Predominant Musical Instrument Classification based on Spectral Features

No name given
Indian Statistical Institute
Kolkata, WB 700 108

Abstract

This work aims to examines one of the corner stone problem of Musical Instrument Retrieval in particular instrument classification. We have presented a very concise summary of past work in this domain. We have included most of code (jupyter notebook) for easy reproducibility.

1 Introduction

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

1.1 Motivation of our work

1.2 Related Works

2 Results & Discussion

2.1 Our Contribution

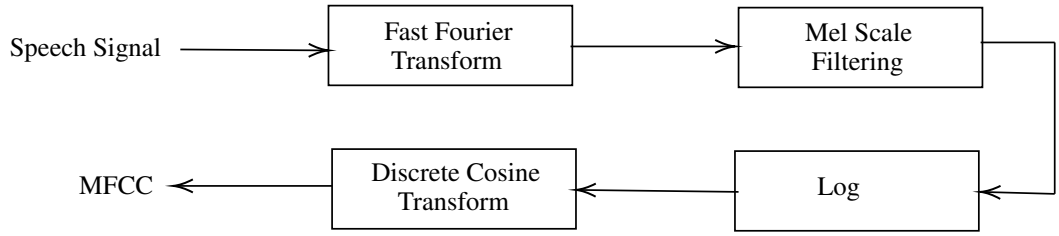
In solving any classification problem of audio and video file the most important thing is to choose how to extract features from given audio/video files. While dealing with our audio dataset we found that despite having the same notes of sound, spectrogram differs based on which instrument is playing that note. As an illustration we recorded the same note with 4 different instruments and generated the corresponding spectrograms in Figure 1. It is evident that we can use this property of spectrogram to predict instrument used while playing particular sound excerpt.

According to Eronen and Klapuri [3] Timbre, perceptually, is the 'colour' of a sound. Experiments have sought to construct a low-dimensional space to accommodate similarity ratings. Efforts are then made to interpret these ratings acoustically or perceptually. The two principal dimensions here are spectral centroid and rise time. Spectral centroid corresponds to the perceived brightness of sound. Rise time measures the time difference between the start and the moment of highest amplitude.

Deng et al. [2] have shown instruments usually have some unique properties that can be described by their harmonic spectra and their temporal and spectral envelopes. They have shown only first few coefficients are enough for proper classification.

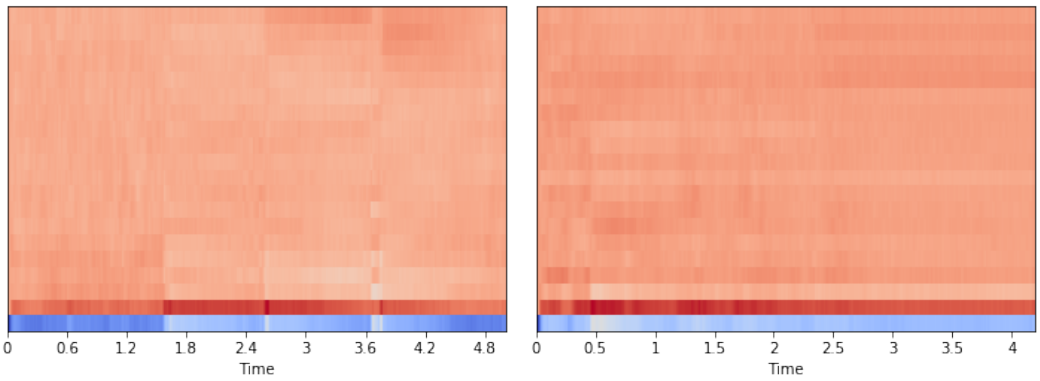
To do the spectral analysis MFCC is best choice. According to Wikipedia[7] "the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear

cosine transform of a log power spectrum on a nonlinear mel scale of frequency". Mel is a number that links to a pitch, which is analogous to how a frequency is described by a pitch. The basic flow of calculating the MFC Coefficients is outlined below

