17], we use a weighted scoring scheme to combine the score of the baseline algorithm with additional personalization scores. The overall relevance score $score_{overall}$ for a possible next track t^* given h is computed as

$$score_{overall}(h,t^*) = w_{base} \cdot score_{base}(h,t^*) + \sum_{pers \in P} w_{pers} \cdot score_{pers}(h,t^*) \quad (1)$$

where P is a set of personalization strategies, each with a different weight w_{pers}, and w_{base} is the weight of the baseline algorithm. The $score_{base}$ and $score_{pers}$ are functions to compute the baseline score and the scores of the individual personalization components, respectively. The selection of the scores and their weights both depend on the available data and the goals that should be achieved, see Section 2.3.6.

Note that for listening logs, the term *short-term history* (h) refers to the most recent (current) listening session of a user and the *long-term history* (lth) refers to all listening sessions of a user before the current session (h). For playlists, the term *short-term history* (h) refers to a playlist beginning and the *long-term history* (lth) refers to all other shared playlist by the same user. Since the terms *listening session* and *playlist* both represent sequences of tracks, we will use them interchangeably in the algorithm descriptions.

2.2 Short-Term Preference Model: Baseline Algorithm

A comparison of various playlister algorithms in [7] showed that in particular for recommendation lists of shorter lengths a k-Nearest-Neighbor-based (kNN) approach outperformed other and often more complex algorithms, including Bayesian Personalized Ranking (BPR) [27], in terms of accuracy for this task. As we are interested in high-quality next-track recommendations, we will use a kNN-based approach as the baseline algorithm as done in [17].

kNN was also used as a baseline in [15]. Small hit rate increases were achieved at long list lengths by combining it with tag-based track information. In [15] however, a very small value for $k = 10$ was used. Since our experiments show that larger values for k ($k = 300$) lead to significantly higher hit rates, we will use kNN with $k = 300$ as a baseline.

The kNN-based approach takes the recent listening history h as an input and looks for other listening sessions in the training data that contain the same tracks[1]. The main assumption is that if there are additional tracks in a similar past session, chances are good that these tracks suit the current listening history h, too. We refer to these similar sessions as "neighbors".

Technically, given a short-term history h, we first compute the binary cosine similarity of h and the other sessions from the training data. The similarity values are then sorted and a set N_h of nearest neighbor sessions of h is determined. The kNN score of a target track t^* is then computed as the sum of the similarity values of h and neighbor sessions $nbr \in N_h$ which contain t^* (Equation 2). Note that the indicator function $1_{nbr}(t^*)$ returns 1 if nbr contains t^* and 0 otherwise.

$$score_{kNN}(h,t^*) = \sum_{nbr \in N_h} sim_{cosine}(h,nbr) \cdot 1_{nbr}(t^*) \quad (2)$$

2.3 Long-Term Preference Models: Personalization Approaches

In the following sections we propose a number of ways to compute the additional user-specific relevance scores.

2.3.1 Track Repetition

Listening to one's favorite tracks repeatedly over time is common in the music domain. To assess the importance of repetitions, we examined the listening histories of 2,000 Last.fm and Twitter users as follows. Given the list of tracks of a user's listening history $T = [t_1, t_2, \cdots, t_n]$, we measure the *proportion of repetitions* $Rep_p(u,T)$ in the listening history of a user u as proposed in [29]:

$$Rep_p(u,T) = 1 - \frac{|T_{uniq}|}{|T|} \quad (3)$$

where $|T_{uniq}|$ is the number of *unique* tracks in the user's listening history and $|T|$ is the length of his listening history.

The results show that, on average, more than 25% of the tracks that the users listen to are repeated tracks. Therefore, recommending and playing tracks that the user has listened to in the past is a simple but promising personalized strategy. To operationalize this idea we have to answer the questions *whether or not* to play known tracks and *which* tracks to select.

Using a coarse classification scheme, we could consider a user – during a listening session – to be either in *exploration* mode when she would like to discover new songs [19] or in *repetition* mode when she prefers to listen to her favorite tracks. To assess the general value of distinguishing between the two modes, we use the following simple heuristic to determine the user's mode and to decide whether or not to include track repetitions: If a pre-defined proportion of the user's recent history (e.g., 50%) consists of tracks that the user has played in the past, we assume that the user prefers to listen to known tracks. More elaborate schemes are of course plausible [19].

Different strategies are also possible to determine *which* tracks from the user's long-term history to recommend. In this paper, we examined the following approaches given a recent listening history h:

[1]As mentioned above, we will use the terms "listening history" or "listening session" in the following descriptions even though in some of the experiments later on "other shared playlists" of a user are meant.

R1: Repeat tracks from the user's history that are *generally popular* in the community in terms of play-counts.

R2: Repeat tracks which have been listened to by the user *at the same time of the day* as the tracks from h.

R3: Repeat tracks which have been performed by the same *artists* as in h.

R4: A weighted combination of the time-based (R2) and artist-based (R3) strategies.

To be able to combine these approaches with the baseline method as described in Section 2.1, various ways of computing a repetition score $score_{rep}$ are possible. In our experiments, we first build a set R_u of repetition candidates for each user u using one of the above mentioned strategies (R1-R4). Given the current listening history h of u, we then compute a score for a target track t^* based on the *recency* of the past listening event and the number of times t^* has been repeated in the user's long-term history.

