Music Genre Classification

Krishna Rukmini, Puthucode
Department of Computer Science
Georgia State University
Georgia, GA 30303
kputhucode1@student.gsu.edu

Abstract

Music genre labels are useful for organizing and categorizing songs, albums, and artists into larger groups with similar musical features. From physical music stores to streaming services, music genres have been widely used to categorize music. However, in the field of music information retrieval, music genre classification has proven to be a difficult task (MIR). Due to their intrinsic subjective nature, music genres are difficult to categorize and classify in a systematic and consistent manner. The goal of this project is to classify the genre of music using neural networks. For this, we would need to extract information from the audio samples such as spectrograms, MFCC, etc. and then use a model to classify the music genre. This model can be used to automatically classify the music genre.

1 Literature Review

Music genre classification has been a widely studied area of research since the early days of the Internet. [1] addressed this problem with supervised machine learning approaches such as Gaussian Mixture model and k-nearest neighbour classifiers. They introduced 3 sets of features for this task categorized as timbral structure, rhythmic content and pitch content. Hidden Markov Models (HMMs), which have been extensively used for speech recognition tasks, have also been explored for music genre classification [2] [3]. Support vector machines (SVMs) with different distance metrics are studied and compared in [4] for classifying genre. In [5], the authors discuss the contribution of psycho-acoustic features for recognizing music genre, especially the importance of STFT taken on the Bark Scale [6]. Mel-frequency cepstral coefficients (MFCCs), spectral contrast and spectral roll-off were some of the features used by [1]. A combination of visual and acoustic features are used to train SVM and AdaBoost classifiers in [7]. With the recent success of deep neural networks, a number of studies apply these techniques to speech and other forms of audio data [8][9]. Representing audio in the time domain for input to neural networks is not very straight-forward because of the high sampling rate of audio signals. However, it has been addressed in [10] for audio generation tasks. A common alternative representation is the MFCC of a signal, a CNN was developed to predict the music genre using the raw MFCC matrix as input in [11].

2 Data Set:

For this project we need a dataset of audio tracks having similar size and similar frequency range. GTZAN genre classification dataset is the most recommended dataset for the music genre classification project.

The GTZAN genre collection dataset was collected in 2000-2001. It consists of 1000 audio files each having 30 seconds duration. There are 10 classes (10 music genres) each containing 100 audio

tracks. Each track is in .wav format. It contains audio files of the following 10 genres:

Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock

3 Methodology

This task of classifying audio files is split into two parts as shown in Figure 1

3.1 Classification based on features

The first methodology is to use hand-crafted features from both the time and frequency domains. We compare the performance of various classical machine learning classifiers that have been trained with these features. The features that are most helpful in this multi-class classification task have been identified.

3.2 Classification based on MFCCs

Another method is to use deep learning to train a CNN model end-to-end to predict the genre label of an audio signal purely based on its Mel-Frequency Cepstral coefficients (MFCCs).

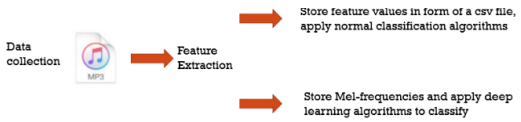

Figure 1: Methodology

4 Data preprocessing

4.1 Feature extraction from Audio signal:

Every audio signal consists of many features. However, we must extract the characteristics that are relevant to the problem we are trying to solve. The process of extracting features to use them for analysis is called feature extraction.

The first method of suggested models is described in this section, namely those that require hand-crafted features to be fed into a machine learning classifier.

The spectral features (frequency-based features), which are obtained by converting the time-based signal into the frequency domain using the Fourier Transform, like fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral roll-off, etc. Librosa, a Python library, was used to retrieve the features.

4.2 Time Domain Features

These are features which were extracted from the raw audio signal.

4.2.1 Central moments:

This consists of the mean, standard deviation and skewness of the amplitude of the signal.

4.2.2 Zero Crossing Rate (ZCR):

In short, The zero crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back. This feature has been used heavily in both speech

recognition and music information retrieval. The frame length is chosen to be 2048 points with a hop size of 512 points. Note that these frame parameters have been used consistently across all features discussed in this section. Finally, the average and standard deviation of the ZCR across all frames are chosen as representative features.

It usually has higher values for highly percussive sounds like those in metal and rock.

4.2.3 Root Mean Square Energy (RMSE):

RMSE is calculated frame by frame and then we take the average and standard deviation across all frames.

