Music Genre Classification

CS725 Project

Indian Institute of Technology, Bombay
Department of Computer Science and Engineering

Anshul Gupta (16305R001)
Khursheed Ali (163059009)
Abhijeet Dubey (16305R006)
Nithin S (16305R007)

Contents

1 Project Description

Sometimes it happens that we listen to a particular music, we instantly develop an affinity towards that genre and want to listen to same type of music. Or sometimes we just want to organize our music collection based on genre. This project aims to classify music into different categories such as:

1. Blues
2. Classical
3. Country
4. Disco
5. Hip Hop
6. Jazz
7. Metal
8. Pop
9. Reggae
10. Rock

2 Papers

- A Comparative Study on Content-based Music Genre Classification [1]
- Deep Content-based Music Recommendation [2]
- A Benchmark Dataset for Audio Classification and Clustering [3]

3 Datasets

We will use *GTZAN Genre Collection* [4]. The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .au format.
We are dividing the dataset into training set of size 800 and test set of size 200. This is done randomly after the features are extracted from the audio files.

4 Feature Engineering

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The design of descriptive features for a specific application is the main challenge in building pattern recognition systems. Once the features are extracted standard machine learning techniques which are independent of the specific application area can be used.

4.1 Timbral Texture Features

The features used to represent timbral texture are based on standard features proposed for music-speech discrimination and speech recognition. The calculated features are based on the short time Fourier transform (STFT) and are calculated for every short-time frame of sound. The following specific features are used to represent timbral texture in our system

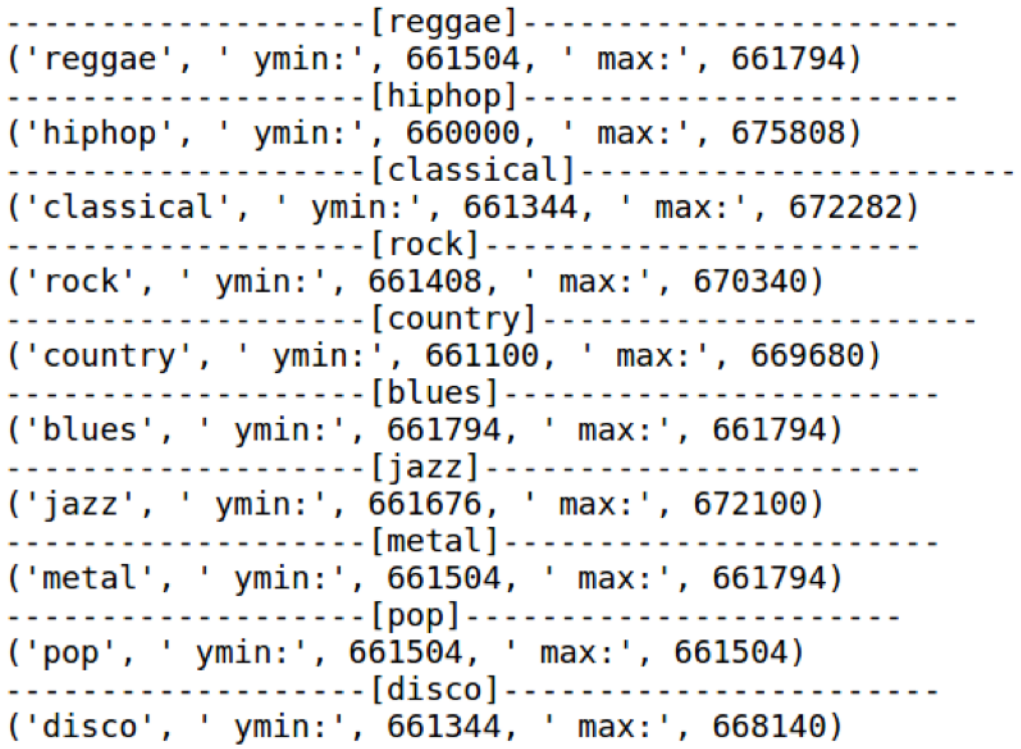

Figure 1: Minimum(ymin) and maximum(max) samples for each genre

4.1.1 Spectral Centroid

The spectral centroid is defined as the center of gravity of the magnitude spectrum of the STFT

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]}$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n. The centroid is a measure of spectral shape and higher centroid values correspond to "brighter" textures with more high frequencies.

The time duration for each audio file is slightly different(30 seconds or 31 seconds). Hence, with a sampling rate of 22050 we are getting approximately 660000 samples per file. For finding the mean and variance, we followed the below process. We are finding centroid for each batch of sample. Size of each batch being 2048 and hop length of 512, thus getting approximately 1290 centroids per audio file. Using the above mentioned formula, we are finding the mean and variance for 1290 centroid samples.

The same process is applied for all the remaining features.

4.1.2 Spectral Rolloff

The spectral rolloff is defined as the frequency R_t below which 85% of the magnitude distribution is concentrated.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n].$$

The rolloff is another measure of spectral shape.

