Midterm Report: GenreClassification

Stephanie Zhou (sz244), Danny Yang (dzy4), James Chen (jzc8)

Overview:

The objective of our project is to train a classifier for predicting the genre of a song given metadata about the song. We are given a 200GB dataset of one million songs from https://labrosa.ee.columbia.edu/millionsong containing the metadata of the songs. The genre labels of the songs were aggregated from from various sources including http://www.tagtraum.com/msd_genre_datasets.html and http://www.ifs.tuwien.ac.at/mir/msd/ which contain genre labels created specifically for the Million Song Dataset (MSD)

Motivation:

There are several motivations to explore this problem or to develop an accurate genre classification system:

- It can help us understand subtle differences that underlie different genres of music such as musical properties and what the key differences between different genres are. This information can be useful for people, especially electronic musicians, who want to quantitatively know the properties of different genres
- Music sharing platforms can use this technology to automatically tag user uploads as well as recommend similar genre music to users.

Exploratory Analysis

We began by investigating the 200GB dataset. The dataset was too large, so we used a representative 2GB subset of 10000 songs for our exploratory analysis which was pre-chosen from the entire dataset. We calculated the Pearson correlation between the features. Almost all of the features have low correlation between each other, implying that they are independent. This may be important when we make independence assumptions for our model later. The only highest correlation was between Time Signature and Duration which was 0.111. Looking at the latitude and longitude locations of the artists, nearly all the artists are from North America and EU, so the subset of data is not representative of the population.

Next, we investigated the genre labels. We made use of two sources of genre labels, both of which were created specifically for the MSD. The first dataset was created by Tagtraum industries, a software company. Around 130,000 genre labels are provided, with a total of 13 genre classes consisting of the following: Reggae, Pop_Rock, RnB, Jazz, Vocal, New Age, Latin, Rap, Country, International, Blues, Electronic, Folk. We notice immediately that there is a class imbalance; 64% of the genres are Pop_Rock, with the other 12 genres nearly uniformly distributed.

Data Preprocessing/Feature Engineering

We worked from a set of ~1300 songs in this step. Each of the songS contains over 30 features. There are 16 scalar features, including key, loudness, index of fade in etc. The rest of the fields contain arrays of proportional to the length of the song, such as the pitch at each segment(the sampling rate is also given). One thing we tried was reducing the number of features and transforming certain other features. In order to keep our models simple, we first eliminated all the features except for genre, loudness, duration, key, and tempo. Originally we were going to keep energy and danceability as well, but every value in those columns was 0 so we dropped them.

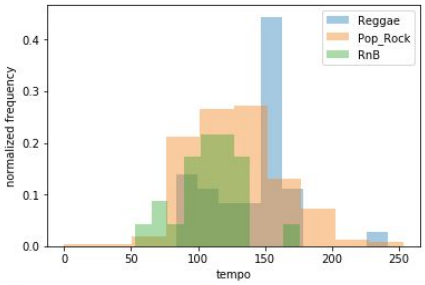
Key was represented by a number from 1-11, but this was not necessarily good to use because key 2 may not be closer to key 3 than key 11. Thus, we created the features key_1...key_11, which are binary variables to represent the one-hot encoding of the key.

The genre field for each song was a list of tags (of which there were 13 possible), and some songs had either one or two genre tags. We wanted to test our model in 2 different ways, so we created some new features to better represent the genre tags.

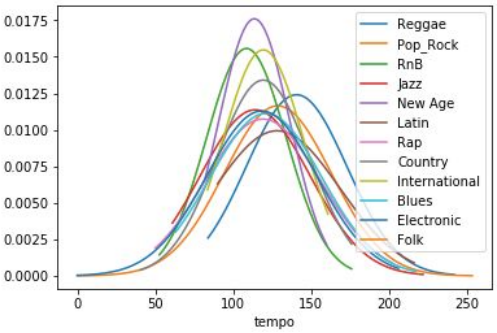
For each genre, we created a binary indicator for whether or not a song had that genre tag. In addition, we created a new feature named "single_genre" that just contained the first genre listed in the song's tagset. The former could be used in models where we train a classifier for each genre that classifies whether or not that song should have the genre tag, and the latter could be used in models where we train a classifier to give each song a single genre tag (the accuracy of this could be measured by testing if the predicted tag was actually part of the song's tagset).

