Automated Playlist Generation

Kade Keith, *Student, Stanford University - Computer Science Department,*
Demetrios Fassois, *Student, Stanford University - Computer Science Department,*

Abstract—Our project generates music playlists based on a set of seed songs, using diverse features ranging from lyrical sentiment to song popularity. We approach the problem as both a graph problem and as a classification problem, and evaluate our results based on real human-curated playlists.

Index Terms—Playlist Generation, Song Recommendation, Sentiment Analysis

1 Motivation

With the growth of musical streaming services, there are now more songs than ever at music listeners fingertips. Because of this growth, the art of constructing playlists has become increasingly challenging, and discovering new music the in the expanse of choices is a daunting task. For this reason, we seek to build an automatic playlist generator, that can take a few songs as a seed set, and generate a complete playlist for the listener.

1.1 Goal

Using popular, human-curated playlists as our training data and test data, our system should construct playlists of similar quality. In particular, we plan to incorporate lyrical analysis in our model, as we believe that lyrical content is a big influence when creating playlists.

2 Method

We combine data from a number of sources in our project. The primary source is the Million Song Dataset [2] (MSD), and the corresponding lyrics dataset, which provided lyrics for roughly a quarter of those songs in a bag-of-words format. In addition to that we use Spotify [3] as the source of our playlist data, as well as using their's and Last.fm's [1] song info to augment the data from the MSD.

In total, that gives us the following the following group of attributes:

Feature	Value(s)	Source
Year	TODO-TODO	MSD, Spotify
Tempo	TODO-TODO	MSD
TODO	TODO	TODO
TODO	TODO	TODO
TODO	TODO	TODO
TODO	TODO	TODO

TODO REGARDING TIMBRE, TALK ABOUT HOW TO COMPARE TIMBRES

With these features, we approach the task two ways; first as a graph problem, and second as a classification problem.

2.1 Graph Problem

The first approach is to think of songs as nodes in a graph. With this you can apply k-nearest neighbors to find most similar songs given a seed or set of seeds. You can also think of a playlist as a path through this song graph. [4]

2.2 Classification

TODO

3 Evaluation Method

TODO - HOW WE SCORE RESULTS

4 Preliminary experiments

TODO

4.1 Graph Problem

TODO - KNN WITH DISTANCE FUNCTION

4.2 Classification

TODO - REGRESSION. PROLLY WITH SCIKIT LEARN

5 Results

5.1 Graph Problem

TODO - HOW'D IT DO?

5.2 Classification

TODO - HOW'D IT DO?

6 Next steps

TODO - WHAT'S NEXT

6.1 Graph Problem

TODO - EXPLORING PLAYLIST AS PATH IN GRAPH

6.2 Classification

TODO - NOT SURE

7 Contributions

TODO - WHO DID WHAT

References

[1] Last.fm. https://www.last.fm/.

[2] Million song dataset. https://labrosa.ee.columbia.edu/millionsong/.

[3] Spotify. https://www.spotify.com/.

[4] Masoud Alghoniemy and Ahmed H. Tewfik. A network flow model for playlist generation. In *In Proc IEEE Intl Conf Multimedia and Expo*, 2001.

Appendix A
EXAMPLE APPENDIX

The following is the output result of the script that tests for the end-game puzzles, looking for checkmate in 1, 2, or 3 move.

```
Random
Elapsed time is 0.0160000324249 seconds.
Baseline
Elapsed time is 0.294000148773 seconds.
minimax1
Elapsed time is 11.1240000725 seconds.
AB1Plus1
Elapsed time is 4.76499986649 seconds.
```