Music Reccomendation System for DJ Sets - Progress Report

Harvey Prewer

BMus/BSc Music & Sound Recording (Tonmeister)
Technical Project

Institute of Sound Recording
University of Surrey

January 2023

0.1 Introduction

Few options exist for algorithmically finding song recommendations for DJ sets. Building a DJ set by manually searching for songs is very time consuming, so improved, automated, recommendation systems would make this process more efficient.

The music recommendation application would cater for professional and aspiring DJs, and music lovers drawn to radio and DJ sets. The application will take a DJ set identifier as input and output song recommendations. The recommendation algorithm will use collaborative filtering or content based filtering, with MixesDB [1] as the source dataset. It will be implemented as a python console application.

0.2 Potential Methods

0.2.1 Collaborative-Based Filtering

1. Input a DJ Setlist to a matrix with the number of plays a song has by a given DJ.

2. The program looks over DJs who played the songs included in the set, removing from the MixesDB data any tracks that are already in the input DJ set.

3. Added weight on DJs who've played the most significant majority of songs on the mix, allowing other songs played by them to make up the majority of the recommendations.

0.2.2 Content-Based Filtering

To implement Content Based Filtering, you would need to:

1. Profile the features of all the tracks in MixesDB. Some of these features would need to come from Spotify, as per the medium article (Chow 2020).

2. For each track, the features would be converted into a vector.

3. Take the input DJ Set, that you're recommending for and repeat the same feature analysis as above. The result would be a vector track, which could then be summarised into a single vector for the whole set.

0.3 Motivation

The hypothesis is that the application will give song recommendations that are both stylistically suitable but aren't necessarily popular (a common trait in most recommendation systems (Lesota et al. 2021)), because of the characteristics of a DJ set. For example, a typical quality of DJ sets is the great lengths the DJ goes to find obscure and unknown songs. Therefore, not prioritising popularity is a desirable trait in a recommendation system for a DJ's music consumption and knowledge, usually being more than the average consumer.

---

[1] https://www.mixesdb.com/w/Main_Page

A DJ set can also be described as musical recommendations curated by a person passionate enough about music to make selecting and playing songs their profession or a well-invested hobby. I want to explore if creating a music recommendation system that builds from archives of these personally selected songs would create a pool of more "human" suggestions than what Spotify or other streaming platforms recommends. Another benefit is that what a DJ plays is not limited to a specific format, so my application suggests music on all platforms.

0.4 Research Questions

0.4.1 What are the effective methods of automating the recommendation for suitable songs added to a given DJ set?

I see the need for this application due to how laborious it is to look for songs from DJ sets you like. Discovering new music from DJ sets includes endlessly digging through the DJs, Artists, Labels and Genres associated with the mix, which is very time-consuming.

The lack of recommendation systems for DJ sets makes this application more noteworthy. Besides manually finding songs, I have discovered two inadequate ways of getting recommendations based on DJ sets:

1. **Spotify** - Find a DJ set on Spotify (not commonly available but labels like !K7 Music releases DJ sets commercially as albums), add that to a playlist, and look at the" Recommended based on the playlist". The main problem with this is that DJ sets on Spotify are few and far between. One can only input a mix one's heard from the radio or a recorded live performance by finding the tracklist and adding the songs into a playlist. The system also finds what other users are listening to, often giving you song recommendations that are usually quite popular and well-known. Spotify has a vast library of music, but a sizeable chunk of music played on Spotify you won't find on DJ sets.

2. **SoundCloud** - When a DJ set gets uploaded to SoundCloud, it usually recommends other DJ sets on its " radio " when getting recommended songs is more desirable. The recommendation system is only limited to what is available on SoundCloud.

0.4.2 Is this proposed method a suitable solution to the above question?

The use of collaborative filtering is used a lot in recommendation systems amongst different formats. It first came to prevalence by the winner of the Netflix Prize in 2007 and 2008 using the technique extensively(Zhou et al. 2008). In a 2020 medium article, a song recommendation system for DJs used collaborative filtering to find valid suggestions(Chow 2020).

The advantage content-based filtering has over collaborative filtering is its ability to recommend new and unpopular music, which as mentioned before is desirable for DJs(van den Oord et al. 2013).

0.4.3   Can this application be further developed to solve other DJ tasks?

The application could be developed to aid aspiring DJs in showing songs suitable to mix in and out of songs they already know. The input is a song, and the output is the songs played before or after in other DJ sets.

0.5   Potential Issues

A problem will arise when multiple DJs are involved in a DJ set. To solve this, I'll label this DJ set and its track selection as being from both DJs, and this is valid because the fact they are DJing together shows some crossover in taste.

Another limitation is that it will only recommend songs from MixesDB. This small dataset may affect the quality of some suggestions.

