SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Santa Clara University
Machine Learning

Music Similarity by Acousticness

Patricia Ornelas Jauregui & Carlos Garcia

Supervisors:
Yi Fang

First version

Santa Clara, December 2019

Motivation

The idea of this project is to automate the process of creating playlists and recommending the best way to order the playlists. This process is similar to what a DJ would do live, but we want to create a system that would be able to generate the music experience from the very beginning without any outside interaction between each song. Without being restricted by music genres, this project will analyze songs via different metrics and machine learning models in order to find their similarity and be able to choose the next song to play (similar to Spotify/Pandora).

Software Python, Librosa, TensorFlow, scikit-learn and Jupyter

Table of Contents

1. Methodology

For this project, we wanted to explore different ways to recommend similar songs. The first method we explored was to use Cosine Similarity to find the similarities between songs. Each of the songs we used had 8 features normalized between 0 and 1 (listed below) so we were able to generate the low-level structure of each song into individual 8 dimensional vector. From this point we computed euclidean distance between each vector, and the shorter the distance the more similar the two songs were. Using the same data set we implemented a K-Means Clustering model. For this model, we tried to imitate the genre classification, so we set the number of clusters equal to the number of genres that we had in our data set which was 13. In order for our system to provide some serendipity, we recommended songs that were within the same cluster, but not necessarily the closest or most similar song. This would mimic a DJ the most as people do not want to continue listening to very similar songs, but they do want to have a smooth transition from one song to the next. Another model that we implemented was a Gaussian Mixture Model, where we used the same data set as the two previous models. Here we wanted to include the probabilities of a song belonging to multiple clusters as we saw that in K-Means there was quite a bit of overlap between genres. Similar to K-means, we created clusters for the songs, but now when recommending new songs we took multiple clusters into consideration. The way we found a song to recommend was to take the song, look at its top probable clusters and find other songs within the same range of probabilities of the original song. This would find more similar songs than K-Means, but they still would not necessarily be the most similar song. Ultimately these two previous models would provide serendipity and ideally recommend songs that the listener would not expect, but they would still enjoy.

Dataset: Pre-Defined Audio Features from Spotifys API in Free Music Archive (FMA)

https://freemusicarchive.org/

Below is an example of the features we used from FMA.

	acousticness	danceability	energy	instrumentalness	liveness	speechiness	tempo	valence
0	0.307539	0.498863	0.468293	0.007844	0.131117	0.117583	0.474367	0.425621
1	0.294663	0.416047	0.643350	0.001457	0.083329	0.363455	0.377140	0.211895
2	0.033163	0.567520	0.533955	0.000530	0.284035	0.094841	0.279499	0.473205
3	0.462145	0.319621	0.448963	0.468825	0.056076	0.016018	0.201340	0.467933
4	0.315940	0.358572	0.391528	0.013584	0.067466	0.367152	0.297662	0.624640

Figure 1.1: Sample Features

We also provided song recommendations by extracting the features ourselves. Since we processed the audio signal itself, we could not use the same data set as we did for the previous models. Instead, we used local mp3 files we already had. Using Librosa, we were able to get the audio time series for each song and then get its Mel Frequency Cepstral Coefficents (MFCCs) to capture information about the song. We used Principal Component Analysis (PCA) to reduce the dimensionality to only two features for each song which consistently captured more than 80 percent of the data. We converted this data into a signature and recommended songs with the lowest Earth's Mover Distance (EMD) from each other.

From the mp3 files of locally sourced songs, we used Librosa to get the tempo of each song and compared which tempo was the most similar. Part of what DJs do is pick the next song, but they also have to transition smoothly between songs. They try to match tempos of songs to make this transition seamless, so we thought tempo would be a good way of determining which songs to recommend as well. For this method, we could not use a lot of data because it takes a long time to process mp3 files.

