Analyzing the Characteristics of Shared Playlists for Music Recommendation

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

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

Geoffray Bonnin
TU Dortmund, Germany
geoffray.bonnin@tu-dortmund.de

ABSTRACT

The automated generation of music playlists – as supported by modern music services like last.fm or Spotify – represents a special form of music recommendation. When designing a "playlisting" algorithm, the question arises which kind of quality criteria the generated playlists should fulfill and if there are certain characteristics like homogeneity, diversity or freshness that make the playlists generally more enjoyable for the listeners. In our work, we aim to obtain a better understanding of such desired playlist characteristics in order to be able to design better algorithms in the future. The research approach chosen in this work is to analyze several thousand playlists that were created and shared by users on music platforms based on musical and meta-data features.

Our first results for example reveal that factors like popularity, freshness and diversity play a certain role for users when they create playlists manually. Comparing such user-generated playlists with automatically created ones moreover shows that today's online playlisting services sometimes generate playlists which are quite different from user-created ones. Finally, we compare the user-created playlists with playlists generated with a nearest-neighbor technique from the research literature and observe even stronger differences. This last observation can be seen as another indication that the accuracy-based quality measures from the literature are probably not sufficient to assess the effectiveness of playlisting algorithms.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing

General Terms

Playlist generation, Music recommendation

Keywords

Music, playlist, analysis, algorithm, evaluation

Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014), collocated with ACM RecSys 2014, 10/06/2014, Foster City, CA, USA.

1. INTRODUCTION

The automated creation of playlists or personalized radio stations is a typical feature of today's online music platforms and music streaming services. In principle, standard recommendation algorithms based on collaborative filtering or content-based techniques can be applied to generate a ranked list of musical tracks given some user preferences or past listening history. For several reasons, the generation of playlists however represents a very specific music recommendation problem. Personal playlists are, for example, often created with a certain goal or usage context (e.g., sports, relaxation, driving) in mind. Furthermore, in contrast to relevance-ranked recommendation lists used in other domains, playlists typically obey some homogeneity and coherence criteria, i.e., there are quality characteristics that are related to the transitions between the tracks or to the playlist as a whole.

In the research literature, a number of approaches for the automation of the playlist generation process have been proposed, see, e.g., [2, 6, 8, 10, 11] or the recent survey in [3]. Some of them for example take a seed song or artist as an input and look for similar tracks; others try to find track co-occurrence patterns in existing playlists. In some approaches, playlist generation is considered as an optimization problem. Independent of the chosen technique, a common problem when designing new playlisting algorithms is to assess whether or not the generated playlists will be positively perceived by the listeners. User studies and online experiments are unfortunately particularly costly in the music domain. Researchers therefore often use offline experimental designs and for example use existing playlists shared by users on music platforms as a basis for their evaluations. The assumption is that these "hand-crafted" playlists are of good quality; typical measures used in the literature include the *Recall* [8] or the *Average Log-Likelihood (ALL)* [11]. Unfortunately, both measures have their limitations, see also [2]. The *Recall* measure for example tells us how good an algorithm is at predicting the tracks selected by the users, but does not explicitly capture specific aspects such as the homogeneity or the smoothness of track transitions.

To design better and more comprehensive quality measures, we however first have to answer the question of what users consider to be desirable characteristics of playlists or what the driving principles are when users create playlists. In the literature, a few works have studied this aspect using different approaches, e.g., user studies [1, 7] or analyzing forum posts [5]. The work presented in this paper continues these lines of research. Our research approach is however different from previous works as we aim to identify patterns in a larger set of manually created playlists that were shared by users of three different online music platforms. To be able to take a variety of potential driving factors into account in our analysis, we have furthermore collected various types of meta-data and musical features of the playlist tracks from public music databases.

Overall, with our analyses we hope to obtain insights on the principles which an automated playlist generation system should observe to end up with better-received or more "natural" playlists. To test if current music services and a nearest-neighbor algorithm from the literature generate playlists that observe the identified patterns and make similar choices as real users, we conducted an experiment in which we analyzed commonalities and differences between automatically generated and user-provided playlists.

Before reporting the details of our first analyses, we will first discuss previous works in the next section.

2. PREVIOUS WORKS

In [14], Slaney and White addressed the question if users have a tendency to create very homogeneous or rather diverse playlists. As a basis for determining the diversity they relied on an objective measure based on genre information about the tracks. Each track was considered as a point in the genre space and the diversity was then determined by calculating the volume of an ellipsoid enclosing the tracks of the playlist. An analysis of 887 user-created playlists indicated that diversity can be considered to be a driving factor as users typically create playlists covering several genres.