The experiments on different datasets reported in [2] reveal a typical tendency of users to *re-consume* or repeat more recently-consumed items. Therefore, as a score, we use the *weighted relative recency* of the last time point $ts(t^*)$ when the target track t^* was played by the same user (Equation 4). If a candidate track was not played by the user before, the score is zero. Otherwise, it is the relation between $ts(t^*)$ and the timestamp of the beginning of the current listening session $ts(h)$ weighted by the number of times the target track has been repeated in the user's long-term history $cnt_{lth}(t^*)$. To further increase the relative importance of the tracks in the recent history, an exponential decay function can also be applied as suggested in [2].

$$score_{rep}(h, t^*) = cnt_{lth}(t^*) \cdot \frac{ts(t^*)}{ts(h)} \cdot 1_{R_u}(t^*) \tag{4}$$

2.3.2 Favorite Artists

Music lovers typically have their favorite artists and online services like Spotify quite obviously recommend tracks performed by artists that a user played in the past. In the proposed "Favorite Artists" personalization approach we adopt this idea and assign higher preference scores to tracks by a user's favorite performers. However, instead of only considering artists that the user actually knows, we also consider artists that are *similar* to the known ones.

Technically, we extend the CAGH (Collocated Artists – Greatest Hits) method from [7], which assigns higher scores to the most popular tracks of artists that are similar to those appearing in the recent listening history.

$$score_{CAGH}(h, t^*) = \sum_{b \in A_h} sim_{artist}(a_{t^*}, b) \cdot cnt(t^*) \tag{5}$$

A_h is the set of artists in current history, $cnt(t^*)$ is the number of occurrences of t^* in the training data, and a_{t^*} is the artist of the target track. As a measure of similarity of two artists $sim_{artist}(a_{t^*}, b)$, we count how often two artists appear together in the sessions of the training set [7].

To personalize the CAGH method, we extend it by also taking into account artists that are similar to those that were played in the user's long-term history. The favorite-artist score $score_{favA}$ of a target track t^* with artist a_{t^*} is then computed as a weighted combination of the original CAGH score (Equation 5) and its application on the long-term history lth of the user:

$$\begin{aligned} score_{favA}(h, t^*) = \alpha \cdot score_{CAGH}(h, t^*) \, + \\ (1-\alpha) \cdot \sum_{b \in A_{lth}} sim_{artist}(a_{t^*}, b) \cdot cnt(t^*) \end{aligned} \tag{6}$$

where A_{lth} is the set of artists of the tracks in the long-term history, a_{t^*} is the artist of the target track, and $cnt(t^*)$ as well as $sim_{artist}(a_{t^*}, b)$ are computed as described above. The weight factor α is used to balance the relative importance of the two factors.

2.3.3 Topic Similarity

Users of music websites like Last.fm are able to annotate tracks with tags. Such tags often describe the genre, mood, artist, or the release year of a track and these social tags can be used for detecting the topic of a certain playlist [15]. Our assumption is that at least some users have long-term preferences to listen to tracks that relate to a certain topic or tracks that have the same tags.

In [17], it was shown that considering the topical match of target tracks with the history (using tags from Last.fm) can be helpful to improve the recommendation accuracy. Two versions of a "content-based" recommendation method using averaged TF-IDF (Term Frequency/Inverse Document Frequency) vectors were proposed. One focused only on the recent listening history and one also included the long-term listening preferences.

Given a recent history h, we compute a tag-based similarity score $score_{tag}$ as the cosine similarity of the averaged TF-IDF vector of the recent history and the TF-IDF vector of the target track t^*. In our work, we use the extended version of the $score_{tag}$ for personalization. We define the topic-similarity score $score_{topicSim}$ as a weighted combination of $score_{tag}$ and its extended version considering the long-term history as follows:

$$\begin{aligned} score_{topicSim}(h, t^*) = & \beta \cdot score_{tag}(h, t^*) \, + \\ & (1-\beta) \cdot sim_{cosine}(\overline{TF}_{lth}, \vec{t^*}) \end{aligned} \tag{7}$$

where $\overline{TF}_{lth}$ is the averaged TF-IDF vector of all playlists (listening sessions) of the user's long-term history and $\vec{t^*}$ is the vector representation of t^*. The weight factor β is used to balance the relative importance of the two factors.

2.3.4 Extended Track Co-occurrences

Similar to the long-term content-based model in Equation 7, we propose to extend the kNN-based scheme (Equation 2) in a way that it considers the user's long-term listening history in combination with the recent history when determining the nearest neighbors.

In particular, for each listening session from the user's long-term history $s \in lth$, we determine a set N_s of similar sessions (nearest neighbors) as described in Section 2.1. The extended long-term kNN score kNN_{lth} of a target track t^* is then computed as the sum of the similarity values of *all* the user's sessions $s \in lth$ and their respective neighbor sessions $nbr \in N_s$ which contains t^*, i.e., $1_{nbr}(t^*) = 1$ if nbr contains t^* and 0 otherwise.

$$score_{kNN_{lth}}(lth, t^*) = \sum_{s \in lth} \sum_{nbr \in N_s} sim_{cosine}(s, nbr) \cdot 1_{nbr}(t^*) \tag{8}$$

2.3.5 Social Friends

The final personalization approach explored here leverages the listening preferences of the *social friends* of a user. Our hypothesis is that the preferences and behavior of friends can have an influence on a user.

