Generating Smooth Mood-Dynamic Playlists with Audio Features and KNN

Shaurya Gaur[1,2(✉)] and Patrick J. Donnelly[2]

[1] Radboud University, Nijmegen, The Netherlands
shaurya.gaur@ru.nl
[2] Oregon State University, Corvallis, USA
patrick.donnelly@oregonstate.edu

Abstract. Users curate music playlists for many purposes, including focus, enjoyment and therapy. Popular music streaming services generate playlists automatically which are constant in genre or mood. We propose a method to automatically create playlists dynamic in both the Arousal-Valence emotion space and the audio features of songs. Our playlist algorithm uses a two-stage approach to sequentially choose songs, employing a K-Nearest Neighbors (KNN) model to gather potential songs based on emotion and analyzing them with acoustic similarity metrics. To evaluate the effectiveness of various audio feature data, KNN parameters, and similarity metrics, we developed a testing protocol which generates playlists that traverse both Arousal-Valence and audio feature spaces. We define evaluation metrics to measure a playlist's smoothness and evenness using the Pearson correlation coefficient between dimensions and the variance of steps between songs, respectively. Our algorithm successfully creates smooth and evenly-spaced playlists that transition cohesively in both mood and genre. We explore how the choice of audio feature data, similarity metric, and KNN parameters all have an effect on playlists' smoothness and evenness across these two spaces.

Keywords: K-Nearest Neighbors · Affective Computing · Automatic Playlist Generation

1 Introduction

Listening to as few as five minutes of music while working or studying has been correlated with improved mood [17] and productivity [14,15]. In particular, listening to songs with a major key [3] and high tempo [13] has correlated with one's happiness in conversations and stress during exercise, respectively, though these relationships vary on listeners' preferences and familiarity with the songs [10] and their listening environment [15]. It has also been shown that playlists with smooth transitions between songs can be more satisfying to the ear [31].

S. Gaur—Work completed at Oregon State University.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
C. Johnson et al. (Eds.): EvoMUSART 2024, LNCS 14633, pp. 162–178, 2024.
https://doi.org/10.1007/978-3-031-56992-0_11

We seek to create playlists that smoothly transition a listener from their current emotional state towards one they desire, while also gently nudging them through different styles of songs. To examine the affect of songs, we rely on Russell's circumplex model [30], which maps emotions to continuous points on a two-dimensional space by their Valence (positivity) and Arousal (intensity). This model's continuous nature makes it a popular representation of emotions and multiple studies use it to generate playlists of songs similar in affect [8,23].

There are several ways to generate automatic playlists. Some researchers approach it as a continuation problem, using audio similarity [25], collaborative filtering [23], or K-Nearest Neighbor (KNN) techniques [19] to recommend the next song for a user based on their listening history or preferred songs. Others approach this recommendation problem in two stages [36], first recalling many candidate songs and then re-scoring them to choose the single best song.

Various studies seek to build playlists that maintain the user's current mood. Some approaches integrate sensor data [16], facial images [24], or text input [22] to recognize a listener's emotion, while others use audio features [8,23] to predict the affect of recently-played songs. Cardoso et al. [5] develop a tool that allows users to make playlists of a single mood based on a seed song, or to draw a path in the Valence-Arousal plane to create a playlist with songs that lie within a specified distance to the path. However, this study does not specify an order for these songs or make an attempt to evaluate their dynamic playlists.

Recent approaches aim to generate playlists which transition between songs or artists. Some create new playlists based on existing playlist data [6,27], while others build similarity graphs of acoustic features [31] or artist popularity data[1] to guide song order. Flexer et al. [12] use spectral features to represent songs as Gaussian mixture models (GMMs) and create playlists based on their similarity to specified start and end songs. While this method creates dynamic playlists, the study only evaluates their quality through discrete genre labels, and existing music datasets do not contain enough emotion data to follow this approach [7].

We build upon previous work [9] to create an algorithm which builds playlists dynamic in both emotional and audio qualities. Our two-stage algorithm [36] utilizes a nearest-neighbors search [5,12] to choose songs sequentially, forming a linear path. We evaluate playlists by their smoothness and evenness, and demonstrate our algorithm's ability to create an emotional and audio journey in music.

2 Datasets and Preprocessing

To approach this task, we select the Deezer 2018 dataset, which contains Valence and Arousal estimations of 18,644 songs [7]. Researchers from the Deezer streaming service collected each song's affect-related tags from LastFM[2], which have been shown to correlate with the perceived emotional qualities of a song [33]. Using a dataset that mapped English words to embeddings in the Valence-Arousal space, the Deezer team aggregated and normalized these LastFM tags'

[1] https://musicmachinery.com/2013/01/02/boil-the-frog-2/.
[2] https://www.last.fm/.

emotion scores. This approach uses crowd-sourced tags for songs instead of manual Valence-Arousal annotations, making these estimations inherently synthetic. As a result, the Deezer dataset yields only 2,762 unique points for 18,644 songs.