Data Visualization

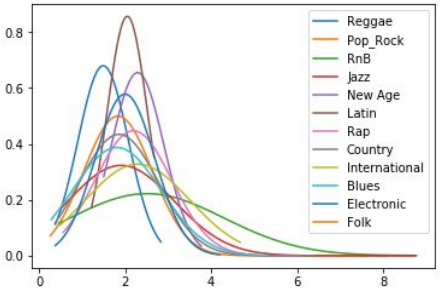
In order to better understand our features, we created a few visualization to compare some of the features across genres. Here is a visualization we created for tempo which shows its normalized distribution for three different genres:

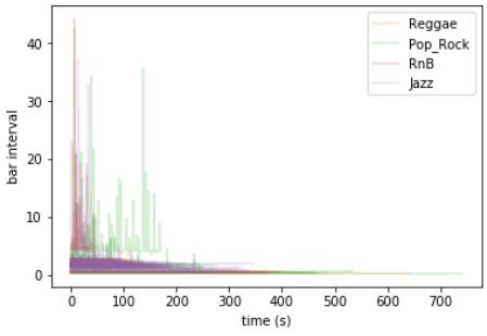


Then we calculate the mean and variance of the variable for each of the genre then plot the normal distribution for it. The difference in tempo across genres does not seem statistically significant enough. Similar result is observed in a few other scalar variables such as loudness. Which is another indication that using those variables alone is insufficient perform genre classification since they do not capture enough information about the song.



Then we decide to take a look at some denser fields produced by Echo Nest. For each song, the bar_start field contains an array which indicate the start time of each bar. Simply looking at the value doesn't tell us much because the song have varying length, we would like to transform it into a format where we can compare across songs in different genres. So for each song in a genre, we take the average time interval between two bars and plot its distribution, then plot the normal distribution using the mean and various across all genres. I think this graph shows that the bar_start field could potentially be a useful indicator. Some genre such as Reggae has relatively small variance for bar-interval so if we see a song with bar interval>3, it's extremely unlikely that the song belongs to Reggae.





The figure on the right shows the change in bar interval over time for different genres(50 samples from each genre displayed). We can see different genre does have different distribution in bar interval across time.

Baseline Models

The baseline for predicting a single genre tag is 56%, which is the frequency of occurrence of the most common genre tag, Pop_Rock. For now we disregarded the binary features we made for each genre, and instead used single_genre as the y. When testing various models, we made the following observations:

- The decision tree had about 64% accuracy (the predicted tag was in the tagset for the song).
- Random forest had worse performance than a single decision tree, about 60%.
- K-nearest neighbors had about 67% accuracy, while incorporating bagging brought our accuracy up to 73%

We then tried training one binary classifier per genre to predict whether or not a song had that genre. Combined, they could be used to generate a predicted tagset for that song. For each classifier, y was the binary feature we generated for the presence of that genre tag. The baseline for the classifier for genre X was the proportion of the data where the most frequently occurring value (either 0 or 1) for the indicator for genre X appeared. Since most of the other genres besides Pop_rock were relatively rare, this resulted in baselines of 90% or higher for every genre besides Pop_rock.

The results were not surprising- we tested decision tree, random forest, K-nearest neighbors, and bagged K-nearest neighbors as the model for each binary classifier, and none of the models performed better than the baseline (they performed exactly as well as the baseline in most cases). This is because our model is underfitting to the dataset due to its lack of complexity. We can address this using more feature engineering and by researching into effective methods to tackle the audio related features. Overfitting to the dataset should not be an unavoidable; we have plenty of data to cross-validate with to keep overfitting in check.

Future Approaches

Instead of directly eliminating features, we are planning to use PCA and CCA to find the features that are most prominent in order to reduce the dimension of the input dataset.

Building a binary classifier for each genre to predict genre is shown to be a not so promising idea. After All, this is a multi-class classification problem with non-trivial number of classes.

We are planning to experiment with Hidden Markov Models (HMMs) and apply other clustering technique such as K means and agglomerative clustering on our data with reduced dimensions. Since some of our data, such as segment pitch, have variable across different songs, we also are considering applying recurrent neural networks (RNN) since it maintains information about the previous input unit. An ensemble of several models should be useful in that we can implement a majority-vote based approach to determine the final prediction.

As an bonus, if we are able to implement a good model and have extra time, we might research into ensembling our model with an NLP based model that uses lyrics to classify genre.