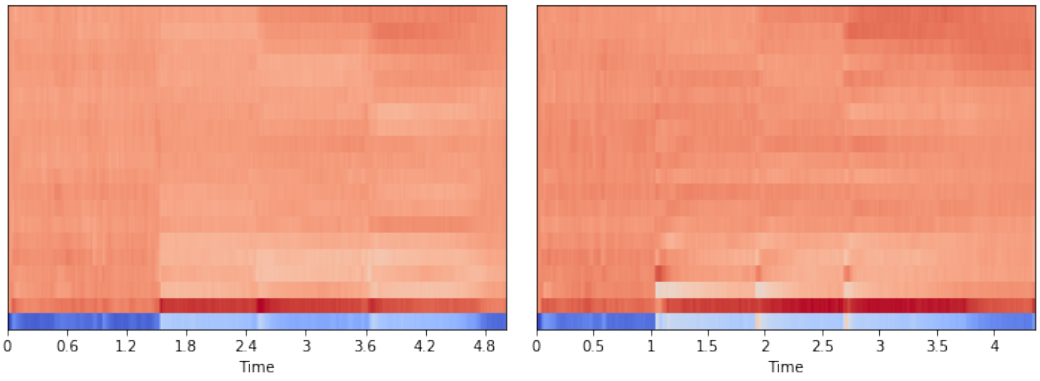


The mathematical formula for frequency-to-mel transform is

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right).$$



(a) EDM (b) Guitar (Accoustic)



(c) Key Board (d) Organ

Figure 1: Same note (audio) played in various instruments and their spectograms

MFCCs are obtained by transforming frequency (hertz) scale to mel scale. Typically, MFCC coefficients are numbered from the 0th to 20th order and the first 13 coefficients are more than enough to do any kind of audio classification task.

2.2 Dataset

We used IRMAS(Instrument recognition in Musical Audio Signals)[1] dataset in our study as this is a polyphonic dataset so we can use it train a robust classifier. The data consists of `.wav` files of 3 seconds duration of many instruments, about eleven. We have chosen six of these instruments (viz. flute, piano, trumpet, guitar, voice and organ) for recognition. Our data has 3846 samples of music running into about three hours, giving sufficient data for training and testing purposes as well. In addition, the data consists of multiple genres including country folk, classical, pop-rock and latin soul. Inclusion of these multiple genres could lead to better training. The dataset has been downloaded from `https://www.upf.edu/web/mtg/irmas`. Number of audio samples per instrument class is reproduced in table 1.

Instrument	**Number of Samples**	**Clip Length (in sec)**
Flute	451	1353
Piano	721	2163
Trumpet	577	1731
Guitar	637	1911
Voice	778	2334
Organ	682	2046
Total	3846	11538 (3 hr 12 min)

Table 1: Instrument Samples and Clip Length

3 Methodology & Our Approach

3.1 Feature Extraction

Deng et al. [2] have shown that for achieving more accurate classification of musical instruments, it is essential to extract more complicated features in our analysis, apart from MFCC. Hence, we considered other features like Zero-crossing rate, Spectral centroid, Spectral bandwidth and Spectral roll-off during our feature extraction via Librosa. Zero Crossing rate indicates the rate at which the signal crosses zero. Spectral Centroid is a measure to indicate the center of mass of the spectrum being located, featuring the impression of brightness characteristic given a sound sample. Spectral bandwidth gives the weighted average of the frequency signal by its spectrum. Spectral roll-off features the frequency under which a certain proportion of the overall spectral energy belongs to.

To extract features from audio files we needed to use any library. We had two options – Librosa[4] and Essentia[5]. We tried with both of libraries.

Essentia is an open-source C++ based distribution package available under Python environment for audio-based musical information retrieval. This library computes spectral energy associated with mel bands and their MFCCs given an audio sample.

Windowing procedure is also implemented in Essentia for analyzing the frequency content of an audio spectrum by creating a short sound segments of a few milli-seconds for a relatively longer signal. By default, we used Hann window, a smoothing window typically characterized with not only having good frequency resolution but also reduced spectral leakage.

The audio spectrum is analyzed by extracting MFCCs based on the default inputs of hopSize (determines the hop length between frames) and frame size given. The default parameters for sampling rate is 44.1 kHz, the hop length (HopSize) 512 and frame size as 1024 in Essentia. This features thus extracted from manifold segments of a sample signal are aggregated with their mean and used as the features for each sample labeled with their instrument class.