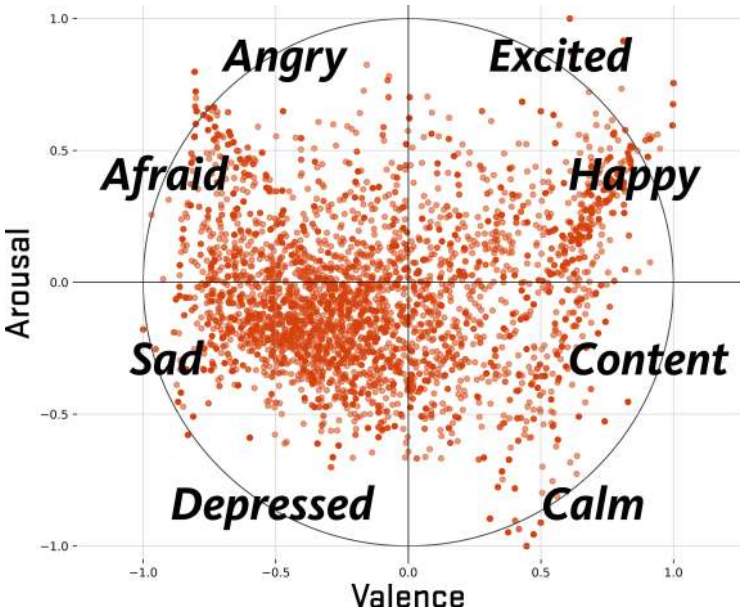


Fig. 1. Visualization of the Deezer dataset [7] in the Valence-Arousal circumplex space. Note the dense clusters of songs in the "Happy" and "Sad" regions.

Figure 1 shows the relative distribution of the dataset in these two dimensions. We observe that this distribution skews towards the region of sad songs, with smaller clusters of songs in the regions that indicate happiness and fear. Although its Valence-Arousal scores are based on synthetic labeling instead of manual human ratings, and they feature duplicate points and an uneven distribution, the Deezer dataset provides a large sample of popular songs from which to build playlists and more easily align with other datasets or streaming services.

2.1 Audio Feature Datasets

We aim to create mood-dynamic playlists that also feature smooth audio-based transitions between their first and final songs. To align with the format of the Deezer dataset, we seek acoustic features for an entire song. There are various datasets which collect audio features and other relevant metadata for large libraries of songs, such as the Spotify API[3] and the Million Song Dataset [2]. We augment the emotion data of the Deezer dataset with features from these

[3] https://developer.spotify.com/documentation/web-api/.

two sources. In particular, we extract as many features as possible which relate directly to either a song's audio or its cultural perception, since a song's content and societal context factor into its emotional effect on listeners [10].

Spotify Audio Features. We first extract features from two endpoints of Spotify's API: `Get Audio Features`, which returns aggregate acoustic features obtained through processing raw audio [26], and `Get Track`, which contains track metadata, such as a song's popularity (from 0 to 100) and a flag indicating explicit language. To locate a matching Spotify track ID for each Deezer song, we query the API with the song's title and artist. This successfully returned IDs for 17,755 of the Deezer dataset's 18,644 songs (95.2%). Table 1 shows the 15 features we extract from Spotify. These are primarily continuous, except for a song's key, modality, time signature, popularity, and use of explicit language.

Table 1. Statistics for 13 features from Spotify's `Get Audio Features` API endpoint, and two features from the `Get Track` endpoint, on 17,755 Deezer songs collected.

	Mean	St.Dev.	Min	Median	Max
acousticness	0.2618	0.3051	0	0.1110	0.995
danceability	0.5242	0.1590	0	0.5270	0.978
duration_ms	251336	81137	35587	239	1449973
energy	0.6222	0.2462	0	0.6510	0.999
instrumentalness	0.0844	0.2089	0	0.0002	0.983
key	5.2506	3.5622	0	5	11
liveness	0.1878	0.1557	0.0136	0.1250	1
loudness	−8.4547	3.9755	−46.2840	−7.6410	3.744
mode	0.6823	0.4656	0	1	1
speechiness	0.0605	0.0632	0	0.0387	0.954
tempo	122.0198	29.2322	0	120.1160	217.520
time_sig	3.9018	0.3680	0	4	5
valence	0.4609	0.2481	0	0.4360	0.984
popularity	34.1625	16.6780	0	33	84
explicit	0.0444	0.2059	0	0	1

Million Song Dataset. To compare the utility of two sources of audio features, we also collect from the Million Song Dataset (MSD) [2]. One of the most popular datasets in music information retrieval [1,18], the MSD includes acoustic features and metadata for 1,000,000 songs released between 1922 and 2011 from 44,745 artists [2]. The Deezer dataset includes a MSD track ID for each song, which we use to connect its Valence-Arousal points with MSD features. We first collect each song's key, mode, and time signature (as well as confidence measures for

these estimations), and estimations on loudness and tempo. The MSD provides peak loudness and spectral features for each segment of the song. To align with our other datasets, we aggregate the mean, variance, minimum, maximum, and median of the segment-based values to capture the peak loudness and timbre for a whole song. In total, we collect 78 features from the MSD dataset.

2.2 Preprocessing

Since the Spotify dataset has only 15 features compared to the MSD's 78, the difference in dimensionality may have an impact, as the "curse of dimensionality" is particularly problematic in nearest-neighbors approaches [20]. To investigate this, we first apply Principal Component Analysis (PCA) on both the Spotify and MSD datasets, and the combination of the two. To maintain a consistent feature size, we reduce each dataset to 12 principal components. Table 2 describes the variance retained from the transformation on each audio feature dataset.

Table 2. Variance retained after reducing to 12 dimensions with PCA on each dataset. The Spotify, containing fewer original features, retains most of its variance.

Dataset	Features	Variance Retained
Spotify	15	96.7%
MSD	78	68.3%
All (Spotify + MSD)	93	62.7%

Because KNN approaches work best when features are on the same scale, we next transform our data such that no distance in one dimension overweighs others. To mitigate against outliers, we employ a Yeo-Johnson power transformation on each feature [35]. While this reduced their prominence, some features still retained extreme outliers, which we define as values that stray $5.5 \times IQR$ from the quartiles. To combat this, we discretize features that contained extreme outliers into 20 bins. Lastly, we scale all features to the $[-1, 1]$ range.

3 Playlist Algorithm

Given a pair of songs and a desired playlist length, our algorithm generates a playlist, such as the one in Fig. 2, in which the songs form a path between the two songs in the Valence-Arousal space and a specified audio feature space. We choose each song in a two-stage loop, calculating the trajectory from the most recent song to the chosen final song with both emotional and acoustic features.

