MUSIC GENRE CLASSIFICATION AND ARTIST SIMILARITY RECOGNITION WITH MACHINE LEARNING

Pengzhan Jiang pjiang@kth.se
Qi Xiong qxiong@kth.se
Riccardo Cinà rcina@kth.se

1. ABSTRACT

From ancient times to the present, music has been an integral part of human history. From Shakespeare's plays to Beethoven's Symphony No. 5, from the Vikings battle cries in the Middle Ages to the clashing sounds of water and birdsong in modern gardens, all of these demonstrate the charming mystery of music. People want to reveal the mask of music all the time, and now, with the help of MIR and machine learning tools, we can better analyze the characteristics of music and implement some interesting work.

In this report, we first use the MIR tool to extract music feature data from the audio files of the GTZAN data set, and then use machine learning and deep learning tools to classify the data set. The models used include DT, SVM, PCA, KNN and CNN. Finally, each model is estimated by the accuracy and confusion matrix, and we reason and draw conclusions based on the results.

In addition, we are also very interested in music similarity analysis, which can be widely used in music recommendation systems and friend-searching apps. We manually selected 8 pieces of classical music and 2 pieces of rock music, and input their MFCC information into Euclidean distance calculating model for similarity and comparison analysis, proving that our model has the ability to analyze the similarity of music and it can be used to attain possible music recommendation applications. And we tried it on three popular singers in China.

2. INTRODUCTION

In order to implement machine learning for music classification, we need a dataset to train our model. Our work is based on a dataset called GTZAN, which offers a diverse collection of audio clips spanning various music genres, making it a go-to resource for tasks like music genre classification and feature extraction. It contains 1000 sound clips, categorized into 10 genres with 100 songs each, and every individual clip has the same duration of 30 seconds. Each sound clip has a label called genre, and it is possible to classify these sound clips using different machine learning and deep learning methods.

3. METHOD

The method we implement comes from the textbook [1] of music informatics and from other machine learning courses. We used basic machine learning methods like Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbor classifiers (KNN), and Principal Component Analysis (PCA). In the end, we see the result of adopting Convolutional Neural Networks (CNN).

3.1 Feature extraction and visualization

Librosa provides the building blocks necessary to create music information retrieval (MIR) systems for music and audio analysis, which can be used to extract a set of features such as chroma, spectral power, spectral centroid, zero crossing, harmony and MFCC.

We use these features as input to our different models to correctly classify the sound clips.

3.2 Genre classification

The classifiers that we used are described in details below.

3.2.1 Decision tree

A decision tree is a graphical representation of a decision-making process used in machine learning. It breaks down a complex decision into a series of simple, binary choices (nodes), resulting in a tree-like structure. Each node represents a decision, and each leaf node signifies an outcome or classification.

One of the most important concepts in a decision tree is the information gain, which is obtained by the entropy decrease after the node. The information gain of an attribute A, relative to a collection of examples S is defined as 1.

$$Gain(S, A) = Entropy(S) - \sum_{k \in value(A)} \frac{|S_k|}{|S|} Entropy(S_k) \quad (1)$$

Where S_k is the subset of examples in S where the attribute A has the value k. The question node which can provide biggest information gain will be chosen and remained in pruning.

3.2.2 Support Vector Machine

A Support Vector Machine (SVM) is another useful tool in machine learning for classification and regression. It aims to find an optimal hyperplane that maximizes the margin between different classes of data points. It is effective in high-dimensional spaces and can handle both linear and non-linear data from low-dimension to high-dimension with suitable kernel functions.

Basically, SVM can be described as follows:

$$minimize\ f(\vec{w}) = \frac{||\vec{w}||^2}{2}$$
$$subject\ to\ y_i \times (\vec{w} \cdot \vec{x_i} + b) \geq 1,\ for\ \forall i \quad (2)$$

Usually, the dual form of the problem is easier to solve, which is to find the values of the Lagrange multiplier which can minimize the objective function under certain boundaries and constraints. A suitable order of polynomial kernel function can handle data that is not easily separable, and slack variables can control the soft margin to make a trade-off between model complexity and accuracy.

3.2.3 PCA

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional representation by identifying the most important features or components while preserving as much variance as possible.

PCA can give us a way to reduce features, which helps eliminate redundant or less informative features, thus reducing computational complexity and improving the classifier's performance by reducing overfitting.