4.1.3 Time Domain Zero Crossings

$$Z_t = \frac{1}{2}\sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])|$$

where the *sign* function is 1 for positive arguments and 0 for negative arguments and $x[n]$ is the time domain signal for frame t. Time domain zero crossings provide a measure of the noisiness of the signal.

4.1.4 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCC) are perceptually motivated features that are also based on the STFT. After taking the log-amplitude of the magnitude spectrum, the FFT bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Finally, in order to decorrelate the resulting feature vectors a discrete cosine transform is performed. Although typically 13 coefficients are used for speech representation, we have found that the first five coefficients provide the best genre classification performance.

4.1.5 Root Mean Square Energy

Root-mean-square (RMS) energy for each frame is calculated from the audio samples y. Computing the energy from audio samples is faster as it doesn't require a STFT calculation. However, another option is using a spectrogram which will give a more accurate representation of energy over time because its frames can be windowed.

4.1.6 Spectral Contrast

Octave-based Spectral Contrast considers the spectral peak, spectral valley and their difference in each sub-band. For most music, the strong spectral peaks roughly correspond with harmonic components; while non-harmonic components, or noises, often appear at spectral valleys. Thus, Spectral Contrast feature could roughly reflect the relative distribution of the harmonic and non-harmonic components in the spectrum.

4.2 Timbral Texture Feature Vector

To summarize, the feature vector for describing timbral texture consists of the following features: means and variances of spectral centroid, rolloff, zerocrossings over the texture window , rms energy, spectral contrast, and means and variances of the first five MFCC coefficients over the texture window (excluding the coefficient corresponding to the DC component) resulting in a 20-dimensional feature vector.

4.3 Librosa

The features were extracted using the python package LibROSA [5]. LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

5 Approaches using in-built libraries

Once the feature extraction is done and we have the dataframe, we can apply standard off the shelf python libraries for classification.

From Figure 2 we can see that the best accuracy achieved so far on test dataset is by **Gradient Boosting**.

6 Approaches using our implementations

The dataset is split into 80:20 ratio between training dataset and test dataset randomly. The dataset then remains fixed for all training models.

In this section, we will explain about the model which we implemented.

	Trained Misclassified Points	Trained Accuracy	Test Misclassified Points	Test Accuracy
Decision Tree	215	73	95	52
Random Forest	63	92	80	60
K-Neighbors	401	49	145	27
Gradient Boosting	1	99	85	57
Logistic Regression	385	51	105	47
Support Vector	631	21	156	22
Bernoulli NB	638	20	154	23
Gaussian NB	462	42	113	43
Adaboost	539	32	146	27

Figure 2: Off the shelf library functions

6.1 Multi-class Neural Network1

The neural network which we implemented had following configuration:

- **Hidden Layers**: 7
- **Neurons per hidden layer**: 40
- **Learning Rate**: 0.01
- **Iterations**: 10,000

Table 1: Neural Network Accuracy

	Misclassified	**Accuracy**
Train	10	98.0%
Test	112	44.0%

From Table 1, we can see that our multi-class neural network outperformed 5 out of 9 in-built library functions (Ref: Figure 2).

6.2 Decision Trees

The configuration of our decision tree is:

- **Maximum depth of tree**: 20
- **Minimum record count**: 2

Table 2: Decision Tree Accuracy

	Misclassified	**Accuracy**
Train	0	100.0%
Test	100	50.0%

From Table 2, we can see that our decision tree model outperformed 6 out of 9 in-built library functions (Ref: Figure 2).

Figure 4 shows the confusion matrix of test data classification by our Decision Tree model.

You can also see the decision tree in Figure 3.

6.3 Random Forest

Our Random Forest model is a wrapper around the decision tree model. We randomly partition the dataset into multiple data subsets and train the decision tree on each subset. The configuration of the Random Forest model is:

- **Number of trees**: 50
- **Maximum depth of tree**: 7
- **Minimum record count**: 20

Table 3: Decision Tree Accuracy

	Misclassified	**Accuracy**
Train	8	99.0%
Test	73	63.5%

From Table 3, we can see that our random forest model outperformed 9 out of 9 in-built library functions (Ref: Figure 2).

Figure 5 shows the confusion matrix of test data classification by our Random Forest model.

6.4 Convolutional Neural Network

We also made an attempt at classification of genres using the Power Spectrogram of the audio files. We processed the audio files and came up with the spectrograms.