3.1 Stage 1: K-Nearest Neighbors

We first employ a K-Nearest Neighbors (KNN) model to recall a specified K candidate points from the Deezer dataset. Given the last chosen song S_c and

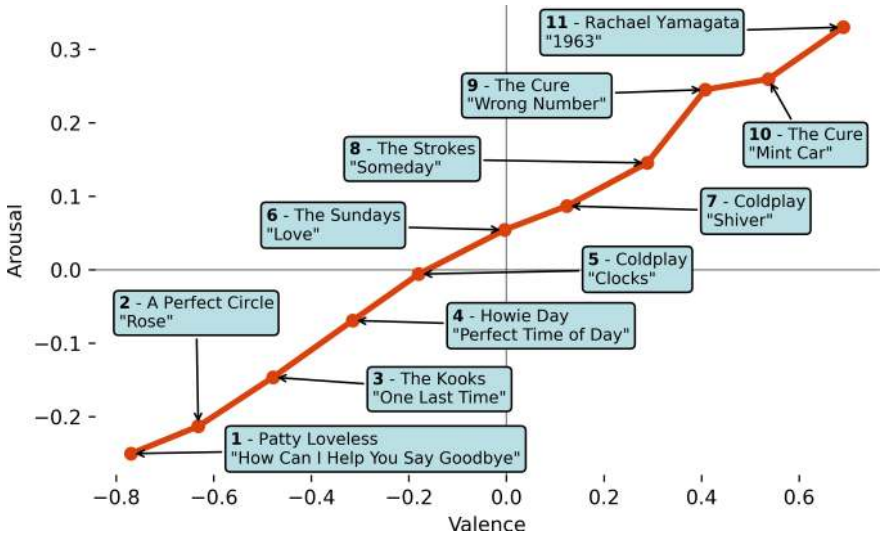


Fig. 2. A playlist generated by our algorithm which begins at sad songs (negative Valence and Arousal) and gradually transitions to happy and excited songs.

the final song S_f as points in the Valence-Arousal space, and the number of songs remaining N, we calculate a hypothetical target point S_t using Eq. (1), representing the first of N equal steps in the linear trajectory from S_c to S_f.

$$S_t = S_c + \frac{S_f - S_c}{N} \tag{1}$$

Utilizing an unsupervised `NearestNeighbors`[4] model from `scikit-learn` [28], which stores distances in a K-dimensional (KD) tree, we query the LK Valence-Arousal points closest to S_t, where L is the playlist's current length. The distance of these candidate neighbors depends on the density of the Deezer dataset's points at the target region. This query has an $\mathcal{O}[P \log(P)]$ time complexity, where $P = 2,762$ is the number of unique points in the Deezer dataset.

We use the same KD tree throughout the playlist creation process for time efficiency, since rebuilding the KD tree has an $\mathcal{O}[P \log(P)]$ time complexity [28]. However, this means that points already in our playlist may reappear as candidates, which requires us to query for LK neighbors and manually filter to find the closest K not yet chosen. This avoids repetition in the playlist. Though this method has a worst-case time complexity of $\mathcal{O}[L^2K]$, L is likely to be far smaller than P or K, making manual filtering more efficient. We then extract the song IDs for all songs corresponding to these K points in the Deezer dataset.

[4] https://scikit-learn.org/stable/modules/neighbors.html.

3.2 Stage 2: Distance Metrics

Given the Deezer song IDs for candidate points, we query an audio dataset for these songs' features. We then use Eq. (1) to calculate a target audio feature vector, and choose the song which minimizes a specified distance score to the target. Evaluating these K neighbors along D audio features has a time complexity of $\mathcal{O}[KD]$, which makes this more efficient than the KNN-based method of the first stage. This two-stage loop repeats to find the following songs in the path from the first to the final song, until the playlist reaches the desired length.

4 Evaluation Metrics

We seek to measure the quality of playlists as paths in both the Valence-Arousal space, and a space of audio feature vectors. For each of these spaces, we evaluate playlists on two key priorities: (1) *Smoothness* and (2) *Evenness*.

First, a playlist should be *smooth* such that the songs fall along a linear path from the first to the final song. This ensures that the emotional and audio characteristics consistently change in the same direction. To evaluate this linear smoothness, we use the Pearson Correlation Coefficient (PCC), a method used to evaluate smoothing techniques [29] which measures how well two variables A and B fall on a straight line (see Eq. (2)). For emotional smoothness, we calculate the PCC score between Valence and Arousal to see how well our algorithm creates a line in these two dimensions. For audio feature spaces, we calculate the average PCC score of all pairs of features using a correlation matrix. A score of 1.0 implies a perfectly smooth playlist as a line from the first to the final song.

$$\rho_{AB} = \left| \frac{\mathrm{Cov}(A, B)}{\sigma_A \sigma_B} \right| \tag{2}$$

Second, a playlist should be *evenly spaced* such that each transition between songs should be an equivalent step in an emotional or audio feature space. Uneven playlists could startle users with a sequence of similar songs followed by one with very different characteristics. We measure evenness as the variance of a playlist's step sizes, using the Root Mean-Square Error (RMSE) to calculate the distance between each adjacent pair of songs S_i and S_{i+1} over N features, as shown in Eq. (3). In a perfectly even playlist, the size of each step from one song to the next should be the same, and the variance of their RMSEs should be zero.

$$RMSE_i = \sqrt{\frac{||S_{i+1} - S_i||_2}{N}} \tag{3}$$

5 Experiments and Results

We examine our playlist algorithm's behavior against a wide variety of parameters. Below, we discuss our tests on the effectiveness of various distance metrics, audio datasets and K values. When we experimentally vary one parameter, we

use default values for the others to ensure consistency. We use $K = 7$ for our algorithm's first stage, Euclidean distance for Stage 2, and audio features from both Spotify and the MSD. We generate 12-song playlists across all tests in this study, a choice informed by a 2014 survey which found that three datasets of playlists from popular sources contained 11.6 tracks on average [4].

