# Biases in Automated Music Playlist Generation: A Comparison of Next-Track Recommending Techniques

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

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

Geoffray Bonnin
LORIA, Nancy, France
geoffray.bonnin@
loria.fr

## ABSTRACT

Playlist generation is a special form of music recommendation where the problem is to create a sequence of tracks to be played next, given a number of seed tracks. In academia, the evaluation of playlisting techniques is often done by assessing with the help of information retrieval measures if an algorithm is capable of selecting those tracks that also a human would pick next. Such approaches however cannot capture other factors, e.g., the homogeneity of the tracks that can determine the quality perception of playlists. In this work, we report the results of a multi-metric comparison of different academic approaches and a commercial playlisting service. Our results show that all tested techniques generate playlists with certain biases, e.g., towards very popular tracks, and often create playlists continuations that are quite different from those that are created by real users.

## Keywords

Music Recommendation; Bias; Evaluation

## 1. INTRODUCTION

The automated generation of playlists is a central feature of today's music player applications and web platforms like Spotify, Deezer, or iTunes. One specific problem setting in this context often is to generate a list of next tracks to be played (a playlist), given the most recent listening history of a user. This playlist generation or playlist continuation problem can be considered as a special form of music recommendation, for which a number of algorithmic proposals were made in the research literature, see [6] or [15].

The evaluation and comparison of such playlist generation techniques is unfortunately challenging in research settings. User studies are typically expensive as the participants would have to actually listen to a number of tracks. In addition, a number of possible quality criteria – as analyzed in [13] or [22] – generally exist, among them the *homogeneity* or *diversity* of the generated playlists or the *smoothness* of track transitions [4, 17, 24, 25].



*UMAP '16, July 13-17, 2016, Halifax, NS, Canada*



DOI: http://dx.doi.org/10.1145/2930238.2930283

One possible alternative to user studies is to use manually created ("hand-crafted" [5]) playlists as a reference point when assessing the quality of a playlister algorithm. This approach is common in the literature, e.g., [5, 7, 9, 18] to assess the capability of an algorithm of recommending those tracks that were also picked by users for the inclusion in a playlist. One of the typical evaluation methods in this context is to hide one track from a known playlist and let the algorithms predict the hidden track. *Recall* as a performance measure from the field of information retrieval can then be used to quantify the recommendation accuracy [5, 9, 21, 26]. One implicit underlying assumption of such approaches is that the user-created playlists were carefully designed which means that the tracks in the playlists are also appropriate in terms of other quality criteria like homogeneity.

Many algorithms are designed to optimize an *accuracy* criterion like the recall. However, recommendation algorithms can be quite different in terms of which items they actually recommend even when their predictive accuracy is comparable [1, 12]. Some algorithms, for example, are strongly biased to popular items whereas others have a tendency to recommend the same small set of items to everyone.

The long-term goal of our work is to better understand to which extent different playlisting techniques are capable of generating playlists that are similar to human-generated ones. In this work, we report the results of a multi-metric analysis in which we compared the playlist continuations that were generated by different algorithms with the continuations of the users. We use different accuracy, coherence, and diversity measures and assess to which extent some algorithms have biases, e.g., toward popular items. Our analysis includes both academic techniques of different types as well as the commercial playlist generation service provided by The Echo Nest, a subsidiary company of Spotify.

## 2. A MULTI-DIMENSIONAL COMPARISON

### 2.1 Experiment Setup

#### 2.1.1 Next-Track Recommendation Algorithms

A variety of playlist generation approaches have been proposed over the last fifteen years that are, e.g., based on content similarity, collaborative filtering, Markov models, discrete optimization, and hybrid techniques [3, 7, 8, 16, 18].

In the experiments reported in this paper, we include two *collaborative filtering* methods that have shown to lead to high accuracy in terms of the recall, a *content-based* technique based on social tags, a *hybrid* method from the recent

literature, and a *commercial* playlisting service. The general task of all techniques is to determine the relevance of each *target track* $t^*$ w.r.t. a given playlist beginning or listening history $h$. The algorithms can be summarized as follows.

