Classification of Artist Genres through Supervised Learning

Richard Ridley. Mitchell Dumovic

Overview

Our project involved building a classifier to classify the musical genre for a specific artist. Our main motivation for this topic was its usefulness to the field of music: better classification techniques could improve upon music recommendation engines, help find similarities between different music genres, and reduce the need for the hand labeling of genres in streaming services. Using Spotify's public API, we retrieved a massive dataset of tracks and artists. We then employed numerous machine learning techniques to classify each artist and found that genre classification using the features we were given was much more difficult than expected.

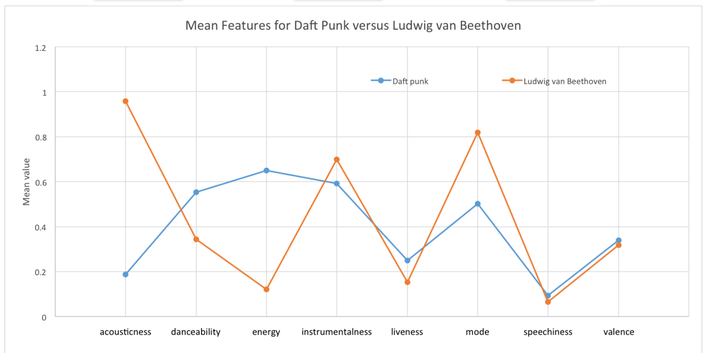
Data

All of our data comes via Spotify's publically available API that allows one to collect data about artists, their albums, and the individual songs within those albums. We ended up collecting information about over **35,000,000** songs and over **120,000** artists. For each of the songs, we measured each of the below statistics:

Acousticness	**Danceability**	**Energy**	**Instrumentalness**
Duration	**Key**	**Liveness**	**Loudness**
Mode	**Speechiness**	**Tempo**	**Valence**

Features

Each artist had a variable number of songs (each with their own song features) associated with them, so we had to collapse all of these features into one for a single artist. For each song feature, we created a single artist feature by calculating some statistical measure over all of the artist's songs. The statistical measures we used included **mean, median, variance, skew,** and **kurtosis.**

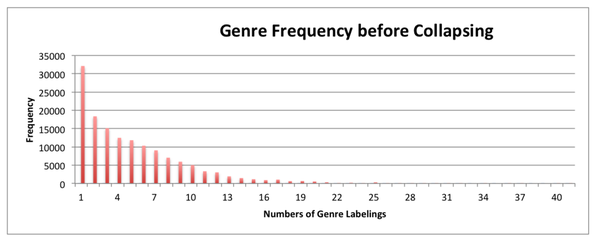


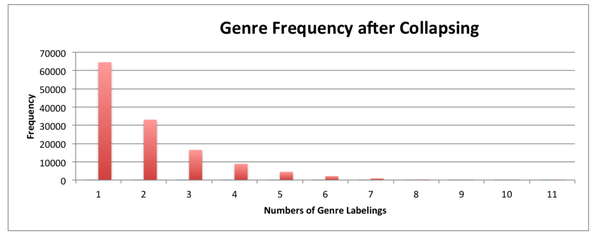
Genre Collapsing

In our original dataset, there were 1241 unique genres, many of which were only used once. In order to reduce the dimensionality of our data, we used Association rule learning techniques in order to collapse similar genres together. The main constraints we used were support (frequency of genres), confidence, and lift, defined as follows for genres A and B:

$$conf(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A)}, lift(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A) * supp(B)}$$

Genre pairs with high confidence/lift were collapsed into one pairing. In the end, we went from 1241 unique genres to **95,** reducing the average number of genres associated with an artist immensely.





Models

We employed **k-nearest neighbors** as well as **stochastic gradient descent** and **support vector machine** techniques to classify our data. Because each artist can be classified with many different genre labelings, we used a **one-versus-all** classification algorithm with SGD, where we trained a single classifier for each of the 95 genre labelings. For SVM, we used a **one-versus-one** classification algorithm using a Gaussian Kernel, which involves training a classifier for each pair of possible genres, each of which is responsible for distinguishing between the two genres.

Results

In the end we trained using a training set consisting of **21838** artists and a test set of **2426** artists. Here are our results in the form of confusion matrices (hits, misses, false positives, true negatives).

K-Nearest Neighbors		SGD		SVM	
0.023	0.977	0.067	0.933	0.115	0.885
0.032	0.968	0.031	0.969	0.028	0.972

Discussion

In general, we were disappointed with our results. Our hit-rate was much smaller than we originally expected. However, this was in large part due to the high dimensionality of our data: with 95 possible genre labelings, a hit-rate of 11.5% is far better than randomly choosing genres. Additionally, we found that we had a better hit rate on some genres than others: our algorithm in general had much more hits on niche genres like classical, metal, and deep electronic music than more generic genres like rock. The two main reasons we believe that our results were poorer than anticipated are due to the high dimensionality of our data and our base features used. The need to use an algorithm to collapse genres together distorted our original data, and the 95 genres we were left with were still far too many to train a great classifier. Additionally, while the Spotify song features may be useful for recommendation purposes, it is likely that we would need more information to be able to train a very accurate classifier than is just present in the song features.

Next Steps

If we were to work improve our results, we would definitely start by improving the quality of our training set. This would involve reinventing our feature selection and extraction algorithms as well as our genre collapsing algorithms so as to both reduce the dimensionality of our problem and provide the best possible features for prediction. Additionally, we may also look at alternative methods at classification, and perhaps first build classifiers for the genre of individual songs, which could be used to classify artists.

References

Prasanna, K., and M. Seetha. "ASSOCIATION RULE MINING ALGORITHMS FOR HIGH DIMENSIONAL DATA – A REVIEW." *International Journal of Advances in Engineering & Technology* (2012): n. pag. Web.

Silva, Vitor Da, and Ana T. Winck. "Multi-Label Classification of Music into Genres." *Applied Data Mining* (2013): 181-203. Web.

Wang, Shu. "Musical Genre Categorization Using Support Vector Machines". N.p., 2016. Web. 12 Dec. 2016.