4.3 Frequency Domain Features

The audio signal can be transformed into the frequency domain by using the Fourier Transform. We then extract the following features.

4.3.1 Mel-Frequency Cepstral Coefficients (MFCC):

Introduced in the early 1990s by Davis and Mermelstein, MFCCs have been very useful features for tasks such as speech recognition . First, the Short-Time FourierTransform (STFT) of the signal is taken with n fft=2048 and hop size=512 and a Hann window. Next, we compute the power spectrum and then apply the triangular MEL filter bank, which mimics the human perception of sound. This is followed by taking the discrete cosine transform of the logarithm of all filterbank energies, thereby obtaining the MFCCs. The parameter n mels, which corresponds to the number of filter banks, was set to 20 in this study.

4.3.2 Chroma Features:

This is a vector which corresponds to the total energy of the signal in each of the 12 pitch classes. The chroma vectors are then aggregated across the frames to obtain a representative mean and standard deviation.

4.3.3 Spectral Centroid:

For each frame, this corresponds to the frequency around which most of the energy is centered. It is a magnitude weighted frequency calculated using spectral magnitude of frequency bin k and frequency corresponding to bin k.

4.3.4 Spectral Band-width:

The p-th order spectral band-width corresponds to the p-th order moment about the spectral centroid

4.3.5 Spectral Roll-off:

This feature corresponds to the value of frequency below which 85(this threshold can be defined by the user) of the total energy in the spectrum lies

The mean and standard deviation of the values taken across frames are considered the representative final feature that is fed to the model for each of the spectral features mentioned above. Machine learning algorithms will be trained with the features outlined in this section.

5 Classifiers Implemented

The three machine learning classifiers used in this study are briefly described in this section.

5.1 Logistic Regression (LR)

Typically, this linear classifier is used for binary classification tasks. The LR is used as a one-vs-rest method to solve this multi-class classification problem. That is, 7 binary classifiers are

trained separately. The predicted class is selected from among the seven classifiers with the highest likelihood during the test period.

5.2 Random Forest (RF)

Random Forest is an ensemble learner that combines the predictions of a predetermined number of decision trees into a single prediction. It is based on the combination of two key principles: 1) Bootstrap aggregation (or bagging) is the process of training each decision tree with only a subset of the training samples. 2) Each decision tree must only use a random subset of the features to make its prediction.The RF's final predicted class is determined by the individual classifiers' majority vote.

5.3 Gradient Boosting (XGB)

Boosting is another ensemble classifier that is obtained by combining a number of weak learners (such as decision trees). However, unlike RFs, boosting algorithms are trained in a sequential manner using forward stagewise additive modelling.

During the early iterations, the decision trees learnt are fairly simple. As training progresses, the classifier become more powerful because it is made to focus on the instances where the previous learners made errors. At the end of training, the final prediction is a weighted linear combination of the output from the individual learners. XGB refers to eXtreme Gradient Boosting, which is an implementation of boosting that supports training the model in a fast and parallelized manner.

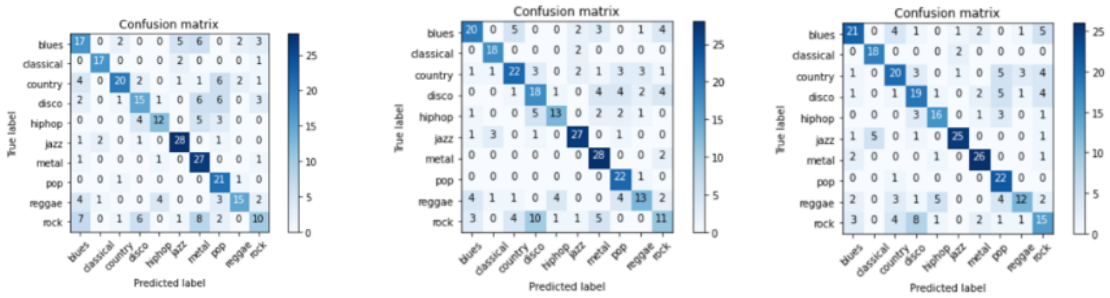

Figure 2: Confusion matrix obtained by Logistic Regression, Random Forest, Gradient Boosting

6 Mel-frequency cepstral coefficients (MFCC)

The mel-frequency cepstral coefficients (MFCC) are a widely adopted audio feature set for various audio processing tasks such as speech recognition, environmental sound recognition, and music genre classification. MFCC features on various time scales and with different modeling techniques, such as autoregressive models. Few researchers compared the MFCCs to the short-time Fourier transform (STFT), beat histogram and pitch histogram feature sets, concluding that MFCCs give best performance as an independent feature set.