Since these playlists are paths in the Deezer dataset's unevenly distributed Valence-Arousal space, we generate playlists to explore its different regions. We sample 100 random songs in each quadrant of the Valence-Arousal space (see Fig. 3), and for each permutation of two quadrants, we make playlists that start at each point in the first and end at each point in the second. Across the 12 quadrant pairs, we generate 120,000 playlists for each parameter we test.

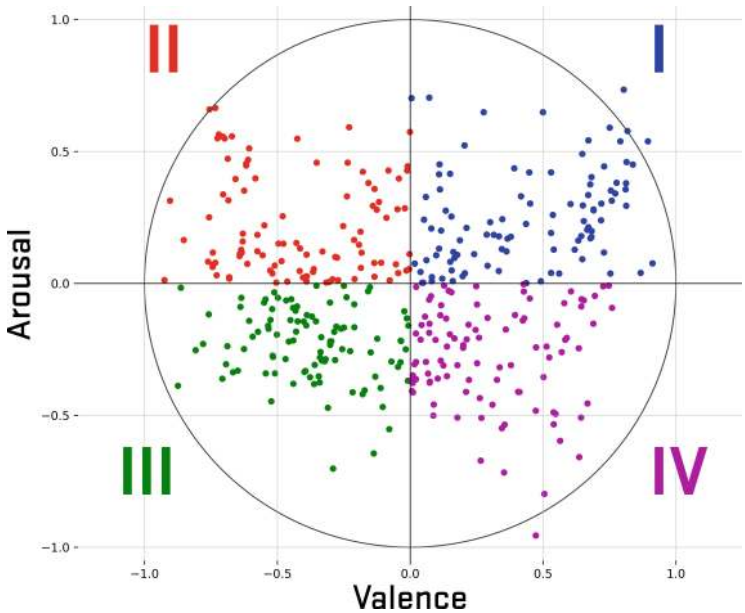


Fig. 3. Map of 100 sample songs per labeled quadrant in the Deezer 2018 dataset [7]. All playlists in the following experiments start from one region and travel to another.

5.1 Distance Metrics

We first test the effects of different common distance metrics in our algorithm's second stage, examining their smoothness and evenness across audio and emotion spaces. Previous studies in mood-dynamic playlist generation [9] only examine the smoothness of a few random playlists built on various distance metrics. Here, we examine the merits in comparing candidate songs with the Euclidean, Manhattan, Cosine, and Jaccard distances, against a simple baseline method which randomly chooses a single candidate song during the second stage.

When testing across regions, we find that the choice of distance metric for the audio features of second stage has no effect on a playlist's smoothness or evenness in the Valence-Arousal space. This indicates that all the candidates chosen in the first stage enable strong transitions emotionally, which allows us to optimize travel on audio feature spaces with no trade-off in emotional transitions.

Table 3. Audio-based scores of playlists by distance, measuring smoothness and evenness across the combined Spotify and MSD audio feature space. The Euclidean and Manhattan distances create the most smooth and even playlists.

Distance Metric	Pearson Correlation			Step-Size Variance		
	Mean	**St.Dev.**	**Median**	**Mean**	**St.Dev.**	**Median**
Euclidean	0.2820	0.0308	0.2789	0.0088	0.0041	0.0081
Manhattan	0.2826	0.0312	0.2795	0.0099	0.0045	0.0091
Cosine	0.2755	0.0283	0.2730	0.0234	0.0105	0.0218
Jaccard	0.2711	0.0348	0.2653	0.0243	0.0116	0.0224
Random	0.2618	0.0256	0.2597	0.0223	0.0099	0.0208

Table 3 shows that employing Euclidean and Manhattan distances in our algorithm's second stage generates the smoothest and most evenly-spaced playlists when considering audio transitions. Given that they are used as the L1 (Manhattan) and L2 (Euclidean) vector norms, it is perhaps unsurprising that they find the most direct distance to a target point in an ideal line. Using the Cosine and Jaccard distances yields smoother playlists than the baseline, but fails to improve playlist evenness. This is likely because these two scores do not consider the length of the vector between the candidate and target points—only the angle between them (Cosine) or the minima and maxima of their values (Jaccard).

5.2 Audio Datasets

Next, we compare the effectiveness of using Spotify, MSD, and both sets of audio features in Stage 2. We also explore the impact of variance and dimensionality on these three datasets by applying a 12-feature PCA to each. To understand if including audio features improves upon mood-only playlists [9], we reuse the Deezer dataset's Valence-Arousal points in Stage 2 as a baseline method.

Employing only affective features for Stage 2 leads to the smoothest and most even playlists in mood. On average, playlists that use the Deezer features for both stages achieve a mood-based PCC of around 0.9336 and step-size variance of 0.0003. The choice in audio features shows no impact on playlist smoothness or evenness in emotional transitions, as all audio datasets yield an average mood-based PCC of 0.8734 and step-size variance of 0.0010. This discrepancy between purely emotional data and audio data is to be expected: utilizing affective features means that Stage 2 chooses the song closest to a target based on emotional, instead of acoustic, similarity, yielding the best emotional transitions.