Sarroff and Casey more recently [13] focused on track transitions in album playlists and made an analysis to determine if there are certain musical characteristics that are particularly important. One of the results of their investigation was that fade durations and the mean timbre of the beginnings and endings of consecutive tracks seem to have a strong influence on the ordering of the tracks.

Generally, our work is similar to [14] and [13] in that we rely on user-created ("hand-crafted") playlists and look at meta-data and musical features of the tracks to identify potentially important patterns. The aspects we cover in this paper were however not covered in their work and our analysis is based on larger datasets.

Cunningham et al., [5], in contrast, relied on another form of track-related information and looked at the user posts in the forum of the Art of the Mix web site. According to their analysis, the typical principles for setting up the playlists mentioned by the creators were related to the artist, genre, style, event or activity but also the intended purpose, context or mood. Some users also talked about the smoothness of track transitions and how many tracks of one single artist should be included in playlists. Placing the most "important" track at the end of a playlist was another strategy mentioned by some of the playlist creators.

A different form of identifying playlist creation principles is to conduct laboratory studies with users. The study reported in [7] for example involved 52 subjects and indicated that the first and the last tracks can play an important role for the quality of a playlist. In another study, Andric and Haus [1] concluded that the ordering of tracks is not important when the playlist mainly contains tracks which the users like in general.

Reynolds et al. [12] made an online survey that revealed that the context and environment like the location activity or the weather can have an influence both on the listeners' mood and on the track selection behavior of playlist creators. Finally, the study presented in [9] again confirmed the importance of artists, genres and mood in the playlist creation process.

In this discussion, we have focused on previous attempts to understand how users create playlists and what their characteristics are. Playlist generation algorithms however do not necessarily have to rely on such knowledge. Instead, one can follow a statistical approach and only look at co-occurrences and transitions of tracks in existing playlists and use these patterns when creating new playlists, see e.g., [2] or [4]. This way, the quality factors respected by human playlist creators are implicitly taken into account. Such approaches, however, cannot be directly applied for many types of playlist generation settings, e.g., for creating "thematic" playlists (e.g., Christmas Songs) or for creating playlists that only contain tracks that have certain musical features. Pure statistical methods are not aware of these characteristics and the danger exists that tracks are included that do not match the purpose of the list and thus lead to a limited overall quality.

3. CHARACTERISTICS OF PLAYLISTS

The ultimate goal of our research is to analyze the structure and characteristics of playlists in order to better understand the principles used by the users to create them. This section is a first step toward this goal.

3.1 Data sources

As a basis for the first analyses that we report in this paper, we used two types of playlist data.

3.1.1 Hand-crafted playlists

We used samples of hand-crafted playlists from three different sources. One set of playlists was retrieved via the public API of last.fm[1], one was taken from the Art of the Mix (AotM) website[2], and a third one was provided to us by 8tracks[3]. To enhance the data quality, we corrected artist misspellings using the API of last.fm.

Overall, we analyzed over 10,000 playlists containing about 108,000 different tracks of about 40,000 different artists. As a first attempt toward our goal, we retrieved the features listed in Table 1 using the public API of last.fm and The Echo Nest (tEN), and the MusicBrainz database.

Some dataset characteristics are shown in Table 2. The "usage count" statistics express how often tracks and artists appeared overall in the playlists. When selecting the playlists, we made sure that they do not simply contain album listings. The datasets are partially quite different, e.g., with respect to the average playlist lengths. The 8tracks dataset furthermore has the particularity that users are not allowed to include more than two tracks of one artist, in case they want to share their playlist with others.

Figure 1 shows the distributions of playlist lengths. As can be seen, the distributions are quite different across the datasets. On 8tracks, a playlist generally has to comprise

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

[2] http://www.artofthemix.org

[3] http://8tracks.com

Source	Information	Description
last.fm	Tags	Top tags assigned by users to the track.
last.fm	Playcounts	Total number of times the users played the track.
tEN	Genres	Genres of the artist of the track. Multiple genres can be assigned to a single artist.
tEN	Danceability	Suitability of the track for dancing, based on various information including the beat strength and the stability of the tempo.
tEN	Energy	Intensity released throughout the track, based on various information including the loudness and segment durations.
tEN	Loudness	Overall loudness of the track in decibels (dB).
tEN	Tempo	Speed of the track estimated in beats per minute (BPM).
tEN	Hotttnesss	Current reputation of the track based on its activity on some web sites crawled by the developers.
MB	Release year	Year of release of the corresponding album.

Table 1: Additional retrieved information.