2. Experimental Results

Cosine Similarity: For 50 randomly selected songs, 54 percent of recommendations were within the same genre. This number may be a bit skewed because our test did not exclude the song itself from being a recommendation. Overall we thought these results were good considering that there were 13 total genres.

```
Amoebic Ensemble - Repetitive Motion Sickness. Album: Limbic Rage. Genre: Jazz
Recommendation: ['Amoebic Ensemble', 'Repetitive Motion Sickness', 'Limbic Rage', 'Jazz']
Similarity: 99.99999999999999%

Recommendation: ['Arie B.', 'Weet je wel', 'Ik wil wat jij wil', 'Hip-Hop']
Similarity: 99.78391286117144%

Recommendation: ['Smoked Meat Fax Machine', 'Slums of Heaven', 'Slums of Heaven', 'Electronic']
Similarity: 99.72883707882082%

Recommendation: ['Nicky Andrews', 'Affective-4', 'Electrified Being', 'Electronic']
Similarity: 99.71902974733712%

Recommendation: ['Eta Carinae', 'Ecos', 'netBloc Vol. 07: 10 From 200 plus one', 'Electronic']
Similarity: 99.69380703338796%
```

Figure 2.1: Cosine Similarity Recommendations

K-Mean Clusters: For 50 randomly selected songs, only 21 percent recommendations were within the same genre. This number makes sense because we noticed that the clusters don't form strictly around the genre so we expected a lot of overlap. The data set also contained more of one genre than another, so that could have also lowered the percentage. However, we did notice that the recommendation for individual songs were fairly accurate as we saw, in the figure below, how one Jazz recommended 5 rock. This also shows the overlap and similarities across different genres. We can see the this in cluster plot that shows the overlap between genres.

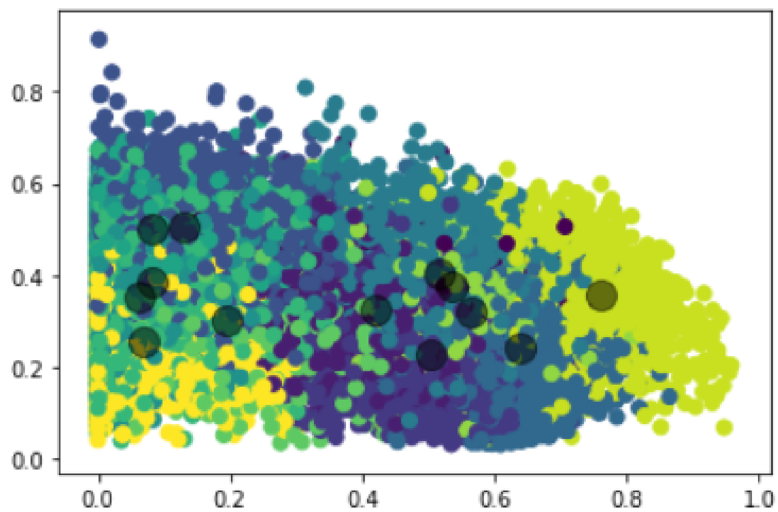

Figure 2.2: K-Means Cluster

```
Song recommedations for ['Amoebic Ensemble' 'Repetitive Motion Sickness' 'Limbic Rage' 'Jazz'
 "['Jazz']" 1995.0]
['Argumentix' 'Lend Me a Blanket' 'Tarantula Downpour 7"' 'Rock'
 "['Punk', 'Rock']" 2009.0]


['Blah Blah Blah' 'Ding dang (skat)' '30th Anniversary Blah Blah Blah'
 'Rock' "['Post-Punk', 'Rock', 'Punk']" 2009.0]


['Blah Blah Blah' 'Complete Shakespeare' 'Green Collection' 'Rock'
 "['Post-Punk', 'Rock', 'Punk']" 2007.0]


['Argumentix' 'Snapped Today' 'Nightmarcher' 'Rock' "['Punk', 'Rock']"
 2007.0]


['Argumentix' 'Boss of Goth' 'Boss of Goth' 'Rock' "['Punk', 'Rock']"
 2008.0]
```

Figure 2.3: K-Means Recommendation

Gaussian Mixture Model: For 50 randomly selected songs, we saw a slightly higher rate of genre recommendation at 31 percent. This was to be expected because in k-means we we picked random songs per cluster, here we looked at songs that were within the same probabilities for each cluster as the original song. This wouldn't give the closest song, or a random song within the same cluster, but provide the most serendipity in which song to play next. We can see that in the clustering diagram below, there is a lot of overlap so a low percentage is still expected. Also we noticed that for individual songs, the recommendations were pretty accurate as one rock song gave 4 rock recommendations.