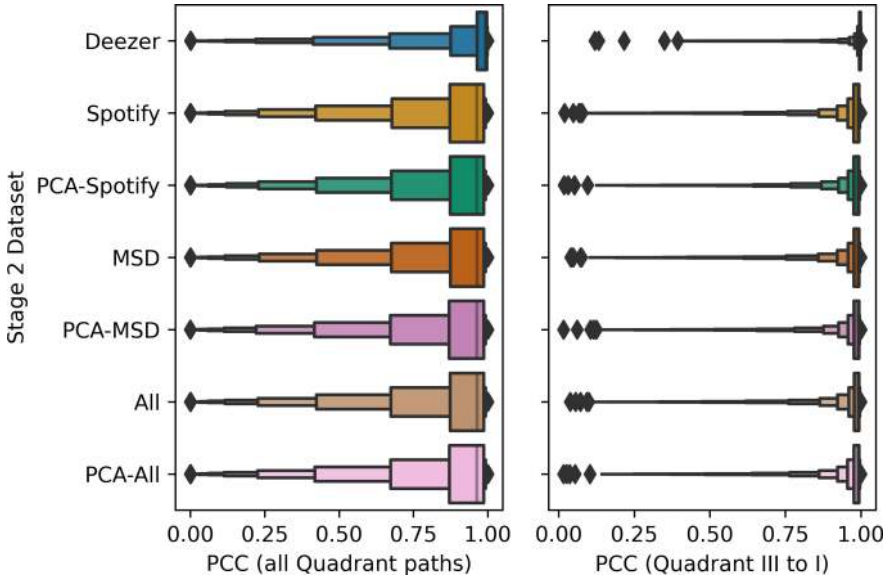


Fig. 4. Smoothness in the Valence-Arousal space of playlists by dataset. The left plot includes all playlists, and the right shows only playlists that travel from the sad quadrant (III) to the happy region (I). Most playlists in both plots have a very high PCC score, regardless of dataset (*Deezer* indicates reusing emotional features for Stage 2).

However, Fig. 4 reveals two other findings. First, the distributions of all of the box-plots indicates that the vast majority of playlists generated with audio-based features still transition smoothly in mood, and those that do not are outliers. Second, the box-plot on the right-hand side visualizes only the playlists that start in Quadrant III (sad and depressed songs) and end in Quadrant I (happy and excited songs)−a diagonal path through two dense regions of the Deezer dataset, as visualized in Fig. 1. The difference in smoothness between using emotional and acoustic features for Stage 2 is far smaller for playlists with this path than with other trajectories. The availability of Valence-Arousal data along this path may improve emotional transitions, as playlists generated from one dense region of this space to another yielded the smoothest playlists.

We evaluate transitions in the MSD and Spotify feature spaces separately in this experiment to better examine the relationships between audio feature sets. Table 4 indicates that in the 78-feature MSD space, no dataset yields significant improvements in smoothness. However, Fig. 5 visualizes that both the original 15 Spotify features and those transformed with PCA yield the smoothest playlists when evaluating against its feature space. This implies that the availability of a large number of acoustic features may in fact limit potential smoothness, because the algorithm must ensure a linear path for far more dimensions.

Table 4. Smoothness (PCC scores) and evenness (step-size variances) of playlists based on dataset, scored in the MSD audio feature space. All datasets yield similarly smooth playlists, and, besides results using the MSD alone, similarly uneven playlists.

Dataset (Stage 2)	Pearson Correlation			Step-Size Variance		
	Mean	**St.Dev.**	**Median**	**Mean**	**St.Dev.**	**Median**
Mood Only						
Deezer	0.2999	0.0171	0.2982	0.0063	0.0029	0.0058
Audio Datasets						
Spotify	0.2966	0.0174	0.2943	0.0063	0.0030	0.0058
MSD	0.2995	0.0208	0.2959	0.0031	0.0014	0.0028
All (Spotify + MSD)	0.2981	0.0203	0.2947	0.0034	0.0015	0.0031
Audio (with PCA)						
Spotify	0.2963	0.0167	0.2943	0.0062	0.0028	0.0057
MSD	0.3063	0.0197	0.3040	0.0055	0.0025	0.0051
All (Spotify + MSD)	0.3050	0.0195	0.3026	0.0055	0.0025	0.0051

Table 4 shows that only the original MSD-based feature sets generate playlists with greater evenness when evaluated in the MSD space. Applying PCA to the MSD features led to a greater playlist step-size variance, and Fig. 5 visualizes the same effect in the Spotify feature space. From this, and from the lack of improvement in smoothness, we conclude that PCA discards key information about song points and do not recommend its use on our algorithm's audio features.

However, Table 4 also indicates no significant difference between employing emotional or acoustic features in Stage 2 for audio-based smoothness or evenness. The first stage of our algorithm chooses candidate songs only based on emotion, which may only present candidates with suboptimal audio-based transitions. This set of candidates may mean that the second stage has very limited ability to improve acoustic smoothness or evenness, resulting in the low PCC scores seen in Table 4. However, the inconclusivity of our findings underscores the need for further exploration of the balance between emotional and acoustic transitions and large high-quality datasets of music annotated with emotion.

5.3 K Values

Finally, we evaluate different numbers for K, or the number of candidate points our algorithm extracts from the Deezer dataset. We compare the effectiveness of extracting $K = 3, 7, 11, 15, 19, 23, 27, 31$ songs in Stage 1 from our KNN model. In Figs. 6 and 7, we see that as K increases, we generate playlists with less smooth and even transitions in the Valence-Arousal space. However, as this change occurs, our algorithm generates playlists that are more smooth and even in acoustic transitions. While the trend in these figures appears when the playlist length is 12 songs, we observe similar results in playlists containing 5 to 20 songs.