	`lastfm`	`AotM`	`8tracks`
Playlists	1,172	5,043	3,130
Tracks	24,754	61,935	29,732
Artists	9,925	23,029	13,379
Avg. tracks/playlist	26.0	19.7	12.5
Avg. artists/playlist	16.8	17.8	11.5
Avg. genres/playlist	2.7	3.5	3.4
Avg. tags/playlist	473.4	418.7	297.4
Avg. track usage count	1.2	1.6	1.3
Avg. artist usage count	3.0	4.3	2.9

Table 2: Some basic statistics of the datasets.

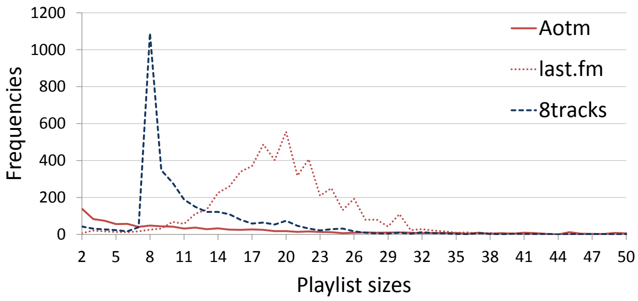

Figure 1: Distribution of playlists sizes.

at least 8 tracks. The lengths of the last.fm playlists seem to follow a normal distribution with a maximum frequency value at around 20 tracks. Finally, the sizes of the AotM playlists are much more equally distributed.

3.1.2 *Generated playlists*

To assess if the playlists generated by today's online services are similar to those created by users, we used the public API of The Echo Nest. We chose this service because it uses a very large database and allows the generation of playlists from several seed tracks, as opposed to, for instance, iTunes Genius or last.fm radios. We split the existing hand-crafted playlists in half, provided the first half of the list as seed tracks to the music service and then analyzed the characteristics of the playlist returned by The Echo Nest and compared them to the patterns that we found in hand-crafted playlists. Instead of observing whether a playlister generates playlists that are generally similar to playlists created by hand, our goal here is to break down their different characteristics and observe on what specific dimensions they differ. Notice that using the second half as seed would not be appropriate as the order of the tracks may be important.

We also draw our attention to the ability of the algorithms of the literature to reproduce the characteristics of hand-crafted playlists. According to some recent research, one of the most competitive approaches in terms of recall is the simple k-nearest-neighbors (kNN) method [2, 8]. More precisely, given some seed tracks, the algorithm extracts the k most similar playlists based on the number of shared items and recommends the tracks of these playlists. This algorithm does not require a training step and scans the entire set of available playlists for each recommendation.

3.2 Detailed observations

In the following sections, we will look at general distributions of different track characteristics.

3.2.1 *Popularity of tracks*

The goal of the first analysis here is to determine if users tend to position tracks in playlists depending on their popularity. In our analysis, we measure the popularity in terms of play counts. Play counts were taken from last.fm, because this is one of the most popular services and the corresponding values can be considered indicative for a larger user group.

For the measurement, we split the playlists into two parts of equal size and then determined the average play counts on last.fm for the tracks for each half. To measure to which extent the user community favors certain tracks in the playlists, we calculated the Gini index, a standard measure of inequality[4]. Table 3 shows the results. In the last column, we report the statistics for the tracks returned by The Echo Nest (tEN) and kNN playlisters[5]. We provided the first half of the hand-crafted playlists as seed tracks and the playlisters had to select the same number of tracks as the number of remaining tracks.

The results show that users actually tend to place more popular items in the first part of the list in all datasets, when play counts are considered. The Echo Nest playlister does not seem to take that form of popularity into account

[4] We organized the average play counts in 100 bins.

[5] We determined 10 as the best neighborhood size for our data sets based on the recall value, see Section 4.

Play counts	1st half	2nd half	tEN
last.fm	1,007k	893k	629k
AotM	671k	638k	606k
8tracks	953k	897k	659k

Gini index	1st half	2nd half	tEN
last.fm	0.06	0.04	0.04
AotM	0.20	0.18	0.22
8tracks	0.09	0.09	0.08

Play counts	1st half	2nd half	kNN
last.fm	1,110k	943k	1,499k
AotM	645k	617k	867k
8tracks	1,008k	984k	1,140k

Gini index	1st half	2nd half	kNN
last.fm	0.12	0.09	0.33
AotM	0.26	0.23	0.43
8tracks	0.15	0.12	0.28

Table 3: Popularity of tracks in playlists (last.fm play counts) and concentration bias (Gini coefficient).

and recommends on average less popular tracks. These differences are statistically significant according to a Student's t-test ($p < 10^{-5}$ for The Echo Nest playlister and $p < 10^{-7}$ for the kNN playlister). This behavior indicates also that The Echo Nest is successfully replicating the fact that the second halves of playlists are supposed to be less popular than the first half.

