Automatic Music Genre Classification

Anand Bhoraskar 130050025
Charvi Rastogi 13D100018
Maitreyee Mulawkar 13D170011
Vivek Arte 13D070004

Abstract

Musical genres are categorical descriptions that are used to characterize music. As such, the increasing amounts of music available on the internet and in digital form make this characterization an important task, be it for efficient indexing and organization, or for quality recommendations for music on the internet.

Traditionally, genre classification for audio has been done manually. However, with the increasing number of artists, it is increasingly important to be able to automatically classify the music according to its genre.

Through this project, we will use previous research on this topic to create our own version of an efficient and accurate program that will classify music according to its genre.

1 Introduction

1.1 Motivation

There has been an explosion of musical content available on the internet. Some sites, such as Spotify and Pandora, carefully curate and manually tag the songs on their sites. Other sources, such as Youtube, have a wider variety of music, but many songs lack the metadata needed to be searched and accessed by users. One of the most important features of a song is its genre. Automatic genre classification would make hundreds of thousands of songs by local artists available to users, and improve the quality of existing music recommenders on the web.

1.2 MFCC

The most commonly used feature extraction method in automatic speech recognition (ASR) is Mel-Frequency Cepstral Coefficients (MFCC). MFCC mimics the logarithmic perception of loudness and pitch of human auditory system and tries to eliminate speaker dependent characteristics by excluding the fundamental frequency and their harmonics. To represent the dynamic nature of speech the MFCC also includes the change of the feature vector over time as part of the feature vector. MFCCs are also increasingly finding uses in music information retrieval applications such as genre classification and audio similarity measures. Our models use MFCC's as feature vectors.

1.3 K-Means classification

K-Means is traditionally viewed as an algorithm for the unsupervised clustering of a heterogeneous population into a number of more homogeneous groups of objects. However, it is not necessarily guaranteed to group the same types (classes) of objects together. In such cases, some supervision is needed to partition objects which have the same label into one cluster. The output field itself cannot be used in the clustering but it is used in developing a suitable metric defined on other fields.

1.4 Support Vector Machines

Support Vector Machines are quite commonly used for genre classification. Characteristics of SVM:

- High dimensional input space
- Document vectors are sparse
- Few irrelevant features

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include (i) between one of the labels and the rest (one-versus-all) or (ii) between every pair of classes (one-versus-one). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification. We use the One vs All strategy

1.5 Neural Networks

Neural networks are an important tool in machine learning and have been used to perform a wide variety of tasks. The two main components of training a neural network are feed-forward computation of activations and the back-propagation of error. A simple neural networks architecture is defined by the user and connections between nodes are acyclic. The user sets each input nodes activation, and then for each non-input node it computes its own activation by computing the weighted sum of the incoming activations from the preceding nodes it is connected to and then passing that weighted sum into a function whose bounds are zero and one (hyperbolic tangent is a commonly used activation function).

Gradient descent is then used to compute the error of each node (along with each weight), which then allows for the weights to be adjusted by small amounts in order to better produce the desired output for each input. The desired output for each input is needed in order to compute the error of each output node, after which each preceding node can use its successors error. Once the neural network has been trained, it is tested by performing feed-forward computation of activations and then comparing its output to the desired output.

1.6 Dataset

The Million Song Dataset (`http://labrosa.ee.columbia.edu/millionsong/`) is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The complete dataset has over 280 GB data. We plan to use a subset of 10000 songs(1% of the dataset). The MSD dataset provides the song in HD5 format, and can be parsed by python libraries. The songs also have been labelled with timbre vectors based on MFCC coeffiecients using the Echonest API (`http://developer.echonest.com/`)

2 Approach

2.1 Neural Networks

For feature vectors, we randomly chose 100 vectors of the MFCC coefficients, and train the neural network on these layers. This is because there was a wide range of variation in the number of MFCC coefficients present for each song. We have however ensured that any song will have same dimentionality on input layer.

We use 3 hidden layers for the binary classification. The structure of this neural network is as follows: The input layer is made up of 1200 nodes, since we choose 100 feature vectors (as mentioned above!), and each feature vector has 12 parameters. The first hidden layer is then taken to have 2400 neurons. The second hidden layer has 120 neurons. The third layer has 30 neurons, and it leads to the output layer which has 2 nodes (as it is binary classification). Note that the neural network thus constructed is completely connected. That is to say, every node in one layer is connected to every node in the subsequent layer.

We use 4 hidden layers for the multi-class classification. The structure of this neural network is as follows:
The input layer is again made up of 1200 nodes, as before. The first hidden layer is taken to have 2400 neurons. The second hidden layer is made up of 120 neurons. The third hidden layer has 60 neurons, while the fourth and final hidden layer has 30 neurons. This leads to the output node which in this case has 5 nodes (corresponding to ”rock”, ”hip hop”, ”pop”, ”jazz” and ”metal”). Note once again that the neural network thus constructed is completely connected. That is to say, every node in one layer is connected to every node in the subsequent layer.

We used the LBFGS Method to optimize the neural network. This is because it was suggested to use this for when we have smaller datasets. Gradient Descent is recommended only when the dataset is much larger.

2.2 MFCC voting

Given the vector of MFCC coefficients of all parts of the song. We expect that some of these might strongly correlate with the genre. For example, saxophone solos in jazz or heavy guitar riffs in metal. So we devise classifiers which attempt to classify individual MFCC coefficients in respective genres with certain confidence. We sum the confidence values for each genre and select the highest one as the genre of the song. We try different approaches in building the classifiers

2.2.1 Support Vector Machines

We train a **one vs rest multi-class SVM** using the MFCC coefficients. The SVM we used, (`LinearSVC()`), only provided 0/1 predictions, but had the functionality to perform multi-class classification through the one-vs-rest extension. Thus, we set the confidence to be 1 for the predicted genre, and 0 for all other genres. The SVM used belonged to the `scikit-learn` library.

