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 song or 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 music 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. A novel aspect of our project is that we incorporate lyrical analysis in our model, as we believe that lyrical content plays an important role when creating playlists.

2 Method

We combine data from a number of sources in our project. The primary source is the Million Song Dataset (MSD) [2], and the corresponding lyrics dataset, which provides lyrics for roughly a quarter of those songs in a bag-of-words format. In addition to that we use Spotify [5] 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 group of attributes:

Feature	Source
Year	MSD, Spotify
Tempo	MSD, Spotify
Timbre	MSD
Tags	Last.fm
Danceability	MSD, Spotify
Energy	MSD, Spotify
Loudness	Spotify
Popularity	Spotify
Speechiness	Spotify
Acousticness	Spotify
Instrumentalness	Spotify
Liveness	Spotify
Valence	Spotify

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. The cluster of songs geometrically close together form a playlist.

2.2 Classification

The second is to think of deciding whether or not a song belongs on a playlist as a classification problem. Positive training examples are a subset of songs on the playlist. Negative examples are a random selection of songs not on the playlist. Then, presented with a previously unseen song, the model classifies it as either belonging on the playlist or not.

3 Evaluation Method

We plan to evaluate our two models with two different metrics. For the graph-based model, we can measure the distance from our proposed playlist songs to the actual songs on the playlist. In the best case that distance will be zero because the songs are the same. For the classification model, we present our model with previously unseen songs from the playlist we trained it on, and see if it correctly classifies those songs. In addition, we take a selection of songs that are not on the playlist and see if the model correctly classifies those. TODO TALK ABOUT CROSS VALIDATION.

4 Preliminary experiments

4.1 Data Gathering

The bulk of our work thus far has been in data collection and processing. We have set up the complete pipeline for our model using a small subset of the MSD. We first take the MSD and filter out the songs for which we do not have lyrics data. We also remove the fields that we are not considering, such as 'sections', and the fields for which the data is too sparse to be useful, such as artist location (latitude and longitude). Doing this filtering and elimination reduces the size on disk by almost a factor of 10, which is very important considering the whole dataset is 280 GB.

For the remaining songs, we perform preliminary sentiment analysis. As a baseline we are just using Naive Bayes (with the NLTK movie review corpus as training data [3]) to score each song as either positive or negative, which then gets included in the features. After that, we enhance our data with audio features from Spotify and Last.fm.

For gathering playlists, we rely on searching for a particular term, such as "summer" or "love", and saving the top playlists for that search. We have automated this process so that all we need to input is the term. Deciding which terms to search for provides an interesting sub-problem. In general, we want to avoid "trending" or "hits" playlists, because even though those are popular, they are not cohesive. We focus instead on themes that we believe users will make cohesive playlists about, such as emotions or seasons.

Unfortunately the overlap of these two sets (the MSD with lyrics and the sampling of popular playlists) is not as high as we would like. Based on preliminary estimates, only 3-5% of the MSD songs with lyrics are showing up in our popular playlists. We attribute this largely to the fact that Spotify surfaces modern songs over older ones, and the MSD is a few years old. Since that would only give us a dataset on the order of a thousand, more gathering of lyrics data will be necessary. We plan to write scripts to automate this process. In addition, we will likely need to dig a little deeper to find more playlists that contain less modern songs.

4.2 Method: Graph Problem

The simplest graph approach is k nearest neighbors, which we have implemented. We represent each song as a point in 11-dimensional space according to our normalized features (all those listed above, excluding Timbre and Tags). Then we select the next songs for that playlist based on proximity to the seed. This is done repeatedly to construct the whole playlist.

4.3 Method: Classification

As a classification baseline, we rely on SciKit Learn's [4] linear regression implementation. For now we have hand-picked a subset of the MSD and fed that in as our playlist.

5 Results

Of the MSD subset, roughly 25% of the songs had lyrics, giving us 2,500 songs to choose from in our preliminary experiments. The subset was chosen at random from the whole dataset. One issue as you can see though is an imbalance in the dataset in terms of genre. There are far more Rock and Pop tracks than their are Hip Hop or Country. This manifests in our algorithm struggling when it comes to the less-well represented genres.

5.1 Graph Problem

We present a couple example playlists given their seed song.

```
Seed: Aerosmith, Milkcow Blues
KNN playlist:
Aiden, She Will Love You (Album Version)
Vixen, American Dream
Gary Moore, All Your Love [Live 1999]
Snow Patrol, Half The Fun
```

We weren't familiar with many of these bands, but after giving them a listen can confirm they are all rock tracks with the signature distorted guitars and clashing cymbals akin to their seed.