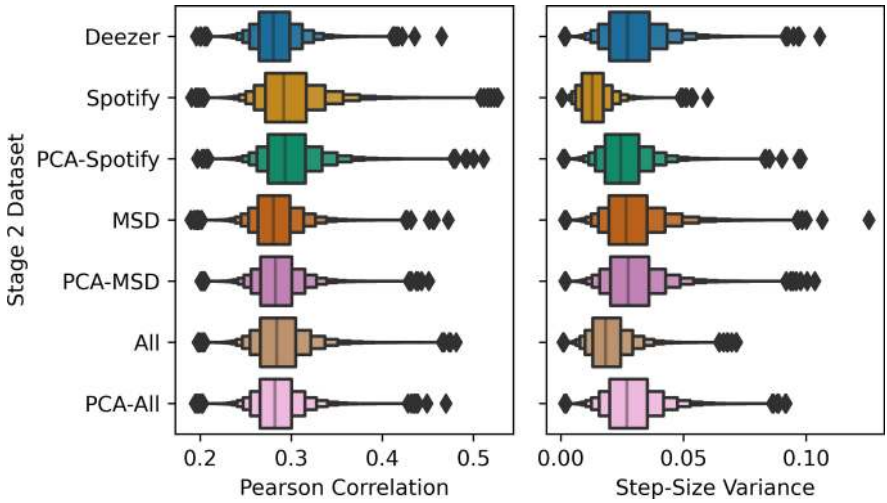


Fig. 5. Smoothness (left) and evenness (right) scores of playlists based on dataset, scored in the Spotify feature space, visualizing PCA's impact on playlist quality.

When recalling fewer candidates, we encourage more precise transitions in the Valence-Arousal space at the expense of acoustic performance. By recalling more songs, we explore more candidates in Stage 2, where songs less than ideal emotionally may lie closest to the ideal audio feature path. For example, when $K = 19$, the algorithm chooses the 19 closest points to the target in the Valence-Arousal space during Stage 1. The song at the 19th closest point will be the farthest candidate from the target point emotionally. However, in Stage 2, this song may be the closest to the target when considering audio features. If so, it will become the next song in the playlist because of this audio similarity even though its emotional transition may be suboptimal. Developers could use K to balance the performance of playlists between emotional and audio feature spaces.

However, it should be emphasized that this behavior is exacerbated by the number of points in our dataset. With only 2,762 unique points, the Deezer dataset is dwarfed by the millions of songs available on streaming services. Within a normalized range of song features at $[-1, 1]$ processed in a similar way to our approach, these songs will yield a much greater density of points in the Valence-Arousal space for our algorithm's first stage. As a result, when $K = 19$, the 19th closest point in the streaming service's data will likely be much closer to the target point than such a point in our current dataset. Therefore, the emotional transformation will be only slightly suboptimal if the 19th closest song in the Valence-Arousal space is chosen compared to the closest song. This could encourage developers to use larger values of K than we used in our experiments to maintain similar smoothness and evenness in emotion spaces.

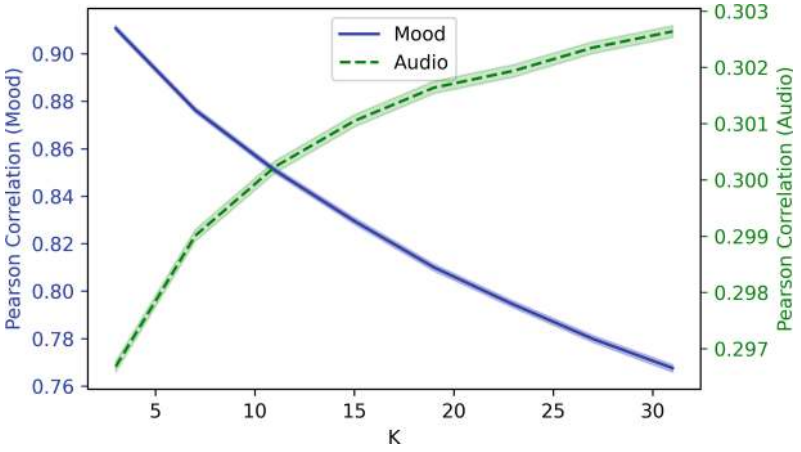


Fig. 6. Smoothness (PCC score) of 12-song playlists by K values, visualizing an inverse relationship between affective and audio smoothness as K changes.

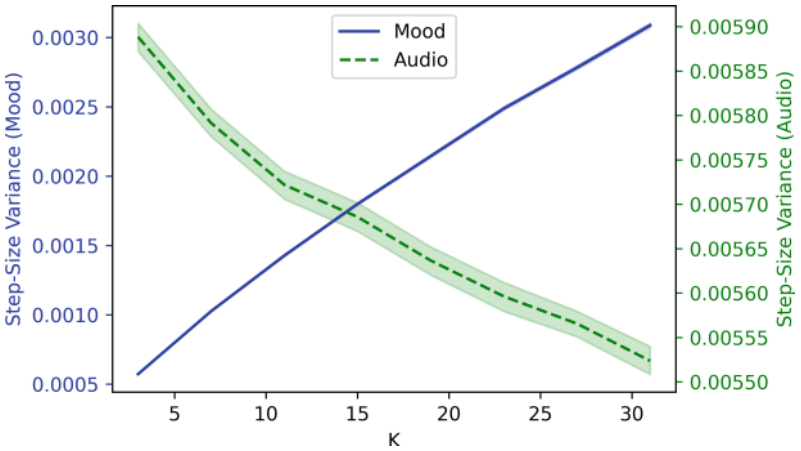


Fig. 7. Evenness (step-size variance) of 12-song playlists by K values, visualizing an inverse relationship between affective and audio smoothness as K changes.

6 Conclusions

In this work, we demonstrate an algorithm which generates playlists that gradually travel through emotion and audio feature spaces. Our approach creates a smooth and linear path through vector spaces of song features, and thus requires no data on existing playlists or users. This can enable applications to create high-quality playlists for users without potential privacy issues. We also define a testing methodology for dynamic playlists which covers many regions of a feature space and evaluates playlists on their smoothness and evenness.

