Music Genre Classification

chunya25

Fall 2017

1 Introduction

A genre is defined as "a category of artistic composition, characterized by similarities in form, style, or subject matter." [1]

Some researchers have demonstrated that music genre preferences vary based on factors such as personality and emotional state. Cultural upbringing, peer influence, generational dynamics [5] all shape music preferences too, resulting in varying attitudes towards different genres, from strong liking, to strong dislike of certain genres.

In the digital, highly connected era, on demand music services are gaining much traction [4] and are financially sound for on-demand music providers [7].

The on-demand music streaming services attract and retain subscribers [6], and are in a position to easily gather and retain information regarding the users' music preferences. Based on this information, these companies can build models capturing the users' music preferences, and recommended similar music to these users. The benefits include:

- The users would prefer an on-demand service that knows the kind of music they like and proactively queues that kind of music for them
- The user wouldn't have to manually sort through and pick songs from a vast directory; it's painfully time consuming

This paper assumes that different music genres are sufficiently different at the bit level, and can therefore be modeled. If this is well done, then users will get better music recommendation.

2 Materials and Methods

2.1 Data Source

This project uses 106,574 processed tracks from 16,341 artists and 14,854 albums from Free Music Archive (FMA)[2]. Among these tracks, 100778 are labeled to 16 top genres, old-time/historic, country, pop, rock, easy listening, soul, rnb, electronic, folk, spoken, hip-hop, experimental and instrumental.

2.2 Data Understanding

Music information is usually described at frequency level and time level. At frequency level, there are pitch features (how high/low a note is), timbre features (the shape of music sound). These two kinds of features are orthogonal. At time level, there are rhythm features[3]. In FMA dataset, the features are generated using librosa, and stored as statics, including kurtosis, max, min, mean, median, std and skew, for each feature [2].

For pitch feature, chroma representations are a preferred way to encode harmony, and suppressing perturbations in octave height, loudness, or timbre[3].

Chroma features could be extracted in different ways, e.g. by convolving a fixed size window Short Time Fourier Transform (`chroma_stft`), or a variable sized window, constant-Q transform (`chroma_cqt`), over the audio signal to extract the time-varying frequency spectrum[3].

The time-varying frequency spectrum seems to vary per genre, indicating that this is a useful feature in helping to distinguish genres[3].

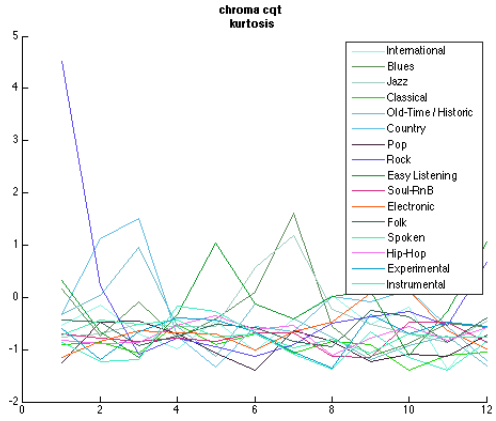

Figure 1: Chroma: Constant-Q Transform Kurtosis

MFCC: The Mel Frequency scale is commonly used to represent audio signals; it provides a rough model of human frequency perception[3].

2.3 Methods

Benchmark Starting with the assumption that examples from the same genre are similar, they'll cluster closer to each other in the n-dimensional space, where n is the number of features). Normalization and dimensionality reduction was attempted as well.

For kNN, there was a `<1%` difference in normalizing vs non-normalized data, performing at `24.8%` for either case. Due to that small difference in normalized/non-normalized data, dimensionality reduction was explored via PCA (principal