In one of our experiments described in the next section, we will therefore utilize the listening logs of a user's Twitter friends[2] and incorporate the favorite tracks of friends into the next-track selection process. We consider a track as a favorite, if a user listened to it at least n times.

To compute the respective personalization score, we first build a set FF_u of favorite tracks for the friends of u. For a target track t^*, the Favorites-of-Friends score $score_{fof}$ is then a weighted sum of the occurrences of t^* in the favorite tracks of the user's friends $cnt_{ff}(t^*)$. A popularity weight for each friend $pop(f)$ can be computed based on, e.g., the number of the followers of each friend, which gives higher relevance to the tracks of more "popular" friends.

$$score_{fof}(h,t^*) = \sum_{ff \in FF_u} cnt_{ff}(t^*) \cdot pop(f) \qquad (9)$$

2.3.6 Combining the Scores

As mentioned above, we use a weighted scoring scheme to compute the final relevance score (Equation 1) as done in [17] and assume that the resulting personalization effect will lead to higher recommendation accuracy.

Generally, the selection of the scores and their weights both depend on the data that is available and the goals that should be achieved. If, for example, discovery is the main goal, including tracks from social friends might be suitable. If the goal is to generate a thematic (e.g., mood-based) playlist, a similarity-based approach seems more helpful.

In this work, we systematically tested different values for the weight of each score and selected the best ones (listed in Table 1) according to the accuracy results on the test set. In addition, to be able to combine the scores with different scales, we apply zero-one normalization by rescaling the scores to values between 0 and 1. Given a list of values E, the normalized score for an element $e_i \in E$ is computed as

$$normalized(e_i) = \frac{e_i - E_{min}}{E_{max} - E_{min}} \qquad (10)$$

where E_{max} and E_{min} are the maximum and minimum values of E respectively.

3. EXPERIMENTAL EVALUATION

We conducted a series of experiments to assess the value of personalizing the next-track recommendation process and the effectiveness of the proposed techniques both in terms of accuracy as well as artist diversity and coherence.

3.1 Datasets

We used three playlist datasets and two datasets that contain listening logs of several thousand users.

Playlists: The playlist datasets are obtained from three different platforms. The *Last.fm* dataset was collected using the public API of the service. The *Art-of-the-Mix* (AotM) data was published by [24] and contains playlists by music enthusiasts. The *8tracks* dataset was shared with us by the 8tracks platform. A particularity of the last dataset is that each playlist can only contain two tracks per artist. To be able to personalize the recommendations we created subsamples of these datasets such that there are at least 4 playlists for each user. We chose this comparably low threshold to have at least 1,000 playlists in the smallest dataset.

Table 1: Weights of personalization scores (w_{pers}) as well as the weights α from the extended CAGH version (Equation 6) and β from the topic similarity method (Equation 7) for different datasets. The weight of the baseline score in all combinations is set to one ($w_{base} = 1$), except for the $score_{kNN_{lth}}$, for which the baseline weights are listed in parenthesis after each w_{pers}. Note that not all methods can be applied for all datasets (indicated by a "–").

	#NP	30Music	Last.fm	AotM	8tracks
w_{rep}	2.0	0.2	–	–	–
w_{favA}	1.5	1.5	2.0	0.5	0.2
$w_{topicSim}$	–	–	0.3	0.3	0.3
$w_{kNN_{lth}}$	0.5 (0.5)	0.7 (0.3)	0.9 (0.1)	0.4 (0.6)	0.7 (0.3)
w_{fof}	0.3	–	–	–	–
α	0.9	0.9	0.9	0.4	0.8
β	–	–	0.9	0.6	0.7

Listening logs: The listening log datasets are sampled from two public datasets. First, we created a subset of the *#nowplaying* (NP) dataset which contains music-related tweets of users on Twitter[3]. Specifically, we sampled users who have posted at least 50 tweets over time. Second, we used the recent *30Music* dataset [28], which contains listening sessions retrieved from Internet radio stations through the Last.fm API. Similar to the playlist datasets, we created subsamples of these datasets in a way that there are 9 listening sessions for each user. Having sufficient past sessions allows us to repeatedly apply the sliding-windows evaluation procedure as described in the next section.

The listening logs provide the timestamps of each listening event, which allows us to apply a time-based repetition scheme as described in Equation 4. Furthermore, the #nowplaying dataset includes the Twitter IDs of the users. We used this information to retrieve the friends of the users on Twitter and explore the effect of our proposed social score on this dataset. Table 2 summarizes the statistics of the used datasets. All datasets used in the experiments except the non-public one from 8tracks are available online[4].

3.2 Metrics and Measurement Method

We use the accuracy measurement protocol that was also used in [6, 7, 15]. The data is split into training and test sets and we then hide the last track of each playlist or listening session in the test set. The goal is to predict this last hidden track. We generate recommendation lists of length 100 and a "hit" is counted when the hidden track was in the top-100 list. Therefore, the hit rate is the fraction of playlists in the

[2]Twitter friends are the users that the specified user is following on Twitter.