We find that audio features make playlists less smooth and even in an emotional space than considering only affect alone, and that their impact on playlist travel through acoustic similarity is inconclusive. However, the balance between these two feature spaces can be tuned easily by adjusting the recall of our algorithm's first stage. Employing dimensionality reduction algorithms such as PCA fail to yield significantly smoother playlists and come at the cost of consistent steps between songs, since they lose key information regarding a song's acoustics.

Our approach is primarily limited by the quantity and quality of music emotion datasets. While the Deezer 2018 dataset [7] is the largest public repository of songs and their Valence-Arousal scores, their synthetic annotation process leads to many duplicate emotional points. Additionally, their uneven distribution in this plane may potentially yield jarring playlists when traveling through a sparse region in the dataset space. Unfortunately, datasets that contain high quality human-collected emotional annotations contain far fewer songs.

This playlist algorithm presents plenty of opportunities for future work. We create promising measures of a playlist's emotional change, but evaluating a change in mood is highly subjective. As a next step, we plan to test this algorithm's effectiveness with a sample of human listeners, with a study that (1) asks users to report their initial mood, (2) creates a playlist to guide them towards a desired state, and (3) requests them to report their mood after they listen to each song. By measuring a listener's emotion during and after the playlist listening experience, we can assess the quality of the algorithm, and of existing datasets in modeling emotional and acoustic qualities relevant for listeners.

Due to the Deezer dataset's limitations, we currently only examine data at the full song level. However, emotions often change within a song. Perhaps through examination of emotional and audio travel within a song's segments, and the transitions between the end of a song to the beginning of the next, we may create an even smoother listening experience for the user.

Researchers building on our approach could explore integrating this algorithm into a human-in-the-loop system that incorporates sensor readings [16], sentiment analysis of writing [22], or foreground computer processes [32] to gather a user's current mood and continuously recommend songs that travel towards a desired emotional state. However, for these types of systems, it is important to maintain transparency with the listener and only operate with their informed consent, since these playlists are designed to change their mood.

Several studies show that music has a positive impact in elementary school classrooms, encouraging positive and productive behavior among children [34], helping them stay relaxed [11], and even as a means for therapy for children with autism or dyslexia [21]. Our algorithm could be extended as a novel therapeutic tool, enabling the generation of dynamic playlists that begin at a listener's current mood and simultaneously consider their preferred music genres to slowly nudge them towards a desired state of excitement, relaxation or focus.

References

1. Allik, A., Thalmann, F., Sandler, M.: MusicLynx: exploring music through artist similarity graphs. In: Companion Proceedings of the The Web Conference 2018. WWW 2018, pp. 167–170. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2018). https://doi.org/10.1145/3184558.3186970
2. Bertin-Mahieux, T., Ellis, D.P.W., Whitman, B., Lamere, P.: The million song dataset. In: Klapuri, A., Leider, C. (eds.) Proceedings of the 12th International Conference on Music Information Retrieval. ISMIR 2011, pp. 591–596. International Society for Music Information Retrieval, Miami (2011). https://doi.org/10.5281/zenodo.1415820
3. Blood, D.J., Ferriss, S.J.: Effects of background music on anxiety, satisfaction with communication, and productivity. Psychol. Rep. **72**(1), 171–177 (1993). https://doi.org/10.2466/pr0.1993.72.1.171
4. Bonnin, G., Jannach, D.: Automated generation of music playlists: survey and experiments. ACM Comput. Surv. **47**(2), 1–35 (2014). https://doi.org/10.1145/2652481
5. Cardoso, L., Panda, R., Paiva, R.P.: MOODetector: a prototype software tool for mood-based playlist generation. In: Proceedings of the 2nd Portuguese National Symposium on Informatics. INForum 2011, Coimbra, Portugal (2011)
6. Chen, S., Moore, J.L., Turnbull, D., Joachims, T.: Playlist prediction via metric embedding. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2012, pp. 714–722. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2339530.2339643
7. Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J., Moussallam, M.: Music mood detection based on audio and lyrics with deep neural net. In: Gómez, E., Hu, X., Humphrey, E., Benetos, E. (eds.) Proceedings of the 19th International Conference on Music Information Retrieval. ISMIR 2018, pp. 370–375. International Society for Music Information Retrieval, Paris (2018). https://doi.org/10.5281/zenodo.1492427
8. Deng, J.J., Leung, C.: Emotion-based music recommendation using audio features and user playlist. In: 6th International Conference on New Trends in Information Science, Service Science and Data Mining. ISSDM 2012, pp. 796–801. Institute of Electrical and Electronics Engineers, Taipei (2012)
9. Donnelly, P., Gaur, S.: Mood Dynamic Playlist: interpolating a musical path between emotions using a KNN algorithm. In: Ahram, T., Taiar, R. (eds.) Proceedings of the 9th International Conference on Human Interaction & Emerging Technologies: Artificial Intelligence & Future Applications. IHIET-AI 2022, vol. 23. AHFE Open Access (2022). https://doi.org/10.54941/ahfe100894
10. Donohoe, R., McNeely, T.: The effect of student music choice on writing productivity. Technical report ED448472, US Department of Education (1999)
11. Feldman, S.: Music affects productivity. Manag. Rev. **80**(7), 6–7 (1991)
12. Flexer, A., Schnitzer, D., Gasser, M., Widmer, G.: Playlist generation using start and end songs. In: Bello, J.P., Chew, E., Turnbull, D. (eds.) Proceedings of the 9th International Conference on Music Information Retrieval. ISMIR 2008, pp. 173–178. International Society of Music Information Retrieval, Philadelphia (2008). https://doi.org/10.5281/zenodo.1418272