Now when given a Hip Hop seed, this approach produced a laughably diverse playlist.

```
Seed: RUN-DMC, Can I Get A Witness
KNN playlist:
Daniel Johnston, Story Of An Artist (Don't Be Scared)
Roy Brown, Good Rocking Tonight
Michelle Tumes, Christe Eleison (Christ Have Mercy)
De La Ghetto, Es dificil
```

In this list we have a lo-fi singer-songwriter, a classic R&B artist, Contemporary Christian musician, and a Reggaeton artist. We believe that Hip Hop is a genre that serves to benefit a lot from incorporating further lyrical analysis in our model.

5.2 Classification

As an example of the classification approach, we train ($y = 1$) the model on all of the songs of a particular artist (first Aerosmith and then Bon Jovi), excluding one song. For our negative examples ($y = 0$), we select a random subset of the remaining songs. Unfortunately since we are just using the small subset, this means we only have 6/7 positive examples. Shown below are some results.

```
Seed: 7 Aerosmith Songs
Scores:
Winds Of Plague, Origins And Endings:
0.386695109939
The Ataris, Make It Last:
0.44327217148
Sweet, Neon Psychedelia:
0.948058795077
Funeral For A Friend, Your Revolution Is A Joke:
0.0841323790295
The Black Crowes, Good Morning Captain:
0.299697761404
Aerosmith, Reefer Head Woman:
1.47295049055
Neil Diamond, Brooklyn On A Saturday Night:
1.07268129368
OV7, Volvere:
−0.399370795516
Tha Liks, Da Da Da Da:
−0.339378432386
```

As expected, the remaining Aerosmith song is highly positive. Also the most negative songs, "Volvere" and "Da Da Da Da" are by a Mexican pop group and a Hip Hop trio, which are quite different, so those scores are as also expected.

```
Seed: 6 Bon Jovi Songs
Scores:
Johnny Horton, The Golden Rocket:
0.71933531467
Roger Miller, Husbands And Wives:
−0.0658498377106
Andy & Lucas, Hasta Los Huesos:
−0.10891481801
Hot Tuna, Hesitation Blues:
−0.346326437273
Olga Ta n, Como Olvidar (Merengue Versi n):
−0.101924816847
Nick Cave & The Bad Seeds, New Morning (Live):
0.357192781716
Bon Jovi, Only Lonely:
−0.258888975598
Frost, Take a Ride:
0.128365501962
Christina Aguilera, Cruz:
−0.66089836728
```

The Bon Jovi playlist did not fair as well. The remaining Bon Jovi song was given a surprising negative score. From looking at the audio features of the seeds and that song manually, nothing jumped out at us that explained the difference. The only positive note is that in general it seems like songs got more negative scores with this dataset, so it is likely a data problem.

6 Next steps

Besides what has already been mentioned (gathering more lyrics/playlists) we have a number of tasks remaining to enhance our model. First is utilizing Timbre information in a meaningful way (TODO - DESCRIBE IN MORE DETAIL). Second is more advanced NLP methods on the song lyrics to provide more meaningful features than just sentiment, and third is expanding the two models we are using to tackle the problem.

6.1 Graph Problem

For the graph version problem, a particularly interesting way to think about a playlists is as a path through a graph [6]. With this approach, you can pick a start song and an end song, and let the algorithm find the shortest path of some predetermined length between them. We believe this and similar approaches will create compelling playlists. In particular we think these playlists will be dynamic and able to "tell a story" more so than playlists that are just clusters of songs.

6.2 Classification

The initial effort for improving the classification version of the task will be in improving the features as previously mentioned. Beyond that, the obvious improvement is to use a non-linear regression model since the assumption our data is linearly separable is problematic.

7 Contributions

Kade was responsible for the initial pre-processing of the dataset to remove songs with incomplete data. He also implemented the baselines for sentiment analysis, k-nearest-neighbors, and linear regression classification, and the initial pipeline of scripts for tying the whole system together. He was the primary author of the report, and ran the experiments to find the reported results.

TODO DEMETRIOS - AUGMENTING MSD DATA WITH SPOTIFY DATA, ENHANCED MODELS TO USE NEW FEATURES, CROSS VALIDATION

References

[1] Last.fm. https://www.last.fm/.
[2] Million song dataset. https://labrosa.ee.columbia.edu/millionsong/.
[3] Nltk text corpora. http://www.nltk.org/book/ch02.html.
[4] Scikit learn. http://scikit-learn.org/stable/.
[5] Spotify. https://www.spotify.com/.
[6] Masoud Alghoniemy and Ahmed H. Tewfik. A network flow model for playlist generation. In *In Proc IEEE Intl Conf Multimedia and Expo*, 2001.