Given a frame of audio, the computation of MFCC involve the following steps that mimic the low-level processing in the human auditory system:
1) transformation of the audio frame to the frequency domain using the STFT;
2) mapping the frequency bins to the mel-scale, using triangular overlapping windows;
3) taking the logs of the mel-band responses;
4) applying a discrete cosine transform (DCT) to the mel-bands.

In this model, I have used the Librosa library to convert the audio file from the GTZAN datasets into MFCC features.In particular,we choose the frame size as 25ms. Each 30second soundtrack has 1293 frames and 13 MFCC features, C1 to C13. Figure below shows some examples of the Mel frequency cepstrum plots of the music signals in the database. We apply the Librosa library to extract audio features, i.e. the Melfrequency cepstral coefficients (MFCC), from the raw data and feed it to the neural networks in order to classify.

6.1 Convolution Neural Network (CNN)

• Convolution: This step involves sliding a matrix filter (say 3x3 size) over the input image which is of dimension image width x image height. The filter is first placed on the image matrix and then we compute an element-wise multiplication between the filter and the overlapping portion of the image, followed by a summation to give a feature value. We use many such filters , the values of which are 'learned' during the training of the neural network via backpropagation.

• Pooling: This is a way to reduce the dimension of the feature map obtained from the convolution step, formally know as the process of down sampling. For example, by max pooling with 2x2 window size, we only retain the element with the maximum value among the 4 elements of the feature map that are covered in this window. We keep moving this window across the feature map with a predefined stride.

• Non-linear Activation: The convolution operation is linear and in order to make the neural network more powerful, we need to introduce some non-linearity. For this purpose, we can apply an activation function such as ReLU4 on each element of the feature map. s.

In order to prevent over-fitting , two strategies are adopted:
1. L2-Regularization: Excessively high weights are penalized using this method. for the weights to be distributed over all model parameters rather than just a few. Smaller weights, on the other hand, would seem to correspond to a less complex model, preventing over-fitting. In this analysis, is set to a value of 0.001
2. Dropout: This is a regularization mechanism in which we shutoff some of the neurons (set their weights to zero) randomly during training. In each iteration, we thereby use a different combination of neurons to predict the final output. This makes the model generalize without any heavy dependence on a subset of the neurons. A dropout rate of 0.3 is used, which means that a given weight is set to zero during an iteration, with a probability of 0.3.

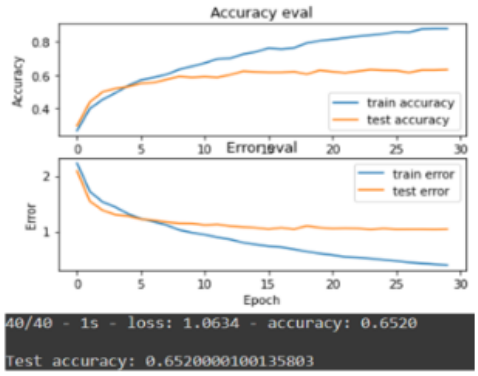

Figure 3: CNN Performance

6.2 The Long Short Term Memory Network (LSTM)

The extracted features are input to the Long Short Term Memory (LSTM) neural network model for training.Our LSTM are built with Keras and Tensorflow.

Convolutional Neural Networks (CNN) are one approach to solving this Music Genre Classification problem. But I've tried something new: Recurrent Neural Networks with Long Short Term Memory (RNN/LSTM). This LSTM is an RNN subdivision. The RNN style is similar to the conventional neural network style.It keeps track of previous data and uses it to forecast future results. Furthermore, LSTM is a modern, improved division that solves the problem of long-term dependencies. Despite this, RNN predicts the present state using the stored data as information from the past. It fails to link the information when the gap is large between the current state and the past state where the information needs to be taken from. The network that I have used has 4 layers.

In 30 epochs, the model is trained with a dataset. For the model's assessment, the accuracy and loss functions are taken into account. To accomplish this, I must use the model to predict the appropriate outcome on the evaluation dataset and then compare that result to the predicted target with the actual response.