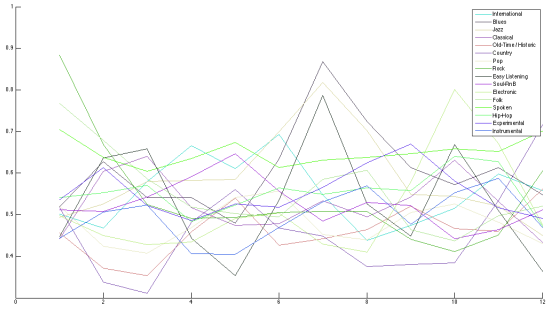

Figure 2: Chroma: Constant-Q Transform Mean

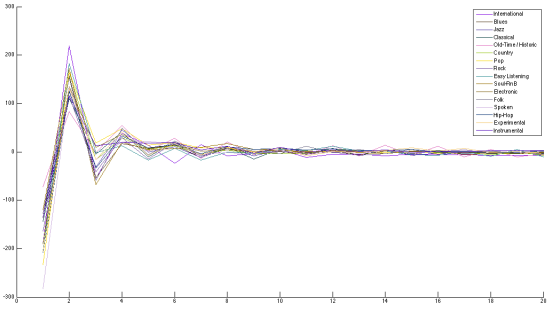

Figure 3: MFCC

component analysis). By mapping data to the top k principle component vector space, more meaningful feature set were obtained, by varying the number of principle components for PCA. Smaller components 50 performed at `19.32%`, 100 at `27.65%` , 150 at `37.27%`. At 200, the performance stood at `28.04%`. Thus, there seem to be a subset of features best suited for the clustering algorithm.

kNN Performance			
PCA	`pca_dimension`	`k_neighbors`	Accuracy
Yes	50	15	`19.32%`
Yes	100	15	`27.65%`
Yes	150	15	`37.27%`
Yes	200	15	`28.04%`

Linear Regression A linear model was trained as a baseline model to compare against Neural Network. The performance is in figure 4.

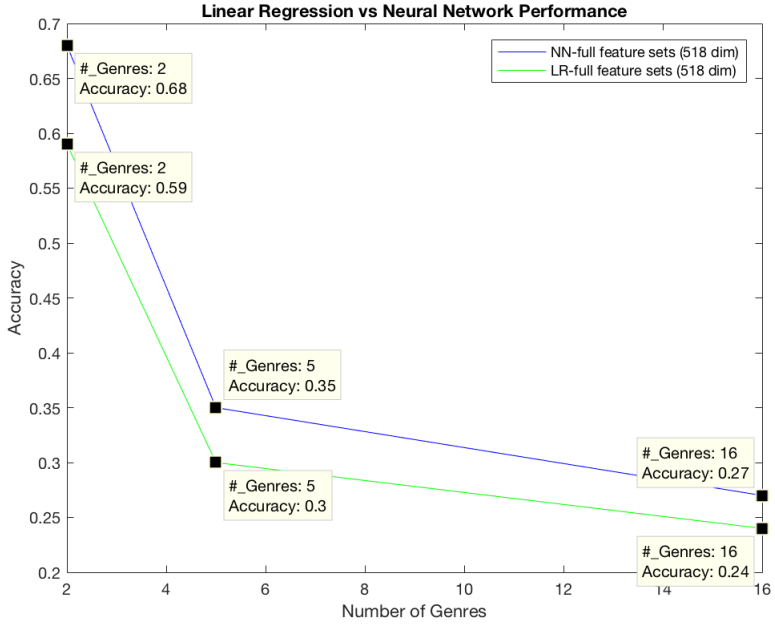

Figure 4: Accuracy vs Number of Genres modeled

Deep Learning Our full data consisted of 518 features, with each example belonging to one of 16 genres. For a particular example, the target/label is a one-hot encoded 16 dimensional vector.

To obtain the log loss, the target value is the one-hot encoded value, and the activation function is a softmax function, a normalized exponential function, whose output represents the normalized probability distribution over K classes. This makes it possible to obtain the cross entropy loss for each iteration. Using different optimization algorithms, different, hidden units, regularization strength, batch size, iterations, steps, learning rate etc, the weights are learned differently.

Regularization: L2 regularization proved most useful, compared to L1, helping us to learn a smoother and more robust model, often performing at least as good on the test data, as the training and dev set.

An extra term is added to the log loss i.e.

$$\lambda \sum_{i=1}^{k} w_i^2$$

to penalize larger weights, preventing over fitting, thus leading to a more robust model, that performs well on dev and test set as well. As seen in figure 4, the accuracy of DNN decreases with the number of genres that need to be classified.

When building the optimal parameters, patient exploration of various batch size, learning rate, hidden units architecture and regularization methods to train the full data set into 16 genres uncover that however well fitting the model is, the accuracy is rather limited. The best accuracy achieved for 16 genres was 27%. With a smaller set of genres to be classified, for this paper, 5 genres were selected, the accuracy increases to 35%. For 2 genres, the accuracy further increases to 68%.