The Gini index reveals that there is a slightly stronger concentration on some tracks in the first half for two of three datasets and the diversity slightly increases in the second part. The absolute numbers cannot be directly compared across datasets, but for the AotM dataset the concentration is generally much higher, which is also indicated by the higher "track reuse" in Table 2. Interestingly, The Echo Nest playlister quite nicely reproduces the behavior of real users with respect to the diversity of popularity.

In the lower part of Table 3, we show the results for the kNN method. Note that these statistics are based on a different sample of the playlists than the previous measurement. The reason is that both The Echo Nest and the kNN playlisters cannot produce playlists for all of the first halves provided as seed tracks. We therefore considered only playlists, for which the corresponding algorithm could produce a playlist.

Unlike the playlister of The Echo Nest, the kNN method has a strong trend to recommend mostly very popular items. This can be caused by the fact that the kNN method by design recommends tracks that are often found in similar playlists. Moreover, based on the lower half of Table 3, the popularity correlates strongly with the seed track popularity. As a result, the kNN shows a potentially undesirable trend to reinforce already popular items to everyone. At the same time, it concentrates the track selection on a comparable small number of tracks as indicated by the very high value for the Gini coefficient.

3.2.2 *The role of freshness*

Next, we analyzed if there is a tendency of users to create playlists that mainly contain recently released tracks. As a measure, we compared the creation year of each playlist with the average release year of its tracks. We limit our analysis to the last.fm and 8tracks datasets because we only could acquire creation dates for these two.

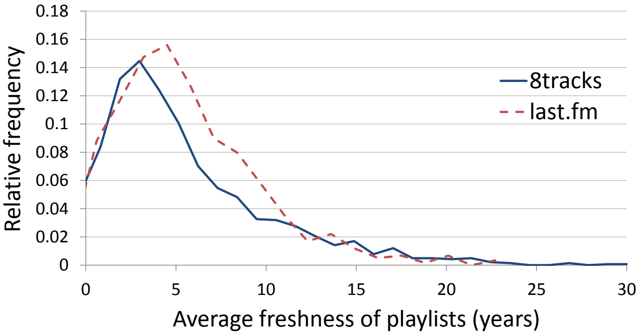

Figure 2: Distribution of average freshness of playlists (comparing playlist creation date and track release date).

Figure 2 shows the statistics for both datasets. We organized the data points in bins (x-axis), where each bin represents an average-freshness level, and then counted how many playlists fall into these levels. The relative frequencies are shown on the y-axis. The result are very similar for both datasets, with a slight tendency to include older tracks for last.fm. On both datasets, more than half of the playlists contain tracks that were released on average in the last 5 years, the most frequent average age being between 4 and 5 years for last.fm and between 3 and 4 years for 8tracks. Similarly, on both datasets, more than 75% of the playlists contain tracks that were released on average in the last 8 years.

We also analyzed the standard deviation of the resulting freshness values and observed that more than half of the playlists have a standard deviation of less than 4 (years), while more than 75% have a standard deviation of less than 7 (years) on both datasets. Overall, this suggests that playlists made by users are often homogeneous with regard to the release date.

Computing the freshness for the generated playlists would require to configure the playlisters in such a way that they select only tracks that were not released after the playlists' creation years. Unfortunately, The Echo Nest does not allow such a configuration. Moreover, for the kNN approach, the playlists that are more recent would have to be ignored, which would lead to a too small sample size and not very reliable results anymore.

3.2.3 *Homogeneity and diversity*

Homogeneity and diversity can be determined in a variety of ways. In the following, we will use simple measures based on artist and genre counts. The genres correspond to the genres of the artists of the tracks retrieved from The Echo Nest. Basic figures for artist and genre diversity are already given in Table 2. On AotM, for example, having several tracks of an artist in a playlist is not very common[6]. On last.fm, we in contrast very often see two or more tracks of

[6] On 8tracks, artist repetitions are limited due to license constraints

one artist in a playlist. A similar, very rough estimate can be made for the genre diversity. If we ordered the tracks of a playlist by genre, we would encounter a different genre on last.fm only after having listened to about 10 tracks. On AotM and 8tracks, in contrast, playlists on average cover more genres.

Table 4 shows the diversities of the first and second halves of the hand-crafted playlists, and for the automatic selections using the first halves as seeds. As a measure of diversity, we simply counted the number of artists and genres and divided by the corresponding number of tracks. The values in Table 4 correspond the averages of these diversity measures.