*(1) Collocated Artists - Greatest Hits (CAGH):* CAGH is an artist-based approach that recommends the greatest hits of artists that already appear in the playlist beginning $h$ or are similar to the artists in $h$. The co-occurrence of artists in the training data is used as a similarity measure [6].

*(2) kNN300:* This k-Nearest-Neighbor-based method takes the playlist beginning $h$ as an input and looks for other playlists in the training data that contain the same tracks. Different works show that this technique represents a strong baseline [6, 9, 11]. In our experiments, we set $k = 300$.

*(3) Content-Based:* This technique is based on social tags and ranks the tracks using the cosine similarity of the social tags assigned to the tracks. Like [11], we first compute TF-IDF vectors for each track. Given a recent history $h$, a tag-based similarity score is then computed as the cosine similarity of the averaged TF-IDF vector of the recent history and the TF-IDF vector of the target track $t^*$.

*(4) kNN300PCPA:* An improved version of the most accurate hybrid playlister from [11]. This playlister is based on a weighted combination of a personalized extension of the kNN-based approach with additional suitability scores.

*(5) TEN:* The commercial playlister of The Echo Nest[1]. Although we cannot know which algorithms are internally used or whether the recommendations are influenced by commercial considerations or constraints, the comparison can be interesting as the recommendations produced by the service result from several years of A/B-testing.

### 2.1.2 Evaluation Datasets

We used pools of playlist collections from three music platforms. One set was obtained from Last.fm via their public API. One was published by [19] and contains playlists created by music enthusiasts on the Art-of-The-Mix website and one collection was shared with us by 8tracks[2]. All datasets used in the experiments except the non-public one from 8tracks are available online[3].

All datasets have certain distinctive characteristics. For example, Last.fm playlists often contain only tracks of one or very few artists. This is very uncommon for playlists from AotM and forbidden for public playlists from 8tracks where each artist can only appear twice. In order to not introduce any bias, we however did not apply any heuristics-based filtering for any of the datasets to retain only playlists for which we assume that the creators invested some effort.

Table 1 shows the basic dataset statistics. Each user has at least 4 playlists, which allows us to personalize the kNN and the content-based methods used in the hybrid approach. As in [11], we retrieved additional track information (e.g., tempo, release year, social tags) using the public APIs of Last.fm, theechonest.com, and musicbrainz.org. Note that this data is incomplete and only 75% of the data points existed on average across all tracks. For all considered playlists we however ensured that certain minimum levels of the different meta-data fields were available.

[1]http://theechonest.com
[2]http://last.fm, http://artofthemix.org, http://8tracks.com
[3]http://ls13-www.cs.tu-dortmund.de/homepage/umap16-music/datasets.zip

**Table 1: Dataset Statistics.**

| Measure | Last.fm | AotM | 8tracks |
|---|---|---|---|
| Playlists | 2,978 | 1,040 | 6,714 |
| Users | 451 | 142 | 996 |
| Tracks | 18,083 | 11,413 | 39,875 |
| Artists | 3,272 | 2,770 | 9,122 |
| Avg. Playlists/User | 6.60 | 7.32 | 6.74 |
| Avg. Tracks/Playlist | 11.68 | 16.98 | 12.97 |
| Avg. Artists/Playlist | 4.55 | 12.76 | 12.06 |
| Avg. Genres/Playlist | 16.83 | 39.18 | 38.06 |
| Avg. Tags/Playlist | 94.88 | 140.62 | 123.07 |

### 2.1.3 Measurement Method

Since our goal is to compare generated playlist continuations with those made by users, it is insufficient to hide only the last track of each playlist as done in the literature[4]. We therefore propose to split the playlists in the test set into two halves as shown in Figure 1. The *seed* half is first provided as an input to the playlister whose task is to generate a continuation that is as long as the seed half. We then compare the generated continuation with the held-out *test* half, to measure how good an algorithm is able to mimic the behavior of a human. Furthermore, we apply different other measures to assess the quality of the continuations. As usual, we split each playlist dataset into training (75%) and test sets (25%) on a per-user criterion and apply a four-fold cross-validation procedure.