On the other hand, Librosa — the Python package used for music and audio data analysis. The only distinction is the rate of audio sampling, where it is resampled at 22.05 kHz during load time.

Among both the libraries we prefered Librosa as it fetched more accuracy as well as being an older software package there was a lot of community support from various online forums.

3.2 Classifier Training

We have applied various machine learning techniques to perform classification of instruments.

MFCC features were extracted using librosa/essentia. We extracted first 13 MFCC features. We got 2048 $\times$ 13 matrix feature for a particular audio clip. We took mean of all the columns to get condensed feature providing us with 1 $\times$ 13 feature vector. As mentioned above 5 other features were also appended in that feature vector. Using the labelencoder function of scikit learn we labeled each vector with the instrument class it belonged to.

Supervised Classification Techniques We used different supervised classification techniques to identify the predominant musical instrument from the audio file. We used scikit learn[6] to train our classifer in a jupyter based python3 notebook. Initially we started with logistic regression and decision tree classifier. Classification trees are usually prone to overfitting, So it did not perform well on the test data. We also fit some bagging and boosting techniques on the mfcc and spectral features. We tried random forest to control the overfitting and with some parameter tuning, it provided us with the better classification. We also tried XGBoost on the same set of features and after some parameter tuning gradient boosting classified the instruments with an accuracy of around 0.7.

Support vector machine (SVM) was also used to fit the extracted features. It outperformed the other traditional classification techniques mentioned above. We used radial basis function kernel for this non linear classification. We also performed some parameter tuning for penalty parameter C and kernel coefficient gamma which improved the overall accuracy on test data.

We also designed a simple neural network to perform this classification. We used three layers with 30, 15 and 6 neurons on these layers respectively. We applied relu activation function on the first two layers and sigmoid on the last layer of the network. Neural network also performed better than most of the traditional techniques mentioned above.

In terms of accuracy bagging and boosting models such as random forest and XGboost performed better than traditional models such as classification trees and logistic regression. Finally SVM turned out to be a more accurate classifier than other techniques mentioned above.

3.3 Evaluation Criteria

The following evaluation metrics were used to judge the performance of the model

- The precision is the ratio $\frac{tp}{(tp+fp)}$ where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. Precision for various models are shown in a boxplot format see Fig: 3a.
- The recall is the ratio $\frac{tp}{(tp+fn)}$ where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. Fig: 3b shows illustrative visulization for various supervised model.
- The F1 score can be interpreted as a weighted average of the precision and recall.

$$F1 = \frac{2 * (\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$$

- Confusion Matrix is a technique to evaluate performance of a supervised classification. Calculating a confusion matrix gives a better idea of what our classification model is getting right and what types of errors it is making. Confusion Matrix for various models is shown in Fig: 2

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.
.

(a) Logistic Regression

(b) Decsion Tree

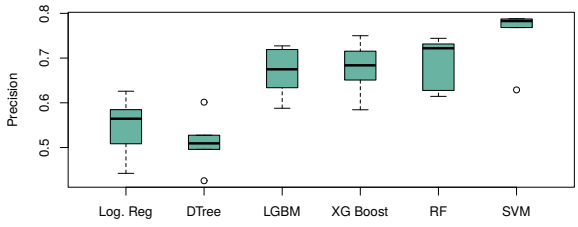
(c) Light GBM

(d) XG Boost

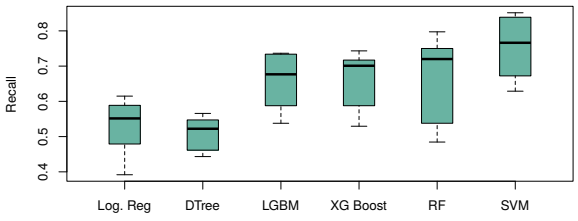
(e) Random Forest

(f) SVM

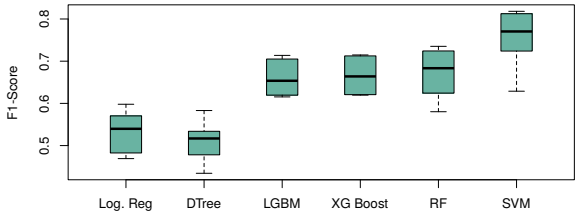
Figure 2: Confusion Matrix for various supervised Algorithms