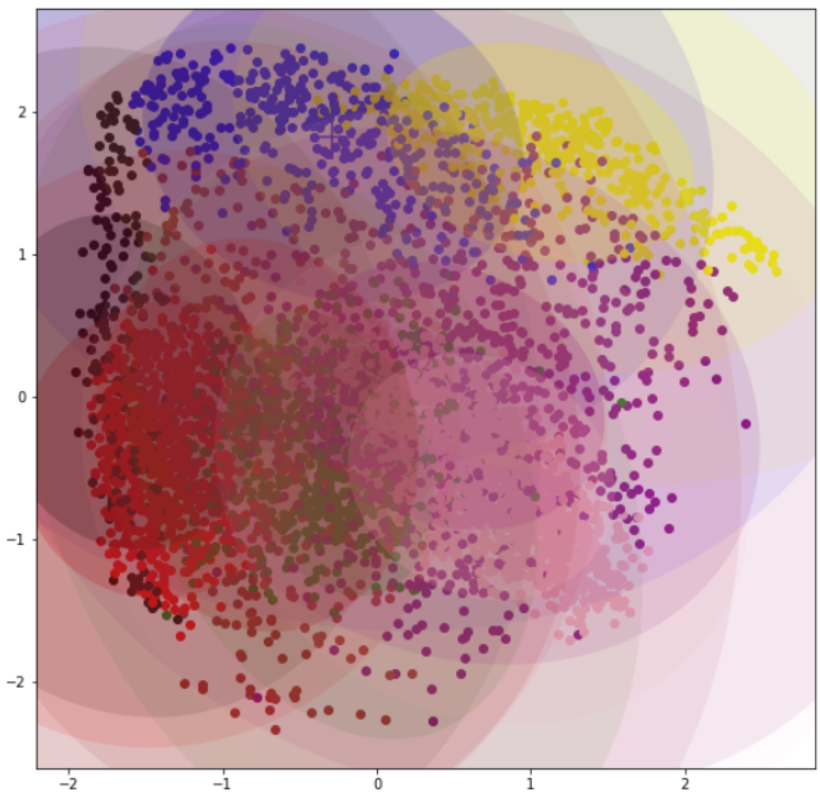

Figure 2.4: Gaussian Cluster

```
Here the recommended Songs from ['Argumentix' 'Industry Standard Massacre' 'Nightmarcher' 'Rock'
 "['Punk', 'Rock']" 2007.0]
['Mors Ontologica' 'Shoes' 'Dead And/Or Famous' 'Rock' "['Rock']" 2008.0]


['Vincebus Eruptum' 'Who Farted' 's/t' 'Rock' "['Rock']" 0.0]


['Means' 'Charlize Theron / Rob Wheeler'
 "Live at WFMU on Cherry Blossom Clinic with Terre T's Show on 11/15/2002"
 'Rock' "['Rock']" 2002.0]


['Triclops!' 'Brown Summer' "Live at WFMU on Diane's Show on 11/6/2008"
 'Rock' "['Rock']" 2008.0]


['The Fresh & Onlys' 'Feelings In My Heart'
 "Live at WFMU on Talk's Cheap 10/13/2009" 'Rock' "['Pop', 'Rock']" 2009.0]
```

Figure 2.5: Gaussian Recommendation

Tempo: After asking five people, an average of 55 percent of the recommendations were subjectively similar to the five test songs we showed. This data point is very subjective. Over half of the songs were of the same genre, so this model works fairly well.

MFCC EMD: Measuring the distance between songs based on PCA-reduced MFCCs resulted in results that were just about the same as the tempo model.

In conclusion, songs that use randomization provide more serendipity as expected, but similarity based on acoustic featues and MFCCs are more precise. This is in alignment with what we expected.

3. Conclusion

Ultimately we are satisfied with our results. We were able to use different different models to give recommendations for new songs. While we did see some low numbers in genre classification, we still believe our approach has promising applications for using machine learning in music. Some of the next steps we would like to take is to able to get the data from the Spotify API ourselves as we noticed that the songs were fairly old and hard to find online. By having a bigger and more modern data set, we believe we could get more applicable results. We also want to explore more models for generating genres and recommending songs. Another next step could be to include voice recognition, similar to Shazam, where we could listen to any song and be able to recommend other songs that are similar.

3.1 References

- Discovering Descriptive Music Genres Using K-Means Clustering by Victor Ramirez (https://medium.com/latinxinai/discovering-descriptive-music-genres-using-k-means-clustering-d19bdea5e443)
- More like this: Machine Learning Approaches to Music Similarity by Brian McFee

  (https://escholarship.org/uc/item/8s90q67r#article_main)
- Gaussian Mixture Models Explained

  (https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95)