3.2.4 KNN

K-Nearest Neighbors (KNN) is a supervised learning method that classifies data points by considering the majority class among their k-nearest neighbors in the feature space.

KNN needs no assumptions about the underlying data distribution, making it suitable for various types of datasets, like complex, non-linear decision boundaries. However, KNN's performance may be sensitive to the choice of how many neighbors are taken into consideration, and it can be computationally expensive for large datasets due to the large number of distance calculations for each point.

3.2.5 CNN

Convolutional Neural Networks, often abbreviated as CNNs, are a powerful class of artificial neural networks designed primarily for processing and analyzing visual data.

A convolutional neural network consists of an input layer, a hidden layer and an output layer. These deep learning models, known for their feature extraction capabilities, can be adapted to process and classify audio data effectively. By treating audio spectrograms, which represent sound signals in a visual form, as images, CNNs can identify unique patterns, timbre, and rhythm features that are crucial in discerning music genres.

This technology has opened new avenues for automating the categorization of music, aiding music recommendation systems, and enhancing our understanding of musical content through data-driven approaches. CNNs, with their ability to extract intricate patterns from audio data, have become a valuable tool in the evolving field of music genre classification.

In this project, we use the Mel Spectrogram of the audio as input, and the output is the classified music genre. The network has three hidden layers: the activation function is a ReLU function and the output layer activation function is a softmax function.

3.3 Similarity Analysis

After correctly classifying the music genre of the clips, the next problem to be addressed is the categorization of music under the same genre, as each artist has a different compositional style. In order to find out who is the most probable artist who composed the songs, we use the MFCC of different songs to analyze the similarity and we aim to explain for their relationships, with the goal of building music recommendation applications.

4. RESULTS AND DISCUSSION

4.1 Classification Accuracy

The feature distribution and BMP of different genre are shown in Fig.1 and Fig.2.

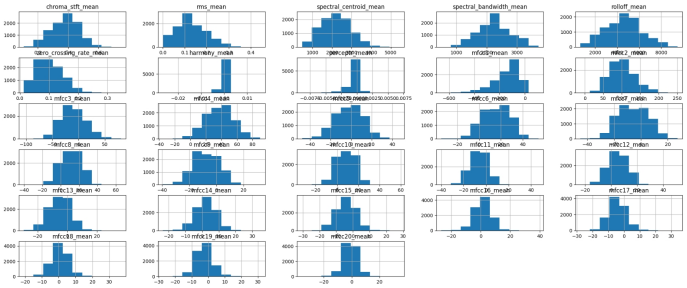

Figure 1. Feature distribution of different genre.

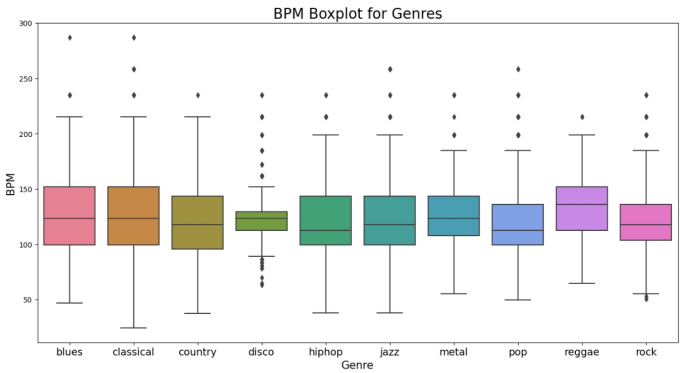

Figure 2. BMP distribution of different genre.

The accuracy for different models in Table.1 are demonstrated. In our models, KNN for feature data of 3 seconds has the highest classification accuracy. And CNN has the best accuracy for image data(spectrogram) of audio files.

	30 second csv	3 second csv	Mel spectrogram
CNN	-	-	**56.25%**
PCA	-	**61.79%**	-
KNN	**62.00%**	**80.98%**	**26.00%**
DecTree	**63.50%**	**65.17%**	**32.50%**
SVM	**69.00%**	**68.61%**	**44.50%**

Table 1. Accuracy for different machine learning models.