		1st half	2nd half	tEN
last.fm	artists	0.74	0.76	0.93
	genres	2.26	2.30	2.12
AotM	artists	0.93	0.93	0.94
	genres	3.26	3.22	2.41
8tracks	artists	0.97	0.98	0.99
	genres	3.74	3.85	2.89

		1st half	2nd half	kNN
last.fm	artists	0.74	0.76	0.87
	genres	2.32	2.26	3.11
AotM	artists	0.94	0.94	0.91
	genres	3.27	3.21	3.70
8tracks	artists	0.97	0.98	0.93
	genres	3.94	3.92	4.06

Table 4: Diversity of playlists (Number of artists and genres divided by the corresponding number of tracks).

Regarding the diversity of the hand-crafted playlists, the tables show that users tend to keep a same level of artist and genre diversity throughout the playlists. We can also notice that the playlists of last.fm are much more homogeneous. The diversity values of the automatic selections reveal several things. First, The Echo Nest playlister tends to always maximize the artist diversity independently of the diversity of the seeds; on the contrary, the kNN playlister lowered the initial artist diversities, except on the last.fm dataset, where it increased them, though less than The Echo Nest playlister. Regarding the genre diversity, we can observe an opposite tendency for both playlisters: The Echo Nest playlister tends to reduce the genre diversity while the kNN playlister tends to increase it. Again, these difference are statistically significant ($p < 0.03$ for The Echo Nest playlister and $p < 0.006$ for the kNN playlister). Overall, the resulting diversities of the both approaches tend to be rather dissimilar to those of the hand-crafted playlists.

3.2.4 *Musical features (The Echo Nest)*

Figure 3 shows the overall relative frequency distribution of the numerical features from The Echo Nest listed in Table 1 for the set of tracks appearing in our playlists on a normalized scale. For the loudness feature, for example, we see that most tracks have values between 40 and 50 on the normalized scale. This would translate into an actual loudness value of -20 to 0 returned by The Echo Nest, given that the range is -100 to 100.

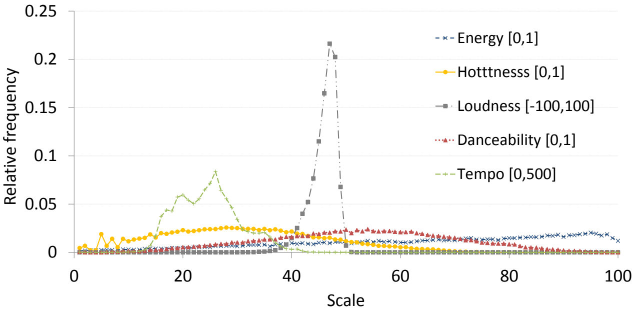

Figure 3: Distribution of The Echo Nest track musical features independently of playlists.

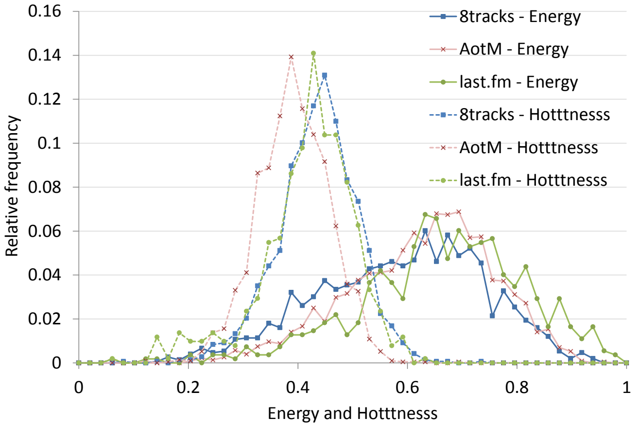

Figure 4: Distribution of mean energy and "hotttnesss" levels in playlists.

To understand if people tend to place tracks with specific feature values into their playlists, we then computed the distribution of the average feature values of each playlist. Figure 4 shows the results of this measurement for the energy and "hotttnesss" features. For all the other features (danceability, loudness and tempo), the distributions were similar to those of Figure 3, which could mean that they are generally not particularly important for the users.

When looking at the energy feature, we see that users tend to include tracks from a comparably narrow energy spectrum with a low average energy level, even though there exist more high-energy tracks in general as shown in Figure 3. A similar phenomenon of concentration on a certain range of values can be observed for the "hotttnesss" feature. As a side aspect, we can observe that the tracks shared on AotM are on average slightly less "hottt" than those of both other platforms[7].

We finally draw our attention to the feature distributions of the generated playlists. Figure 5 as an example shows the distributions of the energy and "hotttnesss" factors for

[7] The results for the "hotttnesss" we report here correspond to the values at the time when we retrieved the data using the API of The Echo Nest, and not to those at the time when the playlists were created. This is not important as we do not look at the distributions independently, but compare them to the distributions in Figure 3.