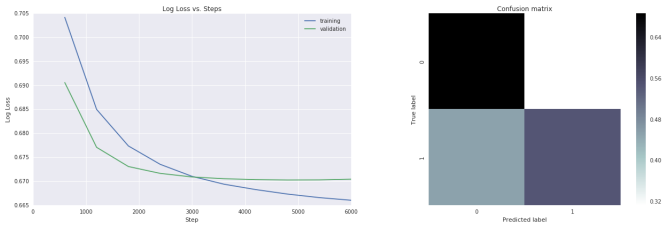

Figure 5: Loss and Confusion Matrix of DNN using Full Feature Set (518 dim) for classifying 2 Genres: Jazz and Classical

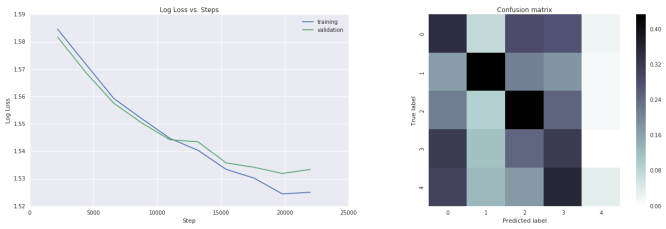

Figure 6: Loss and Confusion Matrix of DNN using Full Feature Set (518 dim) for Classifying 5 Genres: Jazz, Classical, Hip-hop, Rock, Country

3 Discussion

To explore whether feature subset combinations would improve performance on the 5-genre classification task, a few promising features were employed in various combinations e.g. chroma, mfcc, chroma + mfcc. The selected data-set comprised of tracks in 5 Genres (Jazz, classical, hip-hop, rock and country). None of the results beats performance using full feature set for the 5 genres.

- The accuracy using chroma only is 24%.
- The accuracy using mfcc only is 28%.
- The accuracy using chroma and mfcc is 30%. This accuracy is close to the accuracy using full feature set. This result is as expected because chroma and mfcc are two major independent features of music.

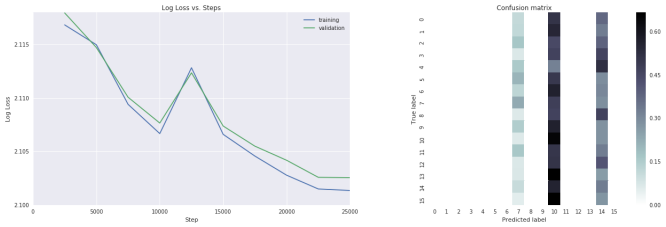

Figure 7: Loss and Confusion Matrix of DNN using Full Feature Set (518 dim) for Classifying 16 Genres: Jazz and Classical, Hip-hop, Rock, Country, Old-time/Historic, Pop, Rock, Easy Listening, Soul, Rnb, Electronic, Folk, Spoken, Experimental and Instrumental

KNN performs the best among three algorithms at classifying 16 genres. The testing accuracy of neural network is better than the testing accuracy of the linear regression model, but the difference is less notable as more genres are included in the model.

When the target label set has 16 genres, the accuracy of the neural network model is low 27%. This result might be caused by the fact that the data used is already compressed (statistical data instead of original audio recordings). Another guess is because the border of genres are fuzzy. If the overlap part of two genres is large, it is difficult to cleanly distinguish them.

Genre classification is a complex problem and the complexity grows with the number of genres to be classified. While the models can be tuned to reduce over fitting, these models don't perform as well as one would think at first, especially is analogy is drawn to similar problems e.g. digit classification. It could be due to the inherent complexity of properly modeling genre data points in a way that captures the richness, but also the distinguishing features that make it easier to tell apart different genres.

4 Conclusion

Explore different techniques, incorporate lyrics and other information such as song tags, semantic data among other meta data to see if there could be improved genre classification metrics.

References

[1] Wikipedia Contributors. Music genre. [Online; accessed 10-December-2017].

[2] Michael Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference*, 2017.

[3] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, P.W., Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python In Science Conference*, 2015.

[4] Sarah Perez. Pandora's on-demand music service finally arrives. [Online; accessed 10-December-2017].

[5] Lizardo O. Skiles Sara. Musical taste and patterns of symbolic exclusion in the united states 1993–2012: Generational dynamics of differentiation and continuity, 2015. [Online; accessed 7-December-2017].

[6] Paul Sawers. On-demand: 11 subscription music streaming services compared. [Online; accessed 7-December-2017].

[7] Papies D. Wlömerta N. On-demand streaming services and music industry revenues — insights from spotify's market entry. [Online; accessed 8-December-2017].