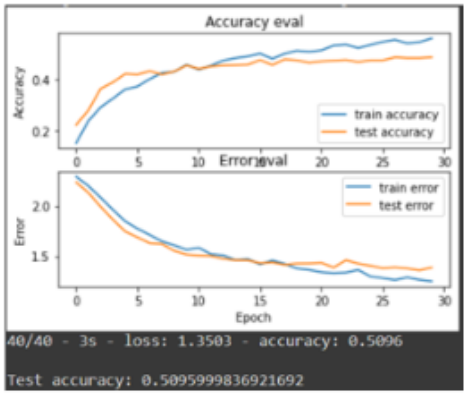

Figure 4: RNN-LSTM Performance

7 Results:

As per the metrics of models applied on MFCCs, RNN-LSTM performed well comparatively. Table below compares the model's metrics as achieved on running it on training and testing data set.

Model Name	Accuracy	Loss	validation accuracy	validation loss
Multi-Layer Perceptron	0.4678	1.8973	0.4193	2.3432
CNN	0.8727	0.4074	0.6320	1.0396
RNN - LSTM	0.5772	1.2283	0.4880	1.3856

As per the metrics of models applied on the selected features, gradient boosting performed well comparatively. Table below compares the model's metrics as achieved on running it on training and testing data set.

Model Name	Accuracy	F-Score	ROC AUC
Logistic Regression	0.61	0.61	0.93
Random forest	0.63	0.63	0.94
Gradient Boosting	0.65	0.65	0.94

7.1 Most Important Features

In this section, we investigate which features contribute the most during prediction, in this classification task. To carry out this experiment, we chose the XGB model, based on the results discussed in the previous section. To do this, we rank the top 20 most useful features based on a scoring metric (Figure 3). The metric is calculated as the number of times a given feature is used as a decision node among the individual decision trees that form the gradient boosting predictor. As can be observed from Figure 3, MFCCs appear the most among the important features. Previous studies have reported MFCCs to improve the performance of speech recognition systems. Our experiments show that MFCCs contribute significantly to this task of music genre classification. The mean and standard deviation of the spectral contrasts at different frequency bands are also important features. The music tempo, calculated in terms of beats per minute also appear in the top 20 useful features.

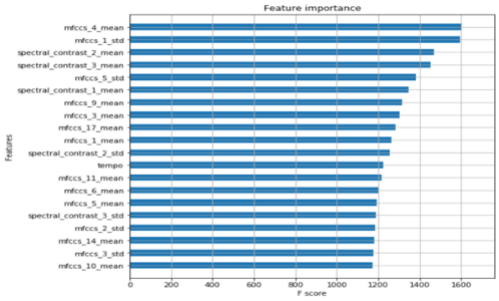

Figure 5: Important features

8 Conclusion

Music genre classification plays a vital role in music industry as manually arranging data is an arduous task. And even recommendations for similar music genres will also be easy.

As per the conclusions both normal machine learning models and the results of RNN-LSTM seems comparatively low, but LSTM is good technique to use for music genre classification as it remembers the past result of the cell in the recurrent layer and classify music more better and efficient way

On the other hand, CNN has highest metric values but there's a saying in machine learning "Not the best algorithm, but the model with more data wins", which is applicable to CNN here, scores would have been better with huge dataset .

9 References

[1] George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing 10(5):293– 302

[2] Nicolas Scaringella and Giorgio Zoia. 2005. On the modeling of time information for automatic genre recognition systems in audio signals. In ISMIR. pages 666–671.

[3] Hagen Soltau, Tanja Schultz, Martin Westphal, and Alex Waibel. 1998. Recognition of music types. In Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. IEEE, volume 2, pages 1137–1140.

[4] Michael I Mandel and Dan Ellis. 2005. Song-level features and support vector machines for music classification. In ISMIR. volume 2005, pages 594–599.

[5] Thomas Lidy and Andreas Rauber. 2005. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In ISMIR. pages 34–41.

[6] E Zwicker and H Fastl. 1999. Psychoacoustics facts and models.

[7] Loris Nanni, Yandre MG Costa, Alessandra Lumini, Moo Young Kim, and Seung Ryul Baek. 2016. Combining visual and acoustic features for music genre classification. Expert Systems with Applications 45:108–117.

[8] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 22(10):1533–1545.

[9] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, pages 776–780.

[10] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 .

[11] Tom LH Li, Antoni B Chan, and A Chun. 2010. Automatic musical pattern feature extraction using convolutional neural network. In Proc. Int. Conf. Data Mining and Applications.