[3]See http://dbis-nowplaying.uibk.ac.at for more details.

[4]http://ls13-www.cs.tu-dortmund.de/homepage/ifup2016/datasets.zip

Table 2: Statistics of the used datasets. Note that for the listening logs datasets, "Sessions" refer to the listening sessions and for the playlists datasets to the playlists.

Measure	#NP	30M	Last.fm	AotM	8tracks
Sessions	9,288	9,000	2,762	1,040	6,714
Users	1,032	1,000	426	142	996
Tracks	76,652	123,315	17,815	11,413	39,875
Artists	15,638	26,122	3,209	2,770	9,122
Avg. S/U	9.00	9.00	6.48	7.32	6.74
Avg. T/S	23.89	25.34	11.72	16.98	12.97
Avg. A/S	12.56	9.76	4.55	12.76	12.06

test set for which the hidden track was found. Besides the hit rate we also report the Mean Reciprocal Rank (MRR), which takes the position of the hit into account. Note that in the literature on the generation of playlists and virtually endless radio stations top-n lists of this and even much larger sizes are common [6, 7, 15].

We furthermore assess the *diversity* and *coherence* of the resulting recommendations based on the artists and, where applicable, based on the tags of the tracks. As done in [17], we use the inverse Intra-List-Similarity (ILS) [31] to quantify the *diversity* level. The overlap of artists (tags) in the history and the next-track recommendations is used as a proxy to assess the *coherence* level.

For the playlist experiments, we report the results of a four-fold cross-validation procedure. For the time-ordered listening logs, we use a four-fold sliding-window validation, in which the newest session of each user is used for testing and the 5 previous sessions for training ($windowsize = 5$). After each iteration, the window is moved one session toward the previous sessions ($stepsize = 1$).

3.3 Evaluation Results

We will first focus on the effects of the individual personalization strategies before we report selected results of combining different scores. We use a significance level of $p = 0.05$ for all statistical tests (Student's t-tests) throughout the section. Results that are statistically significantly different from the baseline are printed in bold face along with an up/down arrow which indicates whether the respective value is higher/lower than the baseline value.

3.3.1 Track Repetition

Reusing the same track in more than one shared playlist is comparably uncommon. We therefore focus on the value of repeated recommendations of known tracks based on the listening logs, for which we also have timestamp information. Table 3 shows the results for the different repetition strategies (R1 - R4) proposed in Section 2.3.1.

To highlight the effect of making repeated recommendations, Table 3 shows the evaluation results for those sessions, for which we assumed that the user is mainly listening to already known tracks, i.e., when more than 50% of the recent tracks are repetitions. On average, this was the case for about 22% of the sessions in the #nowplaying dataset and about 15% of the sessions in the 30Music dataset.

The results clearly show that if our heuristic assumes that the user's listening mode is *repetition* (and not *exploration*) and we then correspondingly increase the scores of already known tracks, the accuracy of the recommendations can be significantly enhanced. All the proposed strategies for selecting the candidate tracks for repetition led to an increase of the accuracy measures compared to the baseline of up to 15%. When applying the measurement to all sessions (regardless if the user is in repetition or exploration mode) the effects are less pronounced but we still observe a statistically significant improvement over the baseline. For the #nowplaying dataset, applying the hybrid strategy to all sessions increases the hit rate from 21% to 27%, and for the 30Music dataset from 17% to 21%. Overall, the hybrid method (R4) that combines both the time aspect as well as artist information led to the best accuracy results for both datasets[5]. More elaborate schemes, e.g., to detect the user's listening mode, are of course possible, but in the context of this work we are more interested in the general usefulness of including repetitions in the recommendations.

In regard to diversity (inverse ILS) and coherence (overlap), repeating *popular* tracks from the user's long-term history (R1) leads to more diverse recommendation lists in terms of artists. In contrast, the *artist-based* strategy (R3) for finding the candidate tracks by design decreases the artist diversity and increases the coherence of the recommendations with the listening history. Even though the popularity-based strategy (R1) seems to be effective in terms of accuracy as well, it can be risky in a sense that recommended next tracks have little to do with the last few tracks, which might lead to a decreased quality perception.

3.3.2 Favorite Artists

Table 4 shows the accuracy (upper part) and diversity and coherence results (lower part) for the personalization strategy which assigns higher scores to tracks by the favorite artists of a user. Our focus is again on the listening log datasets, but we also report the results for the playlist datasets to analyze if the proposed techniques work for this problem setting as well.

The obtained results show that including long-term artist preferences is beneficial in terms of the hit rate in all cases except for the AotM dataset, where we only observed a small but not significant improvement. This phenomenon can be explained by the nature of the AotM platfom, where artist "reuse" by users is very infrequent.

Again by design, the artist diversity (inverse ILS) decreases when the favorite artists are played more often. The favorite-artists scheme also leads to a stronger coherence (overlap) of the next tracks with the most recently listened to tracks. Note that the desirable level of diversity and coherence of the recommendations depends on the domain and the goals that should be achieved.