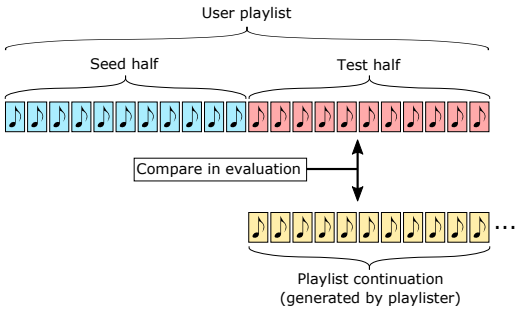


**Figure 1: Proposed Evaluation Protocol**

## 2.2 Accuracy Results

We measured four variants of precision and recall. Specifically, we determined if the algorithms recommended (a) the right tracks, (b) the relevant artists, (c) the correct genres, or (d) tracks with suitable tags. Note that we do not measure precision or recall at a predefined, static list length (e.g., precision@n), but include all recommended tracks in the measurement. Remember that the number of recommended tracks that we consider is equal to the number of seed tracks (Fig. 1). The results are shown in Fig. 2.

*Tracks.* The hybrid method (kNN300PCPA), as its predecessor in [11], outperformed all other techniques in terms of *track* precision and recall. The differences between this algorithms and the other algorithms are all statistically significant at $p < 0.05$ according to a Student's t-test, except for the CAGH playlister on Last.fm and AotM.

[4]The last tracks might in addition be non-representative for the playlist as a whole as discussed in [2].

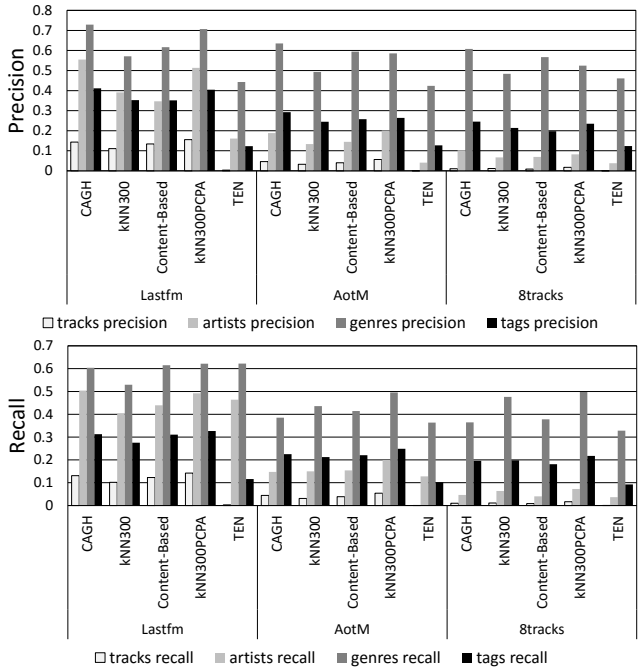


**Figure 2: Precision and recall of the playlisters**

*Artists, Genres, and Tags.* Finding exactly the hidden tracks can be challenging as is also indicated by the comparably low absolute precision and recall values. It might therefore be sufficient in practice to play music of similar artists, related genres or to find tracks that are similar in terms of their social tags. In such situations, using the comparably simple CAGH method appears to be sufficient, which for example leads to competitive precision and recall values in terms of artist, genres, and tags. Particularly on the Last.fm dataset, the effectiveness of CAGH is not surprising as these playlists often contain tracks of a few artists. The recall of CAGH on the other datasets is often a bit lower, as CAGH focuses on a too small set of artists, genres, or tags.