The features we use for the training are the MFCC coefficients, since they are the most widely used measures of the characteristics of a song.We have run code for the binary classification of genres (i.e. distinguishing between ”rock” and ”jazz”), as well as implemented code for the multi-class classification into ”rock”, ”hip hop”, ”pop”, ”jazz” and ”metal”.

2.2.2 K-Means

For each genre, we gather all timbre vectors occurring in songs associated with that genre, and run k-means on the timbre vectors. We store the centroids calculated for all the genres. To classify a new

MFCC vector, we find the nearest centroid in each genre, and calculate the distance to that centroid. The negative of this distance is the confidence measure for each genre.

3 Results

3.1 Neural Networks

$$\begin{array}{c c}
 & \begin{array}{ccccc} \text{rock} & \text{hip hop} & \text{pop} & \text{jazz} & \text{metal} \end{array} \\
\begin{array}{c} \text{rock} \\ \text{hip hop} \\ \text{pop} \\ \text{jazz} \\ \text{metal} \end{array} & \left(\begin{array}{ccccc} 6. & 6. & 13. & 11. & 12. \\ 3. & 15. & 10. & 12. & 9. \\ 7. & 13. & 14. & 5. & 9. \\ 5. & 11. & 7. & 15. & 11. \\ 3. & 1. & 6. & 1. & 36. \end{array}\right)
\end{array}$$

This confusion matrix was obtained for 1000 training examples (balanced out as 200 for each of the five genres), and 250 test examples (also balanced out as 50 for each genre).

$$\begin{array}{c c}
 & \begin{array}{cc} \text{rock} & \text{jazz} \end{array} \\
\begin{array}{c} \text{rock} \\ \text{jazz} \end{array} & \left(\begin{array}{cc} 31. & 17. \\ 14. & 35. \end{array}\right)
\end{array}$$

This confusion matrix was obtained for 400 training examples (balanced out as 200 for each of the two genres), and 100 test examples (also balanced out as 50 for each genre). Here, three test examples were removed as they didn't have enough number of MFCC features.

3.2 Support Vector Machines

$$\begin{array}{c c}
 & \begin{array}{ccccc} \text{rock} & \text{hip hop} & \text{pop} & \text{jazz} & \text{metal} \end{array} \\
\begin{array}{c} \text{rock} \\ \text{hip hop} \\ \text{pop} \\ \text{jazz} \\ \text{metal} \end{array} & \left(\begin{array}{ccccc} 0. & 6. & 0. & 6. & 3. \\ 0. & 11. & 0. & 3. & 1. \\ 1. & 5. & 0. & 9. & 0. \\ 0. & 1. & 0. & 14. & 0. \\ 0. & 1. & 0. & 1. & 13. \end{array}\right)
\end{array}$$

This confusion matrix was obtained for 300 training examples (balanced out as 60 for each of the five genres), and 75 test examples (also balanced out as 15 for each genre).

$$\begin{array}{c c}
 & \begin{array}{cc} \text{rock} & \text{jazz} \end{array} \\
\begin{array}{c} \text{rock} \\ \text{jazz} \end{array} & \left(\begin{array}{cc} 27. & 23. \\ 13. & 37. \end{array}\right)
\end{array}$$

This confusion matrix for binary classification was obtained for 400 training examples (balanced out as 200 for each of the two genres), and 100 test examples (also balanced out as 50 for each genre).

3.3 K-Means

$$\begin{array}{c c}
 & \begin{array}{cc} \text{rock} & \text{jazz} \end{array} \\
\begin{array}{c} \text{rock} \\ \text{jazz} \end{array} & \left(\begin{array}{cc} 34. & 16. \\ 16. & 34. \end{array}\right)
\end{array}$$

This confusion matrix was obtained for 400 training examples (balanced out as 200 for each of the two genres), and 100 test examples (also balanced out as 50 for each genre).

4 Conclusion

We confirm that the MFCC coefficients provide significant insight into the genre of the song. For larger datasets, more finely tuned models and pre-processing in all the three techniques that we discussed have commendable power to recognize genre. However, for a machine to comprehend more features of songs like us humans do, we need better signal processing techniques in particular, ones for reliable chord, beat, and instrument detection will allow us to analyze and organize musical content better. Also speech recognition techniques to understand lyrics can greatly help in the genre classification of songs.

Further, the distinction between genres may not always be clear enough. We can see this from the fact that the tags in fact do have multiple genres in them. This in turn, makes the classification harder.

5 Future Work

Advances in signal processing techniques in particular, ones for reliable chord, beat, and instrument detection will allow us to better analyze and organize musical content. Other features used in voice recognition for example, the zero-crossing rate and the short-term sound power spectrum may also correlate with musical genre and enhance classification accuracy.

6 References

1. Tzanetakis, G., Georg, E., & Cook, P. (2001, October). Automatic musical genre classification of audio signals. In *Proceedings of the 2nd International Symposium on Music Information Retrieval, Indiana.*

2. Li, T., Ogihara, M., & Li, Q. (2003, July). A comparative study on content-based music genre classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 282-289). ACM.

3. Xu, C., Maddage, N. C., Shao, X., Cao, F., & Tian, Q. (2003, April). Musical genre classification using support vector machines. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on* (Vol. 5, pp. V-429). IEEE. Chicago

4. Camenzind, T., & Goel, S. `#jazz`: Automatic Music Genre Detection. *Stanford University (2013)*

5. Scaringella, N., Zoia, G., & Mlynek, D. (2006). Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine, 23*(2), 133-141.