3.3.3 Topic Similarity

Table 5 shows the results when we bias the recommendations towards tracks that have a "topic" that often appeared in the user's long-term history. Again, this personalization approach based on social tags can help to measurably increase the recommendation accuracy.

Since we used social tags as the personalization source in

[5]For the hybridization, we systematically tested different weights and selected the best ones according to accuracy. For #nowplaying, the ratio of R2 to R3 is 3:7 and for 30Music this ratio is 1:1.

Table 3: Results for repetition schemes – accuracy, diversity, and coherence. The applied strategies are R1 = popularity-based, R2 = time-based, R3 = artist-based and R4 = a weighted combination of time-based and artist-based strategies.

	Accuracy (hit rate @ 100)					Accuracy (MRR @ 100)				
	kNN300	+R1	+R2	+R3	+R4	kNN300	+R1	+R2	+R3	+R4
#NP	.493	**.531↑**	**.607↑**	**.643↑**	**.662↑**	.135	.125	.132	.135	**.151↑**
30Music	.486	**.521↑**	**.662↑**	**.672↑**	**.686↑**	.200	.206	**.218↑**	**.237↑**	**.233↑**

	Artist diversity (inv. ILS)					Artist coherence (overlap)				
#NP	.777	**.795↑**	.786	**.689↓**	**.712↓**	.238	**.241↑**	**.319↑**	**.603↑**	**.509↑**
30Music	.744	**.764↑**	**.694↓**	**.641↓**	**.653↓**	.205	**.199↓**	**.254↑**	**.351↑**	**.319↑**

Table 4: Results for a favorite-artist scheme – accuracy, artist diversity, and coherence

	kNN300 vs. kNN300+*favA*			
	hit rate @ 100		MRR @ 100	
#NP	.213	**.290↑**	.051	.049
30Music	.171	**.261↑**	.055	**.065↑**
Last.fm	.273	**.312↑**	.087	**.077↓**
AotM	.100	.118	.039	.042
8tracks	.090	**.094↑**	.018	.018

	inv. ILS		overlap	
#NP	.807	**.707↓**	.155	**.349↑**
30Music	.795	**.634↓**	.121	**.292↑**
Last.fm	.536	**.381↓**	.321	**.476↑**
AotM	.784	.775	.118	.117
8tracks	.981	**.976↓**	.049	**.077↑**

Table 5: Results for topic similarity – accuracy, tag diversity, and coherence

	kNN300 vs. kNN300+*topicSim*			
	hit rate @ 100		MRR @ 100	
Last.fm	.273	**.302↑**	.087	.094
AotM	.100	**.136↑**	.039	**.043↑**
8tracks	.090	.108	.018	**.019↑**

	inv. ILS		overlap	
Last.fm	.379	**.256↓**	.391	**.425↑**
AotM	.564	**.452↓**	.429	**.447↑**
8tracks	.611	**.495↓**	.403	**.417↑**

this experiment, we exceptionally report here the *tag diversity* (inverse ILS) and *coherence* (overlap) of the recommendations in Table 5. The tag diversity by design decreases as we are looking for tracks with similar tags. At the same time a better coherence of the next-track recommendations can be achieved with this approach. As there is no tag information available for the listening logs, we could not repeat the measurement for the remaining two datasets. Our conjecture is that the effects might be less pronounced for listening logs, because playlists are often carefully designed around a certain "theme", which is advantageous for a content-based technique.

3.3.4 Extended Track Co-occurrences

In Table 6, we report the results of considering a user's long-term history to find similar listening sessions (neighbors). Considering the users' long-term preferences in this way again leads to an increase in accuracy. Furthermore, due to the richer user profile, the recommended tracks become more diverse in terms of the artists. The exceptions are the Last.fm and 30Music datasets. One explanation might be that many of the shared playlists and the listening logs from Last.fm contain only one or very few artists. Therefore, considering the long-term preferences of the users can further limit the number of artists of the recommended tracks. Note that the 30Music listening logs are also collected from Last.fm. In terms of coherence, the recommendations are often less connected to the current listening context.

3.3.5 Considering Social Friends

In this approach we consider the recommendation of favorite tracks of the social friends as a means for personalization. In the particular measurement reported here, we considered a track as a favorite if a user listened to it at least 5 times.

The results in Table 7 show the positive effect on accuracy of including these tracks in the recommendations for the #nowplaying dataset, which is the only dataset for which information about social friends was available. The obtained effects on the hit rate are small but significant. We still consider the result promising given that for more than 30% of the users no Twitter friends could be identified who also posted tracks.

3.3.6 Combining the Scores

So far, we have reported the effects of applying the different personalization strategies in isolation. Depending on the available data, different scores can be combined (Equation 1). We have run a variety of additional experiments using different weights to analyze the effects of combining the scores. Similar to the individual strategies, we systematically tested different values for the weight of each score – this time in combination with all other scores – and selected the best ones (listed in Table 8) with respect to accuracy. In this section, we discuss the results of these combinations reported in Table 9.