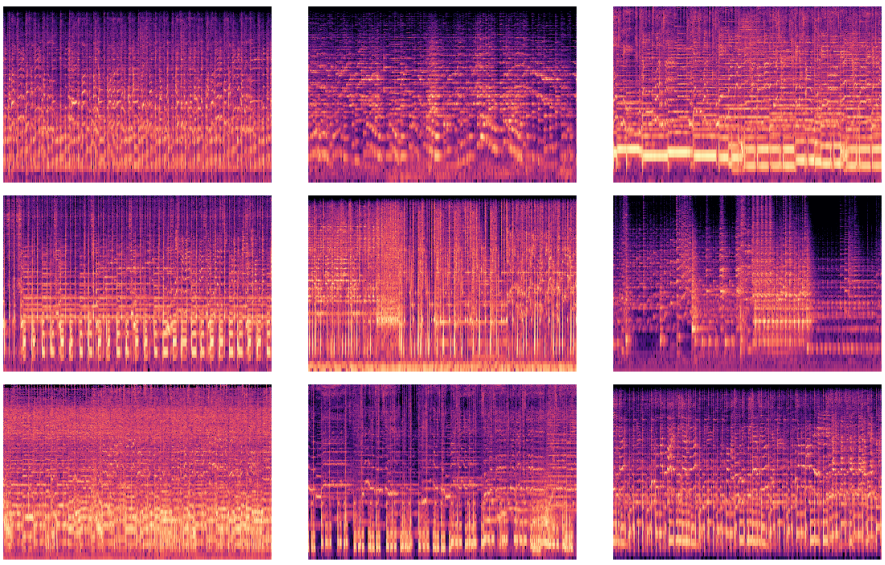

Table 4: Power Spectrograms of Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop and Reggae

From Table 4 we can see that the spectrograms are quite distinct from each other. Each genre has a particular rhythm which is captured in this image. We tried to classify the genres by training a Convolutional neural network on these images.

We used tensorflow library for creating the LeNet model. But due to some logical errors, we were unable to complete it within time.

Each spectrogram is a 341 x 224 x 4 image.

7 Observations and Conclusions

The original paper [4] achieved the accuracy of 61% using Timbral Features as well as Rhythmic Features of the audio files.

We were able to achieve more accuracy (63.5 %) than them using only Timbral Features.

Also we were able to achieve much better accuracy than most standard library implementations of classification algorithms.

8 Future Work

Since we are only using Timbral features and still able to achieve a good accuracy, we believe that if we also include Rhythmic features in our dataset, we may be able to achieve much higher accuracy then what we are getting now.

Also the dataset which we used in this project was very small (only 1000 audio files) and not sufficient to train a neural network. We can train our models on a much bigger dataset available openly at [6].

9 Timeline

9.1 Before Mid Stage Presentation

- Analysis of spectrum of audio files
- Using standard neural network library implementation for classification

9.2 After Mid Stage Presentation

- Setting up audio processing libraries
- Pre-processing of audio files
- Classification using standard off-the-shelf libraries
- Building our own models
  - Multi-Class Neural Network
  - Decision Tree
  - Random Forest
  - Convolutional Neural Network

10 Contributions

- **Abhijeet Dubey**: Preprocessing, spectral analysis
- **Anshul Gupta**: Convolutional Neural Network, Standard Libraries (KNN, Multinomial Naive Bayesian, Logistic Regression, SVC, Random Forest, etc.)
- **Khursheed Ali**: Decision Trees, Random Forest
- **Nithin S**: Multi-Class Neural Networks

Feature extraction: Done by everyone.

References

[1] Tao Li, Mitsunori Ogihara, and Qi Li. 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*, SIGIR '03, pages 282–289, New York, NY, USA, 2003. ACM.

[2] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 2643–2651, USA, 2013. Curran Associates Inc.

[3] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, September 11-15 2005. `http://ismir2005.ismir.net/proceedings/2117.pdf`.

[4] G. Tzanetakis and P. Cook. Gtzan genre collection. `http://marsyas.info/downloads/datasets.html`.

[5] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 18 – 25, 2015.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

Figure 3: Decision Tree

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	8	0	4	0	0	2	0	0	1	0
classical	1	19	0	0	0	2	0	0	0	1
country	3	1	8	3	3	2	0	0	4	0
disco	1	1	3	11	0	1	0	2	1	2
hiphop	1	0	0	1	11	0	2	4	3	1
jazz	1	1	3	0	0	12	0	0	0	0
metal	4	0	0	0	0	0	13	0	0	1
pop	0	0	1	4	3	1	0	8	3	0
reggae	0	0	2	4	5	0	0	3	10	0
rock	3	0	3	4	2	0	2	0	0	0

Figure 4: Confusion Matrix of Decision Tree

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	7	0	2	0	0	2	1	0	3	0
classical	1	19	0	0	0	2	0	0	0	1
country	2	0	14	0	0	0	0	3	2	3
disco	0	0	2	11	1	1	2	2	0	3
hiphop	1	0	0	1	14	0	2	1	1	3
jazz	0	0	2	0	0	14	0	0	0	1
metal	3	0	1	0	0	0	14	0	0	0
pop	0	0	0	1	3	0	0	14	1	1
reggae	0	0	1	2	6	0	0	1	13	1
rock	0	0	5	3	0	0	2	1	1	2

Figure 5: Confusion Matrix of Random Forest