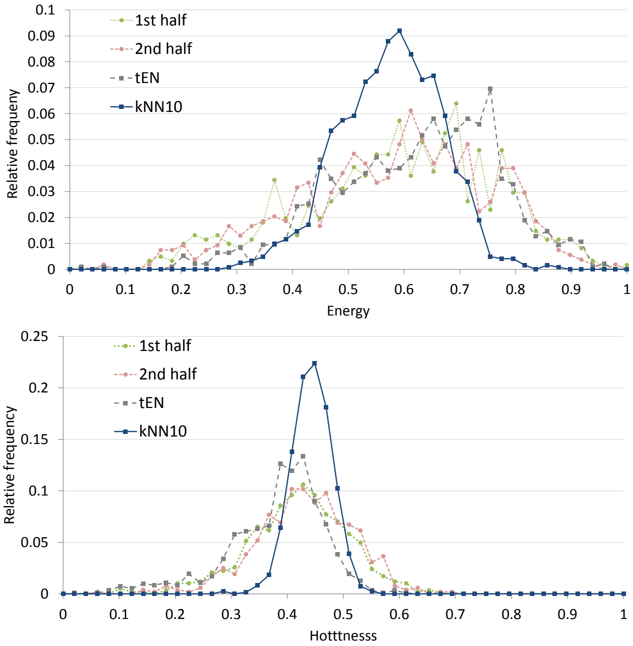

Figure 5: Comparison of the distribution of energy and "hotttnesss" levels for hand-crafted and generated playlists.

the first halves and second halves of the playlists of all three datasets, together with the distributions of the tracks selected by The Echo Nest and kNN playlisters.

The figure shows that The Echo Nest playlister tends to produce a distribution that is quite similar to the distribution of the seed tracks. The kNN playlister, in contrast, tends to concentrate the distributions toward the maximum values of the distributions of the seeds. We could observe this phenomenon of concentration for all the features on all three datasets, except for the danceability on the AotM dataset.

3.2.5 *Transitions and Coherence*

We now focus on the importance of transitions between the tracks, and define the *coherence* of a playlist as the average similarity between its consecutive tracks. Such similarities can be computed according to various criteria. We used the binary cosine similarity of the genres and artists[8], and the Euclidean linear similarity for the numerical track features of The Echo Nest. Table 5 shows the corresponding results for the first and second halves of the hand-crafted playlists, and for the automatic selections using the first halves as seeds.

We can first see that for all datasets and for all criteria, the second halves of the playlists have a lower coherence than the first halves. If we assume that the coherence is representative of the effort of the users to create good playlists, then the tracks of the second halves seem to be slightly less carefully selected than those of the first halves.

		1st half	2nd half	tEN
last.fm	artists	0.19	0.18	0
	genres	0.43	0.40	0.56
	energy	0.76	0.71	0.77
	hotttnesss	0.81	0.76	0.83
AotM	artists	0.05	0.05	0
	genres	0.24	0.22	0.50
	energy	0.75	0.74	0.75
	hotttnesss	0.83	0.82	0.85
8tracks	artists	0.02	0.01	0
	genres	0.22	0.22	0.52
	energy	0.73	0.71	0.76
	hotttnesss	0.81	0.79	0.85

		1st half	2nd half	kNN
last.fm	artists	0.22	0.21	0.02
	genres	0.44	0.42	0.14
	energy	0.76	0.76	0.75
	hotttnesss	0.83	0.82	0.83
AotM	artists	0.05	0.05	0.03
	genres	0.22	0.21	0.13
	energy	0.75	0.74	0.73
	hotttnesss	0.83	0.82	0.84
8tracks	artists	0.02	0.01	0.03
	genres	0.22	0.22	0.17
	energy	0.74	0.73	0.74
	hotttnesss	0.82	0.80	0.84

Table 5: Coherence of first, second and generated halves.

Another interesting phenomenon is the high artist coherence values on the last.fm dataset. These values indicate that last.fm users have a surprisingly strong tendency to group tracks from the same artist together, which was not successfully reproduced by the two playlisters. Both playlisters actually seem to have a tendency to produce always the same coherence values, independently of the coherence values of the seed. A last interesting result is the high coherence of artist genres on the AotM and 8tracks datasets – the high genre coherence values on last.fm can be explained by the high artist coherence values.

4. STANDARD ACCURACY METRICS

Our analysis so far has revealed some particular characteristics of user-created playlists. Furthermore, we observed that the nearest-neighbor playlisting scheme can produce playlists that are quite different to those generated by the commercial Echo Nest service, e.g., in terms of average track popularity (Table 3).