The results for playlist datasets are based on a combination of favorite artists, topic similarity and long-term kNN scores. For the #nowplaying listening log dataset, we used a combination of all scores (except topic similarity) and for the 30Music dataset the combination includes favorite artists,

Table 6: Results for track co-occurrences – accuracy, artist diversity, and coherence

	kNN300 vs. kNN300+*lth*			
	hit rate @ 100		MRR @ 100	
#NP	.213	**.235**↑	.051	.048
30Music	.171	.177	.055	.055
Last.fm	.273	**.280**↑	.087	.086
AotM	.100	.104	.039	.038
8tracks	.090	**.101**↑	.018	.019

	inv. ILS		overlap	
#NP	.807	**.824**↑	.155	**.140**↓
30Music	.795	**.713**↓	.121	.123
Last.fm	.536	**.472**↓	.321	.321
AotM	.784	**.882**↑	.118	**.078**↓
8tracks	.981	.985	.049	.048

Table 7: Results for Favorites-of-Friends – accuracy, artist diversity, and coherence

	kNN300 vs. kNN300+*fof*			
	hit rate @ 100		MRR @ 100	
#NP	.213	**.216**↑	.051	.051

	inv. ILS		overlap	
#NP	.807	**.811**↑	.155	**.152**↓

repetition and long-term kNN scores (see Table 8).

The results indicate that a weighted combination of multiple scores with the baseline method can further increase the accuracy. Specifically, the multi-score combinations outperform every other single-score combination with the baseline in terms of hit rate across all five datasets.

The multi-score method also has effects on the diversity and coherence of the recommendations, which are the combined result of the individual scoring methods. For instance, for the Last.fm dataset, all of the single combinations reduce the artist diversity and as a result, the multi-score combination also generates recommendations that are not only less diverse than the baseline but also less diverse than each of the single combinations. As another example, consider the coherence of the recommendations for the #nowplaying dataset. Two of the single combinations (*kNN + rep* and *kNN + favA*) lead to a higher coherence than the kNN method, whereas the other two combinations (*kNN + lth* and *kNN + fof*) lead to a lower coherence than the kNN method. The multi-score combination, which is a combination of all these four scores with the kNN method, leads to a higher coherence than the baseline method. The explanation for this effect is that the first two scores with weights of $w_{rep} = 2.0$ and $w_{favA} = 1.5$ have a higher influence on the recommendations than the other two scores with lower weights ($w_{kNN_{lth}} = 0.5$ and $w_{fof} = 0.3$). Therefore, this multi-score combination – similar to the first two single combinations – generates more coherent recommendations.

Generally, how to set the diversity and coherence level of the recommendations depends on the goals that should be achieved. For instance, if a user is in *exploration* mode and intends to discover new tracks or artists, a more diverse recommendation list would be desired. Similarly, when a user is creating a playlist with a specific theme, e.g., a workout playlist, she would probably prefer tracks with a coherent tempo. This can be achieved by selecting and weighting the individual personalization components in our multi-score combination approach according to the desired goals.

Table 8: Weights of the baseline scores (w_{base}), the personalization scores (w_{pers}) of the multi-score combinations as well as the weights of α for the extended CAGH version (Equation 6) and β for the topic similarity method (Equation 7) for different datasets. Note that not all methods can be applied for all datasets (indicated by a "–").

	#NP	30Music	Last.fm	AotM	8tracks
w_{base}	0.5	0.7	0.9	0.4	0.7
w_{rep}	2.0	2.0	–	–	–
w_{favA}	1.5	4.0	2.0	2.0	1.0
$w_{topicSim}$	–	–	0.3	1.0	0.3
$w_{kNN_{lth}}$	0.5	0.3	0.1	0.6	0.3
w_{fof}	0.3	–	–	–	–
α	0.9	0.9	0.9	0.4	0.8
β	–	–	0.9	0.6	0.7

Overall, we view our results to be promising as we could show the benefits of incorporating various forms of personalization signals into the recommendation process even though the available data for each user is partially very sparse and, in the case of the social tags, relatively noisy.

4. RELATED WORK

Next-track recommendation can be considered a specific type of music recommendation where the goal is to generate a playlist as a continuation of a recent listening experience. A variety of playlist generation strategies have been proposed over the last decade (see [7]). They base their recommendations, e.g., on content similarity [21], collaborative filtering [13, 15], Markov models [12], discrete optimization [3], and hybrid techniques [23]. While personalization is considered a key feature in general recommender systems research and also for various types of music recommendation [18], most existing approaches to playlist generation from the literature only rely on the users' most recent listening histories and not on their long-term preferences.

A few exceptions exist. In [29], the value of considering repetition patterns is explored to develop a novelty model for generating personalized music recommendations. In [10], a method is proposed that leverages the online friendship relations and other signals for recommending. In contrast to these works that focus on one single source for personalization, we propose an extension of the general faceted weighted track scoring approach from [17]. Instead of musical features and track meta-data in the scoring process used in [17], here we rely on a variety of user-specific listening patterns and the user's social network. Adding musical features to the process is however generally possible.

There are other examples of combining music recommendation algorithms in the literature [11, 23, 30]. In the latter, for instance, music tracks are recommended by combining

Table 9: Results when using multiple scores – accuracy, artist diversity, and coherence. The Scores are *rep* = track repetition, *favA* = favorite artists, *topicS* = topic similarity, kNN_{lth} = extended track co-occurrence, *fof* = social friends, and *all* = combination of all applicable scores. Note that not all methods can be applied for all datasets (indicated by a "–").