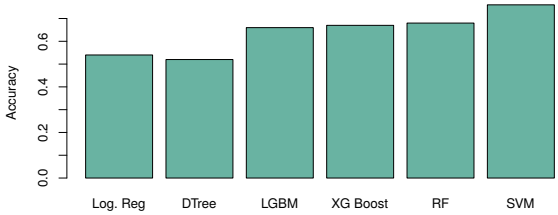
(a) Precision



(b) Recall



(c) F1 Score



(d) Accuracy

Figure 3: Evaluation Metric for Various Supervised Algorithms

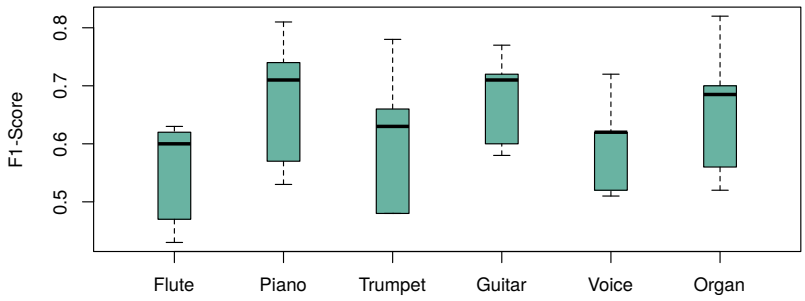


Figure 4: Instrument wise classification

	Logistic Regession			Decision Tree			LGBM		
Instrument	P	R	F1	P	R	F1	P	R	F1
Flute	0.58	0.39	0.47	0.43	0.44	0.43	0.66	0.59	0.62
Piano	0.55	0.59	0.57	0.53	0.54	0.53	0.69	0.73	0.71
Trumpet	0.44	0.53	0.48	0.50	0.46	0.48	0.59	0.67	0.63
Guitar	0.63	0.57	0.60	0.60	0.57	0.58	0.73	0.68	0.71
Voice	0.58	0.48	0.52	0.52	0.50	0.51	0.72	0.54	0.62
Organ	0.51	0.61	0.56	0.50	0.55	0.52	0.63	0.74	0.68
	XG Boost			**RF**			**SVM**		
Instrument	P	R	F1	P	R	F1	P	R	F1
Flute	0.66	0.59	0.62	0.72	0.48	0.58	0.63	0.63	0.63
Piano	0.72	0.71	0.71	0.72	0.75	0.74	0.79	0.84	0.81
Trumpet	0.58	0.69	0.63	0.61	0.72	0.66	0.78	0.77	0.78
Guitar	0.71	0.72	0.71	0.73	0.72	0.72	0.77	0.76	0.77
Voice	0.75	0.53	0.62	0.74	0.54	0.62	0.78	0.67	0.72
Organ	0.65	0.74	0.69	0.63	0.80	0.70	0.79	0.85	0.82

Table 2: Precision, Recall & F1 Score for various Supervised Models

References

[1] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *ISMIR*, pages 559–564, 2012.

[2] Jeremiah D Deng, Christian Simmermacher, and Stephen Cranefield. A study on feature analysis for musical instrument classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):429–438, 2008.

[3] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *Proceedings of the Acoustics, Speech, and Signal Processing, 2000. On IEEE International Conference - Volume 02*, ICASSP '00, pages II753–II756, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7803-6293-4. doi: 10.1109/ICASSP.2000.859069. URL `http://dx.doi.org/10.1109/ICASSP.2000.859069`.

[4] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[5] MTG upf. Essentia open-source library and tools for audio and music analysis, description and synthesis, 2019. URL `https://essentia.upf.edu/documentation/index.html`. [Online; accessed 23-November-2019].

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Wikipedia contributors. Mel-frequency cepstrum — Wikipedia, the free encyclopedia, 2019. URL `https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=917928298`. [Online; accessed 23-November-2019].