13. Flint, M.: The effects of music on physical productivity. Ph.D. thesis, The Ohio State University (2010)
14. Fox, J., Embrey, E.: Music – an aid to productivity. Appl. Ergon. **3**(4), 202–205 (1972). https://doi.org/10.1016/0003-6870(72)90101-9
15. Kellogg, R.T.: Writing habits and productivity in technical writing. Technical report ED249502, US Department of Education (1982)
16. Kim, H.G., Kim, G.Y., Kim, J.Y.: Music recommendation system using human activity recognition from accelerometer data. Trans. Consum. Electron. **65**(3), 349–358 (2019). https://doi.org/10.1109/TCE.2019.2924177
17. Lesiuk, T.: The effect of music listening on work performance. Psychol. Music **33**(2), 173–191 (2005). https://doi.org/10.1177/0305735605050650
18. Liebman, E., Saar-Tsechansky, M., Stone, P.: DJ-MC: a reinforcement-learning agent for music playlist recommendation. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. AAMAS 2015, pp. 591–599. International Foundation for Autonomous Agents and Multiagent Systems, Istanbul (2015). https://doi.org/10.5555/2772879.2772954
19. Ludewig, M., Kamehkhosh, I., Landia, N., Jannach, D.: Effective nearest-neighbor music recommendations. In: Proceedings of the ACM Recommender Systems Challenge 2018. RecSys Challenge 2018. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3267471.3267474
20. Marimont, R.B., Sharpiro, M.B.: Nearest neighbour searches and the curse of dimensionality. IMA J. Appl. Math. **24**(1), 59–70 (1979). https://doi.org/10.1093/imamat/24.1.59
21. Mayer-Benarous, H., Benarous, X., Vonthron, F., Cohen, D.: Music therapy for children with autistic spectrum disorder and/or other neurodevelopmental disorders: a systematic review. Front. Psychiatry **12**, 643234 (2021). https://doi.org/10.3389/fpsyt.2021.643234
22. Meyers, O.C.: A mood-based music classification and exploration system. Master's thesis, Massachusetts Institute of Technology, Cambridge, USA (2007)
23. Moscato, V., Picariello, A., Sperlí, G.: An emotional recommender system for music. IEEE Intell. Syst. **36**(5), 57–68 (2021). https://doi.org/10.1109/MIS.2020.3026000
24. Nathan, K.S., Arun, M., Kannan, M.S.: EMOSIC - an emotion based music player for Android. In: Proceedings of the 17th IEEE International Symposium on Signal Processing and Information Technology. ISSPIT 2017, pp. 371–276. Institute of Electrical and Electronics Engineers, Bilbao (2017). https://doi.org/10.1109/ISSPIT.2017.8388671
25. Pampalk, E., Pohle, T., Widmer, G.: Dynamic playlist generation based on skipping behavior. In: Proceedings of the 6th International Conference on Music Information Retrieval. ISMIR 2005, pp. 634–637. International Society of Music Information Retrieval, Victoria (2005). https://doi.org/10.5281/zenodo.1414932
26. Panda, R., Redinho, H., Gonçalves, C., Malheiro, R., Paiva, R.P.: How does the Spotify API compare to the music emotion recognition state-of-the-art? In: Mauro, D.A., Spagnol, S., Valle, A. (eds.) Proceedings of the 18th Sound and Music Computing Conference. SMC 2021. Axea sas/SMC Network, Virtual, pp. 238–245 (2021). https://doi.org/10.5281/zenodo.5045100
27. Pauws, S., Verhaegh, W., Vossen, M.: Music playlist generation by adapted simulated annealing. Inf. Sci. **178**(3), 647–662 (2008). https://doi.org/10.1016/j.ins.2007.08.019
28. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011). https://doi.org/10.48550/arXiv.1201.0490

29. Rosen, P., Quadri, G.J.: LineSmooth: an analytical framework for evaluating the effectiveness of smoothing techniques on line charts. IEEE Trans. Vis. Comput. Graph. **27**(2), 1536–1546 (2021). https://doi.org/10.1109/TVCG.2020.3030421
30. Russell, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. **39**(6), 1161–1178 (1980). https://doi.org/10.1037/h0077714
31. Sakurai, K., Togo, R., Ogawa, T., Haseyama, M.: Music playlist generation based on graph exploration using reinforcement learning. In: Proceedings of the IEEE 3rd Global Conference on Life Sciences and Technologies. LifeTech 2021, pp. 53–54. Institute of Electrical and Electronics Engineers, Nara (2021). https://doi.org/10.1109/LifeTech52111.2021.9391870
32. Sen, A., Popat, D., Shah, H., Kuwor, P., Johri, E.: Music playlist generation using facial expression analysis and task extraction. Annales Universitatis Mariae Curie-Sklodowska, sectio AI - Informatica **16**(2), 1–6 (2017). https://doi.org/10.17951/ai.2016.16.2.1
33. Song, Y., Dixon, S., Pearce, M.T., Halpern, A.R.: Perceived and induced emotion responses to popular music: categorical and dimensional models. Music. Percept. **33**(4), 472–492 (2016). https://doi.org/10.1525/mp.2016.33.4.472
34. White, K.N.: The effects of background music in the classroom on the productivity, motivation, and behavior of fourth grade students. Master's thesis, Columbia College, Columbia, SC, USA (2007)
35. Yeo, I.K., Johnson, R.A.: A new family of power transformations to improve normality or symmetry. Biometrika **87**(4), 954–959 (2000). https://doi.org/10.1093/biomet/87.4.954
36. Zamani, H., Schedl, M., Lamere, P., Chen, C.W.: An analysis of approaches taken in the ACM RecSys Challenge 2018 for automatic music playlist continuation. ACM Trans. Intell. Syst. Technol. **10**(5), 1–21 (2019). https://doi.org/10.1145/3344257