	Accuracy (hit rate @ 100)							Accuracy (MRR @ 100)						
	kNN	+ *rep*	+ *favA*	+ *topicS*	+ kNN_{lth}	+ *fof*	+ *all*	kNN	+ *rep*	+ *favA*	+ *topicS*	+ kNN_{lth}	+ *fof*	+ *all*
#NP	.213	**.268↑**	**.290↑**	–	**.235↑**	**.216↑**	**.323↑**	.051	.052	.049	–	.048	.051	.054
30Music	.171	**.212↑**	**.261↑**	–	.177	–	**.269↑**	.055	**.061↑**	**.065↑**	–	.055	–	**.053↓**
Last.fm	.273	–	**.312↑**	**.302↑**	**.280↑**	–	**.315↑**	.087	–	**.077↓**	.094	.086	–	.076
AotM	.100	–	.118	**.136↑**	.104	–	**.143↑**	.039	–	.042	**.043↑**	.038	–	**.022↓**
8tracks	.090	–	**.094↑**	**.108↑**	**.101↑**	–	**.115↑**	.018	–	.018	**.019↑**	.019	–	.014

	Artist diversity (inv. ILS)							Artist coherence (overlap)						
#NP	.807	.810	**.707↓**	–	**.824↑**	**.811↑**	**.756↓**	.155	**.255↑**	**.320↑**	–	**.140↓**	**.152↓**	**.397↑**
30Music	.795	**.716↓**	**.634↓**	–	**.713↓**	–	**.599↓**	.121	**.162↑**	**.292↑**	–	.123	-	**.356↑**
Last.fm	.536	–	**.381↓**	**.464↓**	**.472↓**	–	**.363↓**	.321	–	**.476↑**	**.369↑**	.321	–	**.485↑**
AotM	.784	–	.775	**.764↓**	**.882↑**	–	.823	.118	–	.117	**.136↑**	**.078↓**	–	**.082↓**
8tracks	.981	–	**.976↓**	**.974↓**	**.985↑**	–	**.912↓**	.049	–	**.077↑**	**.065↑**	.048	–	**.206↑**

a collaborative-filtering, an artist-based and a clustering-based algorithm through rank interpolation. Similar to our scoring scheme, their goal is to increase the recommendation accuracy as well as other factors like novelty, diversity, and serendipity. However, their main focus lies on creating more serendipitous recommendations and the technique is not based on exchangeable baseline methods.

In our work, we utilize *seed tracks* for estimating the desired characteristics of the next-track recommendations. Our seed tracks stem partially from hand-crafted playlists shared by users on different music platforms as in [5] and partially from listening logs collected from Internet radio stations and online social media as in [16]. Other approaches exist in which only one single seed track [22] or the start and the end tracks are specified [14]. Relying only on one single track is in principle possible with our approach but would not allow us to reliably guess the underlying theme of the current playlists or listening sessions. As we are interested in (endless) next-track recommendations, the provision of a desired end track is not reasonable for us.

In order to assess whether a track fulfills the desired characteristics of a listening history, various types of information can be used, e.g., musical features extracted from the audio signals [26], meta-data features about artist, genre, or the release year [1], as well as popularity or usage data [24], and other social web data like tags [15]. In this paper, we used the user-specific signals extracted from the usage data as input source.

To evaluate the quality of our proposed algorithms, we compare the next-track recommendations of each algorithm with the respective listening history, i.e., a hand-crafted playlist or a listening session. The assumption is that the desired quality level of certain criteria such as diversity can be determined from these listening logs. To compare the accuracy of different algorithms in Section 2.3, we use the *hit rate* measure and the configuration proposed in [15], i.e., we hide the last track in each test listening history. For measuring the *diversity* and *coherence* of recommendations, as done in [17], we use the inverse ILS and the overlap of artists (or tags) in the history and the recommendations respectively. Generally, other evaluation approaches are possible as well as discussed in [7], among them *user studies* [4] and *log analysis* [8].

5. CONCLUSIONS

In this work, we have proposed and evaluated a number of strategies to leverage a variety of signals for personalizing next-track music recommendations based on the users' long-term preferences. One general insight of our work is that even though the selection of the immediate next tracks should be primarily governed by the most recently played tracks, personalization based on user-specific signals extracted from the long-term preferences can further improve the quality of the recommendations.

Technically, in order to extract potentially relevant user-specific signals, we analyzed a larger number of "hand-crafted" and publicly shared playlists as well as long-term listening logs. Specifically, we explored the value of track repetition, favorite artists, topic similarity, track co-occurrence, and social friends as possible personalization signals. These signals were then incorporated in a multi-faceted scoring scheme that combines a baseline algorithm that focuses on the short-term interests with one or more long-term personalization components.

Despite the noisiness, sparseness and partial incompleteness of the data, the results show that our proposed strategies not only help to further increase the recommendation accuracy but can also help to influence other quality dimensions like diversity and coherence depending on the desired goals. Generally, deciding on (a) the types of auxiliary information that should be incorporated in the recommendation process and (b) the relative importance of short-term and long-term preferences has to be done with the particularities of the given application domain in mind.