*The Echo Nest.* The commercial playlister consistently leads to the *lowest precision values* ($p < 0.05$). It also has the lowest recall values in 10 out of the 12 cases, although statistically significant differences could not always be found. This can be an indication that the playlists that are generated by the commercial service are not necessarily (exclusively) optimized for precision or recall and that also other criteria govern the track selection process. Although we cannot know the internals of The Echo Nest playlister, the analyses presented in the next sections can give us some hints about inherent biases of the commercial technique, which has assumedly been optimized over time[5].

## 2.3 Coherence Analysis

So far, we have focused on determining to which extent different algorithms are capable of finding the right tracks, artists, or genres to play. Analyses like [12] however show that algorithms that lead to high precision and recall values often have a popularity bias. However, recommending only popular tracks as a continuation to a playlist that also contains niche tracks might lead to a limited quality perception by users. Likewise, if the user has just listened to a set of tracks with a certain "theme" and all tracks have a similar, e.g., low tempo, playing popular tracks with mixed tempos might be inappropriate.

In this section, we therefore analyze if our playlisters produce continuations that are *coherent* with the playlist beginnings in terms of two selected features, popularity and tempo. In addition, we will compare the generated continuations with those that were created by the users. We will look at the mean as well as at the distribution of the values.

### 2.3.1 Comparing Averaged Playlist Features

Figure 3 shows the mean tempo and popularity (playcount) values of the seed halves, test halves and generated continuations for each dataset. To make a fair comparison, we only considered those cases in the chart when each playlister could actually create a continuation. In a few cases, some playlisters could not return a playlist, which is why the first two bars (for the seed and test halves) not always have the exact same height.

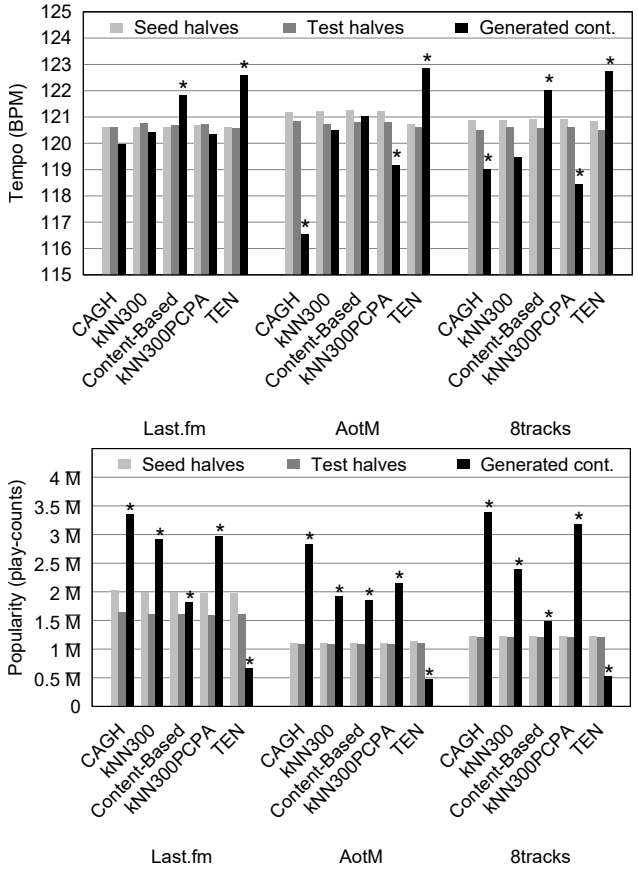


**Figure 3: Comparison of average popularity and tempo values (* indicates statistical significance).**

*Tempo.* The upper part of Fig. 3 shows that in general the tempo difference between the two halves in user-created playlists is small, as indicated by the light gray and dark gray bars. Most playlisters are successful in reproducing this behavior of users. Some playlisters (Content-Based

[5]As of 2016, The Echo Nest covers more than 35 million tracks. Nonetheless, there might be some tracks in our playlist datasets that are not known to the commercial service and could therefore not be recommended.

and CAGH) seem to have a slight tendency to play slower tracks and the TEN playlister seems to increase the tempo. The maximum average distance is however only 6 beats per minute, which might not be noticed by users. We repeated the measurement for the *year of release*, and found a similar trend that the playlisters could reproduce what users did.