It is interesting to see the confusion matrix in Fig.3, which gives us information on which genre is the hardest to classify. It was achieved through the model generated by the

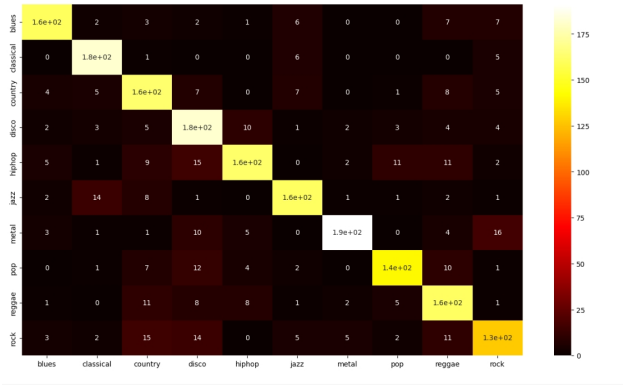

Figure 3. Confusion matrix of ten genre.

best classification algorithm, which is KNN. The (i, j) cell is equal to the number of observations known to be in group i and predicted to be in group j.

From the figure of the confusion matrix of ten genres, we can see that our model has difficulty predicting rock. This doesn't conflict with what we know, because within the genre of rock, there are many sub-genres such as blues rock and country rock that have been merged into the genre of rock, making it more challenging to categorize them.

At last, let's take a look at the CNN accuracy for each class in Fig.4. In this project, we are using a CNN containing 3 hidden layers, where the batch size is 32 and the epoch numbers are 10.

From the results in Table.1, it seems that the classification result is not satisfactory. However, we can find out through the CNN training results graph in Fig.4 that the CNN classification accuracy is gradually increasing with the increase in the number of epochs. It is reasonable to infer that as the epochs increase, the classification accuracy will get closer to the theoretical value, but the time consumed will also become larger and larger. In this project, the time for one epoch is close to 1200 s.

```
Epoch 1/10
25/25 [==============================] - 1161s 47s/step - loss: 2.4234 - acc: 0.2300 - val_loss: 2.0834 - val_acc: 0.2240
Epoch 2/10
25/25 [==============================] - 1153s 47s/step - loss: 1.5754 - acc: 0.4387 - val_loss: 1.5434 - val_acc: 0.4635
Epoch 3/10
25/25 [==============================] - 1147s 46s/step - loss: 1.1294 - acc: 0.6250 - val_loss: 1.4677 - val_acc: 0.4583
Epoch 4/10
25/25 [==============================] - 1147s 46s/step - loss: 0.9444 - acc: 0.6787 - val_loss: 1.4575 - val_acc: 0.5573
Epoch 5/10
25/25 [==============================] - 1056s 42s/step - loss: 0.8679 - acc: 0.7038 - val_loss: 1.5162 - val_acc: 0.5417
Epoch 6/10
25/25 [==============================] - 1056s 43s/step - loss: 0.7867 - acc: 0.7487 - val_loss: 1.3302 - val_acc: 0.5573
Epoch 7/10
25/25 [==============================] - 1064s 43s/step - loss: 0.6224 - acc: 0.7987 - val_loss: 1.5620 - val_acc: 0.4948
Epoch 8/10
25/25 [==============================] - 1072s 43s/step - loss: 0.5288 - acc: 0.8300 - val_loss: 1.5919 - val_acc: 0.5521
Epoch 9/10
25/25 [==============================] - 1119s 45s/step - loss: 0.3839 - acc: 0.8950 - val_loss: 1.4828 - val_acc: 0.5521
Epoch 10/10
25/25 [==============================] - 1121s 45s/step - loss: 0.2625 - acc: 0.9175 - val_loss: 1.4065 - val_acc: 0.5625
```

Figure 4. Accuracy for each class using CNN.

4.2 Similarity Prediction

Similarity is an enhanced goal of our project. We train our model with nine jazz songs and one hip-hop song; the difference is quite clear in Fig.5. We used the MFCC similarity matrix to calculate the similarity between different songs. For each pair of these sound clips, we compute the average Euclidean distance in mfcc, store it in a matrix, and display it. The darker the colour, the smaller the distance between the features of different sound clips. The

