Musical Instrument Classification via Low-Dimensional Feature Vectors

Zishuo Zhao¹, Haoyun Wang²

¹²Institute for Interdisciplinary Information Sciences, Tsinghua University
wingedkuriboh@126.com, whymahina@163.com

Abstract

Music is a mysterious language that conveys feeling and thoughts via different tones and timbre. For better understanding of timbre in music, we chose music data of 6 representative instruments, analysed their timbre features and classified them. Instead of the current trend of Neural Network for black-box classification, our project is based on a combination of MFCC and LPC, and augmented with a 6-dimensional feature vector designed by ourselves from observation and attempts. In our white-box model, we observed significant patterns of sound that distinguish different timbres, and discovered some connection between objective data and subjective senses. With a totally 32-dimensional feature vector and a naive all-pairs SVM, we achieved improved classification accuracy compared to a single tool. We also attempted to analyze music pieces downloaded from the Internet, found out different performance on different instruments, explored the reasons and suggested possible ways to improve the performance.

Index Terms: timbre modeling, multi-class classification

1. Objectives and motivations

Music is an art of the sound, which can be analyzed by a variety of tools on its waveform. On the other hand, it is a language of emotions and thought, with psychological senses on every note and melody.

With the basic knowledge about sounds, we may notice that synthesis does make sense in some way. The smooth sine wave sounds plain and soft, the square wave sounds rigid, and the sawtooth wave sounds sharp. It is not easy to make a complete mapping between waveforms and human senses, but it is still an enchanting job to find out the features that make timbres different and tell them apart.

In our project, we adopted several speech analysis tools and observed their results, did classification to examine their soundness to indicate timbre features, and attempted to design new models based on them for further exploration and improvement of performance.

2. Background and previous work

As music and speech share some characteristics in common, speech analysis tools as MFCC and LPC, etc are also applicable, though they do have some innate difference. MFCC also performs well in extracting features in popular music. Some simple statistics of the spectrum, e.g. spectral centroid, spectral flatness also contributes to the recognition.

For multi-class classification, it can be implemented by methods that supports it directly, e.g. Gaussian Mixture Model(GMM), or by a set of binary classifiers, e.g. Support Vector Machine(SVM). Two basic methods are one-to-all and one-to-one, and due to the hierarchical families of musical instruments, hierarchical classification also works.

Generally, music timbre is a high-dimensional feature, so Neural Network can also extract these features well.

3. Methodology

For all the features that are extracted, the window length is about 100ms.

3.1. Feature extraction

3.1.1. MFCC

The MFCC(Mel-frequency Cepstrum Coefficients) describes how energy of the given audio spread across Mel-scale frequency domain. The Mel-scale is based on human hearing test and is widely used in speech/music signal processing. We use standard MFCC with 12 coefficients, and filter bank between 0Hz and 0.5fs.

3.1.2. Normalized LP Coefficient

The LPC(Linear Prediction Coefficients) serves as a linear filter, through which the original wave were supposed to be eliminated to 0. We use steepest descent to obtain the 13-dimensional LP-coefficient \( c_0 \sim c_{12} \) \( (c_0 = 1) \), and then normalize them in order to make the coefficients of the same instrument closer to each other and to get an extra “LPC Magnitude” feature that in practice distinguishes instruments in a way. The normalization is:

\[
    c_i' = \frac{c_i}{\sigma_c}, \quad i = 0, 1, \quad (1)
\]

where

\[
    \sigma_c = \sqrt{\frac{\sum_{i=0}^{12} (c_i - \tau)}{13}} \quad (2)
\]

3.1.3. Envelop of log-spectrum & Linear regression

The envelop of the log-spectrum (i.e. spectral outline) consist of the low-frequency components of the DCT(Discrete Cosine Transform), denoted by \( y = c(\omega) \), \( \omega \in [0, f_s] \). Having perceived that the envelop below 10kHz can represent the characteristic of the music, we do linear regression to the envelop...
from the first peak $f = \omega_0$ to $\omega_0 + 10kHz$:

$$(\hat{a}, \hat{b}) = \operatorname{argmin}_{\omega \in [\omega_0, \omega_0 + 10k]} \sum (e(\omega) - \hat{a}\omega - \hat{b})^2.$$  

(3)

The average slope $\hat{a}$ represents how fast the magnitude drops when the frequency increases in the spectrum.

We also compute the mean square error which represents how the envelop fluctuates around the regression function:

$$\sigma = \sqrt{\sum_{\omega \in [\omega_0, \omega_0 + 10k]} (e(\omega) - \hat{a}\omega - \hat{b})^2}.$$  

(4)

### 3.1.4. Cepstrum peaks

The cepstrum coefficients describe how the formants are located or the envelop of the spectrum. Here we take the first 4 peaks of the cepstrum and record their magnitude. This feature represents how much the log-spectrum varies with frequency, and also the ratio of magnitude between peaks and troughs in the spectrum.

### 3.2. Classification

For this part, we use SVMs (Support Vector Machines) to classify the data. The basic idea is to apply a round match between every pair of instruments. Since we have 6 different types, we train $\binom{6}{2} = 15$ SVMs, each tries to distinguish between a pair of instruments. When we are going to predict which type a new window of sound wave belongs to, we run a round match among the 6 types, and predict the window to be the type that wins the most.