*Popularity.* All academic playlisters, in particular those with high accuracy, show a strong bias toward popular tracks. For kNN300, kNN300PCPA and CAGH, this can be explained by their co-occurrence based design. The Content-Based playlister might have this bias because more social tags exist for popular tracks.

The Echo Nest, in contrast, exhibits exactly the opposite bias and tends to recommend less popular tracks than the users would have chosen. Overall, no playlister successfully reproduced the general level of popularity preferred by the users. We found similar trends for the *loudness* feature of the tracks which we do not report here due to space limitations.

*Discussion.* Both in terms of tempo and popularity, users seem to prefer playlist continuations (second halves) that are coherent with the first halves. The evaluated playlisters however can only reproduce this behavior for some features but also often introduce comparably strong biases.

### 2.3.2 Diversity (Variance) Observations

To assess whether the playlisters are capable of matching the diversity levels of user-created playlists, we determined the average tempo and popularity distance between all pairs of tracks of each playlist (Fig. 4). We report absolute distance values which allows us to interpret the results directly.

*Tempo.* All playlisters were more or less able to mimic the behavior of the users. The overall tempo diversity (variance) is comparably high and was reproduced by all playlisters. Similar observations were made for the *year of release*, where the variance in general was very low and playlists often only contained tracks from one single decade.

*Popularity.* Almost all academic playlisters not only focus on more popular tracks as seen in the previous section, the variability in terms of the popularity is also much higher than for the user-created test halves[6]. The only exception is the Content-Based playlister which is comparably successful in maintaining the diversity level preferred by the users.

The Echo Nest playlister, in contrast, tends to reduce the diversity more than users do in their continuations. Similar trends could be observed for the *loudness* of the tracks and all playlisters except the Content-Based method and TEN led to higher diversity levels when compared to the user-created continuations.

Overall, the measurements confirm that some of the playlisters exhibit biases to generate either comparably homogeneous or slightly more diverse next-track recommendations.

## 3. DISCUSSION AND OUTLOOK

Through the novel measurement method proposed in this paper we could see that automated algorithms sometimes create playlist continuations that are quite different from user-created ones and often exhibit certain, possibly undesired, biases. The results therefore emphasize that considering one single evaluation measure in offline experiments can

[6]These observations have to be put in perspective with the overall high diversity values of the seed and test halves. If we refer to Fig. 3, the mean popularity values of these halves range between a play-count of 1 and 2 million.