Conducting user studies to evaluate our approaches with respect to the perceived quality of the next-track recommendations and user satisfaction is part of our ongoing work.

6. REFERENCES

[1] N. Aizenberg, Y. Koren, and O. Somekh. Build Your Own Music Recommender by Modeling Internet Radio Streams. In *Proc. WWW*, pages 1–10, 2012.

[2] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii. The Dynamics of Repeat Consumption. In *Proc. WWW*, pages 419–430, 2014.

[3] J.-J. Aucouturier and F. Pachet. Scaling Up Music Playlist Generation. In *Proc. ICME*, volume 1, pages 105–108, 2002.

[4] L. Barrington, R. Oda, and G. R. G. Lanckriet. Smarter than Genius? Human Evaluation of Music Recommender Systems. In *Proc. ISMIR*, pages 357–362, 2009.

[5] D. Baur, S. Boring, and A. Butz. Rush: Repeated Recommendations on Mobile Devices. In *Proc. IUI*, pages 91–100, 2010.

[6] G. Bonnin and D. Jannach. Evaluating the Quality of Playlists Based on Hand-Crafted Samples. In *Proc. ISMIR*, pages 263–268, 2013.

[7] G. Bonnin and D. Jannach. Automated Generation of Music Playlists: Survey and Experiments. *ACM Comput. Surv.*, 47(2):26:1–26:35, 2014.

[8] K. Bosteels, E. Pampalk, and E. E. Kerre. Evaluating and Analysing Dynamic Playlist Generation Heuristics Using Radio Logs and Fuzzy Set Theory. In *Proc. ISMIR*, pages 351–356, 2009.

[9] M. Braunhofer, M. Kaminskas, and F. Ricci. Location-aware music recommendation. *International Journal of Multimedia Information Retrieval*, 2(1):31–44, 2013.

[10] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He. Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content. In *Proc. MM*, pages 391–400, 2010.

[11] Z. Chedrawy and S. S. R. Abidi. A Web Recommender System for Recommending, Predicting and Personalizing Music Playlists. In *Proc. WISE*, pages 335–342, 2009.

[12] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist Prediction via Metric Embedding. In *Proc. KDD*, pages 714–722, 2012.

[13] C. Desrosiers and G. Karypis. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *RSs Handbook*, pages 107–144. Springer US, 2011.

[14] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer. Playlist Generation Using Start and End Songs. In *Proc. ISMIR*, pages 173–178, 2008.

[15] N. Hariri, B. Mobasher, and R. Burke. Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns. In *Proc. RecSys*, pages 131–138, 2012.

[16] Y. Hu and M. Ogihara. NextOne Player: A Music Recommendation System Based on User Behavior. In *Proc. ISMIR*, pages 103–108, 2011.

[17] D. Jannach, L. Lerche, and I. Kamehkhosh. Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks. In *Proc. RecSys*, pages 187–194, 2015.

[18] M. Kaminskas and F. Ricci. Contextual Music Information Retrieval and Recommendation: State of the Art and Challenges. *Computer Science Review*, 6(2-3):89–119, 2012.

[19] K. Kapoor, V. Kumar, L. Terveen, J. A. Konstan, and P. Schrater. "I like to explore sometimes": Adapting to Dynamic User Novelty Preferences. In *Proc. RecSys*, pages 19–26, 2015.

[20] P. B. Lamere. I've Got 10 Million Songs in My Pocket: Now What? In *Proc. RecSys*, pages 207–208, 2012.

[21] A. Lehtiniemi and J. Seppänen. Evaluation of Automatic Mobile Playlist Generator. In *Proc. MC*, pages 452–459, 2007.

[22] B. Logan. Content-Based Playlist Generation: Exploratory Experiments. In *Proc. ISMIR*, pages 295–296, 2002.

[23] B. McFee and G. R. G. Lanckriet. The Natural Language of Playlists. In *Proc. ISMIR*, pages 537–542, 2011.

[24] B. McFee and G. R. G. Lanckriet. Hypergraph Models of Playlist Dialects. In *Proc. ISMIR*, pages 343–348, 2012.

[25] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull. Learning to Embed Songs and Tags for Playlist Prediction. In *Proc. ISMIR*, pages 349–354, 2012.

[26] T. Pohle, E. Pampalk, and G. Widmer. Generating Similarity-Based Playlists using Traveling Salesman Algorithms. In *Proc. DAFx*, pages 220–225, 2005.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proc. UAI*, pages 452–461, 2009.

[28] R. Turrin, M. Quadrana, A. Condorelli, R. Pagano, and P. Cremonesi. 30Music Listening and Playlists Dataset. 2015.

[29] X. Wang, Y. Wang, D. Hsu, and Y. Wang. Exploration in Interactive Personalized Music Recommendation: A Reinforcement Learning Approach. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(1):7:1–7:22, Sept. 2014.

[30] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor. Auralist: Introducing Serendipity into Music Recommendation. In *Proc. WSDM*, pages 13–22, 2012.

[31] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *Proc. WWW*, pages 22–32, 2005.