The training data we have are unbalanced, which is, the amount of data we have in different types are different. To solve this, we add weight $w_t$ to type $t$,

$$w_t = \frac{1}{N_t},$$  

(5)

where $N_t$ is the size of data in type $t$. Then, the model can be balanced.

### 4. Experimental set-up

#### 4.1. Instruments & Data source

For this experiment, we use notes from 6 instruments: acoustic guitar, Alto saxophone, flute, piano, trumpet and violin. The data is from RWC music database.

The source code is on [https://github.com/wiku30/SST](https://github.com/wiku30/SST).

#### 4.2. Train Data & Test Data

Our train data is selected randomly (50%) from the normal form notes played in all instruments. The pitches go from the lowest possible one to the highest possible one for every instrument.

We carried out two tests. One is based on all the remaining normal form notes, the other is based on all notes in the dataset, including staccatos, interchanging notes and so on. The results will be shown in the next section.

### 4.3. Test on real Music

When applying our model on real music episodes, we make predictions every 100ms, and find the most frequently predicted instrument in the past 1 second as the final prediction. In this way, the system provides a real-time prediction, which can adapt itself to the changing of instruments.

### 5. Results

#### 5.1. Cross validation on single notes

The overall classification accuracy on basic dataset is 91% in 50/50 cross validation, and the confusion matrix is shown in Figure 1.

![](https://example.com/confusion_matrix.png)

Figure 1: Confusion matrix on basic dataset.

To show the contribution of each set of features, we did partial-feature experiments in which some features are disabled in the classification. The accuracies are shown in Figure 2.

![](https://example.com/accuracy_features.png)

Figure 2: Accuracy on partial features.

It shows that every feature contributes to the accuracy. For a single set of feature, MFCC performs the best, indicating that it is truly a promising tool for music recognition, even though designed for speech analysis. Our designed SO&CP does not work very well singly (yet it has only 6 dimensions), but it does decrease the error rate by roughly 1/4 when combined with other tools. Generally, the 32-dimensional feature vector is fairly compact compared to the complexity of music signals, and this accuracy is achieved with the least machine learning tools—only naive SVM classifier. It shows that we managed to extract the representative features that tells different timbre apart.

#### 5.2. Real music experiments

##### 5.2.1. Discovery and analysis

When tested on real music, the accuracy is generally lower, as expected. In real music, variations of playing styles, interference of multiple instruments, pitch change within one window, etc, will all increase the difficulty of recognition and classification.
To our notice, we observed that music pieces played by the piano, violin and guitar are generally well classified, worse but acceptable for the flute, while the saxophone and trumpet are poorly recognized.

Via listening to different pieces of trumpet and saxophone music, we noticed that their timbres vary with playing style and pitch more than the other four, and change significantly in one note with time. We conjectured that because we did not analyze temporal features in this project, and had not included special playing styles in our basic dataset, the model failed to recognize them well.

5.2.2. Analysis on augmented datasets

For analysis of this problem, we added music materials of special playing styles into the test set, and resulted in a sharp decrease of accuracy in cross validation. The confusion matrix is shown in Figure 3.

![Figure 3: Confusion matrix on augmented test set.](image)

It shows that the trumpet and the saxophone are worst recognized, which confirmed our conjecture.

In attempts to ameliorate this problem, we tried adding the special styles into the datasets and do cross validation, but the performance is not improved either (as in Figure 4), probably as a result of complicated relations between instruments and timbres.

![Figure 4: Confusion matrix on augmented dataset.](image)

From both Figure 3 and Figure 4, we can see that the SVMs incline to confuse sax with violin, and trumpet with violin. It is mentioned before that the features of the sound of a sax or a trumpet can vary greatly, so probably the distribution of sax or trumpet features surrounds that of the violin, and makes these instruments unlikely to be distinguished by the SVM, since SVM is a linear classifier.

One possible solution is to regard different playing styles as different instruments, but this will increase the number of classes significantly, resulting in a sharp increase of running time and the expected number of mis-classifications, which leads to a decrease on the accuracy for other instruments.

6. Future work

One of the shortcomings of our work now is that the model is sensitive to noises. Even we artificially add a white noise which takes up to 1% of the total energy, it would affect the outcome of our prediction. This is a problem that should be dealt with in our future work.

Another improvement can be made by adding temporal analysis to the features we extracted from waveform, such as the derivative of MFCC, the onset duration, and the changing rate of the amplitude envelop of the waveform. Hopefully these can further increase the accuracy of our model.
7. References

[1] A. Eronen, “Automatic musical instrument recognition,” Master’s thesis, 2001.

[2] C.-M. Mak, T. Lee, S. Senapati, Y.-T. Yeung, and W.-K. Lam, “Similarity measures for chinese pop music based on low-level audio signal attributes,” ISMIR 2010, pp. 513–518, 2010.

[3] X. Zhang and Z. W. Ras, “Analysis of sound features for music timbre recognition,” International Conference on Multimedia & Ubiquitous Engineering, 2007.

[4] J.-W. Lee, S.-B. Park, and S.-K. Kim, Music Genre Classification Using a Time-Delay Neural Network. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 178–187.