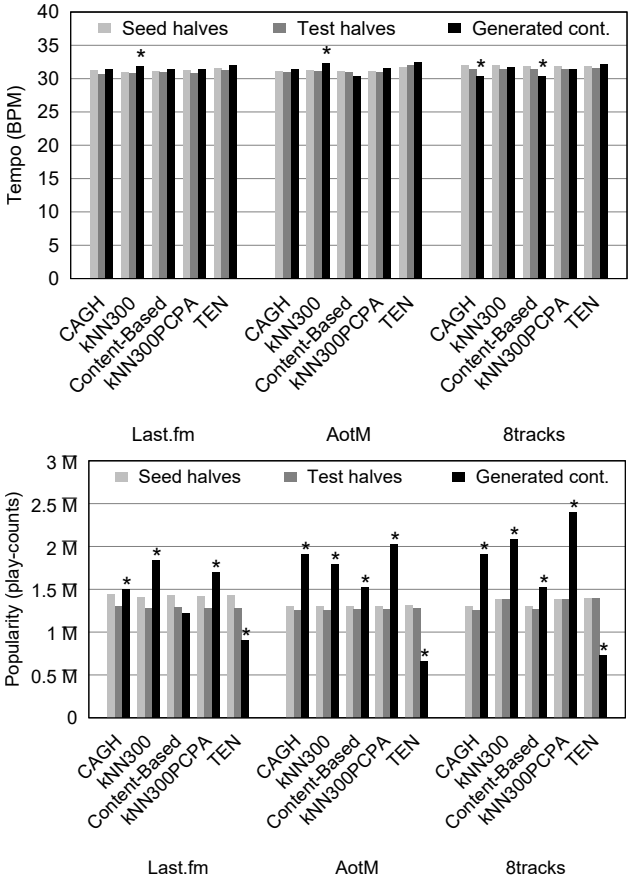


**Figure 4: Comparison of Diversity Values in Terms of Tempo and Popularity (play-counts).**

be insufficient also in the domain of music recommendation, which calls for multi-metric evaluation approaches as advocated, e.g., in [12] or [23]. Our work furthermore revealed that academic playlisting algorithms often produce recommendations that are largely different from those generated by a commercial service, an aspect which has not been examined in the literature before. Whether these variations can lead to measurable differences in quality perception by the users can only be answered through user studies.

A particularity of the playlist generation problem is that characteristics of the recommendation set *as a whole* (including homogeneity, diversity and track transitions) as well as the transitions between the tracks can determine the quality perception of the users [14]. So far, we have only considered a limited number of musical or meta-data features which we could obtain from public music databases.

The goal of our current work is to acquire and exploit additional information about the tracks, which shall help us better understand the underlying "theme" of a playlist (e.g., based on the lyrics) and to assess to which extent different algorithms are able to continue the playlist in this respect.

The ultimate goal is then to design new algorithmic approaches which are explicitly designed to create playlists that match one or several of the characteristics of the playlist beginnings. Specifically, one goal could be to focus on those characteristics that are particularly relevant for the individual user, as proposed in [10, 11] or [20].

# 4. REFERENCES


[1] P. Adamopoulos and A. Tuzhilin. On Over-specialization and Concentration Bias of Recommendations: Probabilistic Neighborhood Selection in Collaborative Filtering Systems. In *Proc. RecSys '14*, pages 153–160, 2014.

[2] A. Andric and G. Haus. Automatic Playlist Generation based on Tracking User's Listening Habits. *Multimedia Tools and Applications*, 29(2):127–151, 2006.

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

[4] W. Balkema and F. van der Heijden. Music Playlist Generation by Assimilating GMMs into SOMs. *Pattern Recognition Letters*, 31(11):1396–1402, 2010.

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

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

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

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

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

[10] T. Jambor and J. Wang. Optimizing Multiple Objectives in Collaborative Filtering. In *Proc. RecSys '10*, pages 55–62, 2010.

[11] 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.

[12] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac. What Recommenders Recommend: An Analysis of Recommendation Biases and Possible Countermeasures. *User Modeling and User-Adapted Interaction*, pages 1–65, 2015.

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

[14] P. Lamere. I've got 10 million songs in my pocket: now what?. In Proc. RecSys 2012, pages 207–208, 2012.

[15] 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.

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

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

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

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

[20] J. Oh, S. Park, H. Yu, M. Song, and S. Park. Novel Recommendation Based on Personal Popularity Tendency. In *Proc. ICDM '11*, pages 507–516, 2011.

[21] J. C. Platt, C. J. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian Process Prior for Automatically Generating Music Playlists. In *Proc. NIPS*, pages 1425–1432, 2001.

[22] 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.

[23] A. Said, B. J. Jain, and S. Albayrak. A 3D Approach to Recommender System Evaluation. In *Proc. CSCW '13 Companion Volume*, pages 263–266, 2013.

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

[25] M. Slaney and W. White. Measuring Playlist Diversity for Recommendation Systems. In *Proc. AMCMM '06*, pages 77–82, 2006.

[26] L. Xiao, L. Lu, F. Seide, and J. Zhou. Learning a Music Similarity Measure on Automatic Annotations with Application to Playlist Generation. In *Proc. ICASSP*, pages 1885–1888, 2009.