In the research literature, "hit rates" (recall) and the average log-likelihood ($\mathcal{ALL}$) are often used to compare the quality of playlists generated by different algorithms [2, 8, 11]. The goal of our next experiment was to find out how The Echo Nest playlister performs on these measures. As it is not possible to acquire probability values for the tracks selected by The Echo Nest playlister, the $\mathcal{ALL}$ cannot be

[8] In the case of artists, this means that the similarity equals 1 if both tracks have the same artist, and 0 else. The metric thus measures the proportion of cases when the users consecutively selected tracks from the same artist.

used[9]. In the following we thus only focus on the precision and recall.

The upper part of Figure 6 shows the recall values at list length 100 for the different datasets[10]. Again, we split the playlists and used the first half as seed tracks. Recall was then computed by comparing the computed playlists with the "hidden" tracks of the original playlist. We measured recall for tracks, artists, genres and tags. The results show that the kNN method quite clearly outperforms the playlister of The Echo Nest on the recall measures across all datasets except for the artist recall for the last.fm dataset. The differences are statistically significant for all the experiments except for the track and artists recall on last.fm ($p < 10^{-6}$) according to a Student's t-test. As expected, the kNN method leads to higher absolute values for larger datasets as more neighbors can be found.

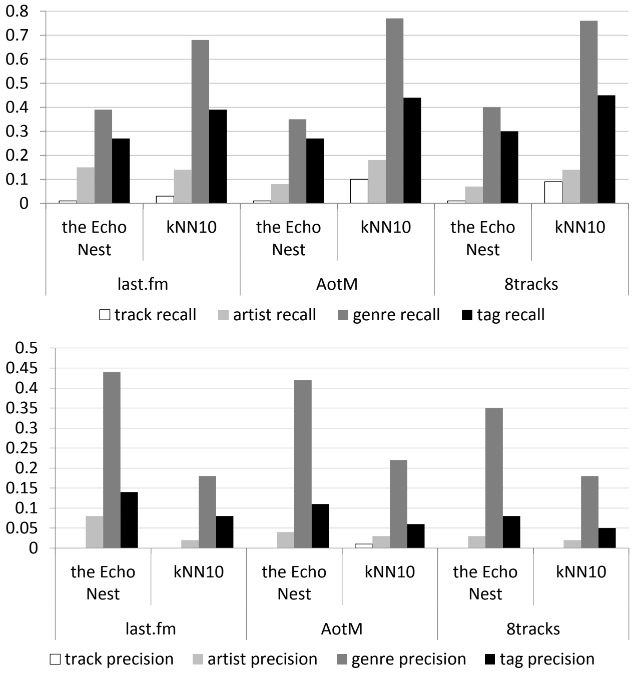

Figure 6: Recall and Precision for the covered cases.

The lower part of Figure 6 presents the precision results. The precision values for tracks are as expected very low and close to zero which is caused by the huge set of possible tracks and the list length of 100. We can however observe a higher precision for the kNN method on the AotM dataset ($p < 10^{-11}$), which is the largest dataset. Regarding artist, genre and tag prediction, The Echo Nest playlister lead to a higher precision ($p < 10^{-3}$) than the kNN playlister on all datasets.

With respect to the evaluation protocol, note that we only measured precision and recall when the playlister was able to return a playlist continuation given the seed tracks. This was however not always the case for both techniques. In Table 6, we therefore report the detailed coverage figures, which show that the kNN method was more often able to produce a playlist. If recall is measured for all seed playlists, the differences between the algorithms are even larger. When measuring precision for all playlists, the differences between the playlisters become very small.

Dataset	tEN	kNN
last.fm	28.33	66.89
AotM	42.75	86.52
8tracks	35.3	43.8

Table 6: Coverage of the playlisters.

Overall, measuring precision and recall when comparing generated playlists with those provided by users in our view represents only one particular form of assessing the quality of a playlist generator and should be complemented with additional measures. Precision and recall as measured in our experiments for example do not consider track transitions. There is also no "punishment" if a generated playlist contains individual non-fitting tracks that would hurt the listener's overall enjoyment.

5. PUBLIC AND PRIVATE PLAYLISTS

Some music platforms and in particular 8tracks let their users create "private" playlists which are not visible to others and public ones that for example are shared and used for social interaction like parties, motivation for team sport or romantic evening. The question arises if public playlists have different characteristics than those that were created for personal use only, e.g., because sharing playlists to some extent can also serve the purpose of creating a public image of oneself.

We made an initial analysis on the 8tracks dataset. Table 7 shows the average popularity of the tracks in the 8tracks playlists depending on whether they were in "public" or "private" playlists (the first category contains 2679 playlists and the second 451). As can be seen, the tracks of the private playlists are much more popular on average than the tracks in the public playlists. Moreover, as indicated by the corresponding Gini coefficients, the popular tracks are almost equally distributed across the playlists. Furthermore, Figure 7 shows the corresponding freshness values. We can see that the private playlists generally contained more recent tracks than public playlists.