Another issue is how well the level of cohesiveness matches the suggestions. For example, if a set plays a variety of genres, will the pool of recommendations reach that level of diversity? The application should pick DJs that match the input set's cohesion level. If not, the programme could solve this by capping the number of song suggestions from specific DJs. For example, suppose the input DJ set is House, Techno and Rock songs, but the DJ themselves usually just DJs dance music. In that case, this cap will limit the suggestion pool from House and Techno and, hopefully, let a suitable amount of Rock suggestions be selected.

0.6   Summary of Literature Review

The literature review will answer the first and last questions in the literature review.

The literature review will explore the best methods used for state-of-the-art music song recommendation systems. This chapter will include mentioning collaborative filtering (Chow 2020), content-based filtering (Chang 2021) and clustering(Piyush 2019). This part will also discuss the need to use deep learning to expand a recommendation system beyond its database. It will then discuss how a DJ's taste is diverse and diverse listeners are more refrained from using less algorithmic means to find music (Anderson et al. 2020).

For the final one, the chapter will discuss other machine learning-aided DJ technologies. An example of this is the source separation stuff found (Kirn 2023), and it will discuss whether my proposed method can contribute to the problem the other applications are fixing. Then, it will relate to how my approach can be helpful for other DJ-related necessities.

The readings here will determine the answer to the second research question.

0.7 Time Plan

	Due Date	8 January	15 January	22 January	29 January	5 February	12 February	19 February	26 February	5 March	7 March
Report											DEADLINE
Finish progress report		■									
Literature review			■	■	■						
Experiment						■	■				
Results, discussion, refractoring								■	■	■	
Code											
Spiking				■							
Dataset maintainance				■							
First vertical slice					■						
Second vertical slice						■	■				
Code complete, analyse results								■	■		

0.8 Outline Experiment Plan

Make a python console application that uses collaborative filtering and MixesdB as the source dataset. The input will be a text file of a DJ set you enjoy and the program outputs a text file listing songs similar to the input DJ set.

For testing there is got two methods:

1. First is using spectrogram comparisons of different methods. Spectrograms of the songs in the set list and the recommended songs will be taken. Within the recommended songs it will compare itself amongst the songs in the DJ set concluding which one it's most similar to. Then there will be a discussion about the overall similarity rating of each song and how well distributed it is amongst the inputted DJ set.

2. After this, backtesting will be used. This is when the songs of a said year is inputted, and looked at the recommendations it gives, and it will then compare the recommendations to the songs that were played in DJ sets the year after. This will be a good indication of whether the songs recommended are ones that a DJ would go out and play.

0.9 Chapter Layout of Final Report

1. Chapter 1 Introduction

2. Chapter 2 - Machine Learning used in Music Recommendation Systems

    (a) The need for Recommendation Systems

    (b) Collaborative Filtering

    (c) Content Based Filtering

    (d) Clustering

    (e) Convoluted Networks

3. Chapter 3 - Machine Learning used for DJing

    (a) Source Separation

    (b) BPM, key, genre classification

References

Anderson, A., Maystre, L., Anderson, I., Mehrotra, R., and Lalmas, M. (2020): 'Algorithmic Effects on the Diversity of Consumption on Spotify', in *Proceedings of The Web Conference 2020*, Taipei Taiwan: ACM, pp.2155–2165, doi:10.1145/3366423.3380281.

Chang, E. (2021): 'Building a Song Recommendation System with Spotify', URL `https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7`.

Chow, D. (2020): 'Music Recommendation System for DJs Part 2', URL `https://ai.plainenglish.io/music-recommendation-system-for-djs-part-2-db14e7ece6ea`.

Kirn, P. (2023): 'In Review: Serato Stems · Feature RA', URL `https://ra.co/features/4105`.

Lesota, O., Melchiorre, A., Rekabsaz, N., Brandl, S., Kowald, D., Lex, E., and Schedl, M. (2021): 'Analyzing Item Popularity Bias of Music Recommender Systems: Are Different Genders Equally Affected?' in *Fifteenth ACM Conference on Recommender Systems*, Amsterdam Netherlands: ACM, pp.601–606, doi:10.1145/3460231.3478843.

Piyush (2019): 'Building Music Playlists Recommendation System', URL `https://towardsdatascience.com/building-music-playlists-recommendation-system-564a3e63ef64`.

van den Oord, A., Dieleman, S., and Schrauwen, B. (2013): 'Deep content-based music recommendation', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 26.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008): 'Large-Scale Parallel Collaborative Filtering for the Netflix Prize', in R. Fleischer and J. Xu, eds., *Algorithmic Aspects in Information and Management*, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, pp.337–348, doi:10.1007/978-3-540-68880-8_32.