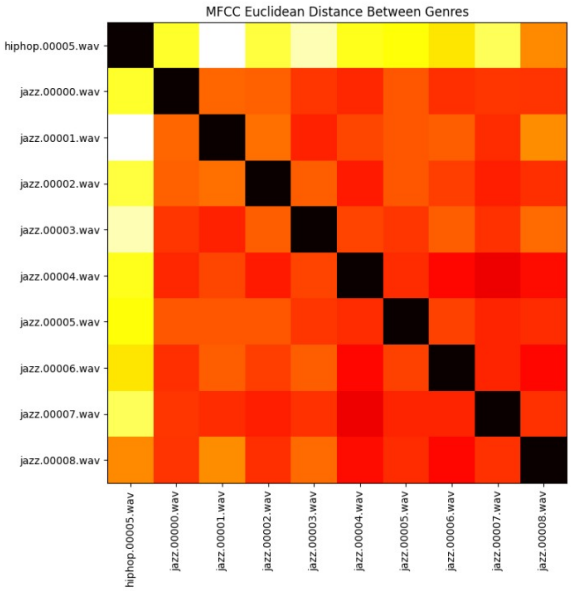

Figure 5. Similarity analysis.

main diagonal has the darkest colour, black, as it represents each clip being similar to itself. From Fig.5, we find that there is indeed a similarity in jazz; from the figure, it is the 9×9 matrix in the bottom right corner that is darker in colour, whereas hip-hop and jazz do not have much similarity, corresponding to the lighter coloured segments. In summary, we know that the MFCC similarity matrix can indeed represent the similarity between songs, so we can analyse the creative style of each artist by calculating the MFCC similarity matrix of different artists' songs.

Then we take into consideration three Chinese singers, two from Taiwan and one from Mainland China, and analyze the similarity among their songs in Fig.6. The three singers are Jay Chou (Taiwan), Wu Bai (Taiwan), and Jason Zhang (Mainland China). We decided to use five different music clips of Jay Chou, one music clip of Wu Bai and one music clip of Jason Zhang; these artists have been chosen to clearly show the results of our work and what we mean by artist similarity recognition.

As can be seen, the similarity matrix is able to spot paramount relationships between them that can be attributed to their interactions in real life. In fact, Wu Bai, who was a former member of the music industry, mainly wrote blues and metal songs that inspired Jay Chou who comes from a younger generation. Due to this, Jay Chou was highly influenced by Wu Bai and in his performances, this can be seen from the very similar style he has. This is well demonstrated in Fig.6, where the blocks between Wu Bai and Jay Chou's tracks are darker in colour.

As a mainland singer, Jason Zhang's song style tends to be pop music, and from the chart, it is less similar to Wu Bai and Jay Chou's music, as highlighted by the lighter colours in the colour blocks.

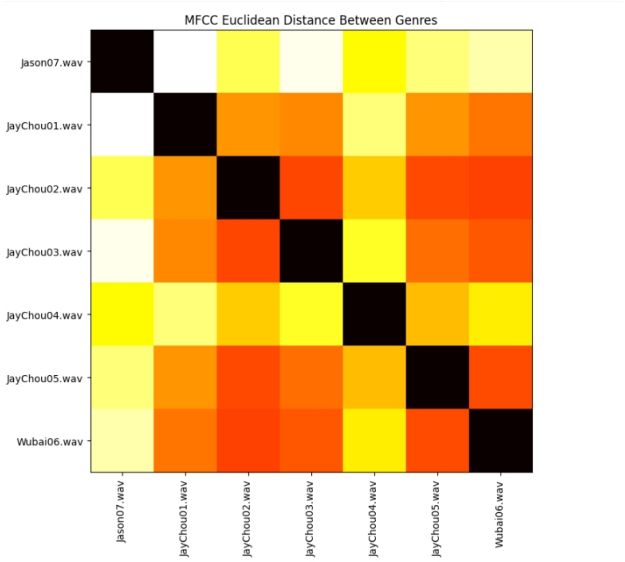

Figure 6. Similarity between two singers and difference between another singer.

5. CONCLUSION

In this project, we extracted music features to classify genres of musical clips with the help of two tools, MIR and machine learning, and compared the classification accuracy of different machine learning models. After training the models, we show the confusion matrix under the best classification model, which is a good indication of which genre model is the most difficult to predict.

Furthermore, after completing the previous steps, we conducted an in-dept analysis of music similarity by selecting and analysing three artists and their songs. By using MFCC's Euclidean distance for similarity comparisons, it was proven that our model not only classifies the clips according to their different genres but also based on their similarity; this is an important result as it could be relevant for music recommendation systems and, more broadly, also for friend recommendation in online applications.

6. REFERENCES

[1] M. Müller, *Fundamentals of Music Processing*. Springer-Verlag, 2015.