	Play counts	Gini index
Public playlists	870k	0.20
Private playlists	935k	0.06

Table 7: Popularity of tracks in 8tracks public and private playlists and Gini index.

These results can be interpreted at least in two different ways. First, users might create some playlists for their personal use to be able to repeatedly listen to the latest popular tracks. They probably do not share these playlists because

[9] Another possible measure is the Mean Reciprocal Rank (MRR). Applied to playlist generation, one limitation of this metric is that it corresponds to the assumption that the rank of the test track or artist to predict should be as high as possible in the recommendation list, although many other tracks or artist may be more relevant and should be ranked before.

[10] We could not measure longer list lengths as 100 is the maximum playlist length returned by The Echo Nest.

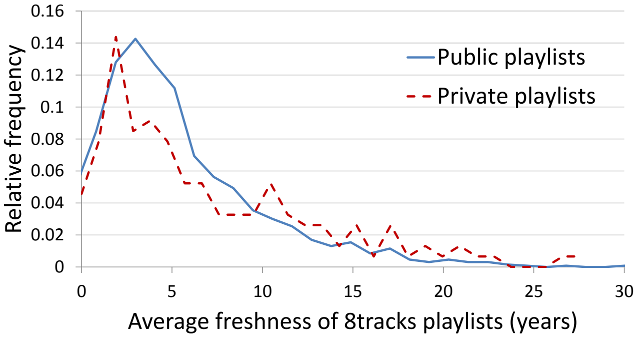

Figure 7: Distribution of average freshness of 8tracks public and private playlists.

sharing a list of current top hits might be of limited value for other platform members who might be generally more interested in *discovering* not so popular artists and tracks. Second, users might deliberately share playlists with less popular or known artists and tracks to create a social image on the platform.

Given these first observations, we believe that our approach has some potential to help us better understand some elements of user behavior on social platforms in general, i.e., that people might not necessarily only share tracks that match their actual taste.

6. SUMMARY AND OUTLOOK

The goal of our work is to gain a better understanding of how users create playlists in order to be able to design future playlisting algorithms that take these "natural" characteristics into account. The first results reported in this paper indicate, for example, that features like track freshness, popularity aspects, or homogeneity of the tracks are relevant for users, but not yet fully taken into account by current algorithms that are considered to create high-quality playlists in the literature. Overall, the observations also indicate that additional metrics might be required to assess the quality of computer-generated playlists in experimental settings that are based on historical data such as existing playlists or listening logs.

Given the richness of the available data, many more analyses are possible. Currently, we are exploring "semantic" characteristics to automatically identify the underlying theme or topic of the playlists. Another aspect not considered so far in our research is the popularity of the playlists. For some music platforms, listening counts and "like" statements for playlists are available. This additional information can be used to further differentiate between "good" and "bad" playlists and help us obtain more fine-granular differences with respect to the corresponding playlist characteristics. Last, we plan to extend our experiments and analysis by considering other music services, in particular last.fm radios, and other playlisting algorithms, in particular algorithms that exploit content information.

7. REFERENCES

[1] A. Andric and G. Haus. Estimating Quality of Playlists by Sight. In *Proc. AXMEDIS*, pages 68–74, 2005.

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

[3] G. Bonnin and D. Jannach. Automated generation of music playlists: Survey and experiments. *ACM Computing Surveys*, 47(2), 2014.

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

[5] S. Cunningham, D. Bainbridge, and A. Falconer. 'More of an Art than a Science': Supporting the Creation of Playlists and Mixes. In *Proc. ISMIR*, pages 240–245, 2006.

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

[7] D. L. Hansen and J. Golbeck. Mixing It Up: Recommending Collections of Items. In *Proc. CHI*, pages 1217–1226, 2009.

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

[9] M. Kamalzadeh, D. Baur, and T. Möller. A Survey on Music Listening and Management Behaviours. In *Proc. ISMIR*, pages 373–378, 2012.

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

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

[12] G. Reynolds, D. Barry, T. Burke, and E. Coyle. Interacting With Large Music Collections: Towards the Use of Environmental Metadata. In *Proc. ICME*, pages 989–992, 2008.

[13] A. M. Sarroff and M. Casey. Modeling and Predicting Song Adjacencies In Commercial Albums. In *Proc. SMC*, 2012.

[14] M. Slaney and W. White. Measuring Playlist Diversity for Recommendation Systems. In *Proc. AMCMM*, pages 77–82, 2006.