Machine Learning Driven Recognition and Smart Classification of Audios of Bharatanatyam Dance

K.V.R.K.Vivek

March 30, 2020

1 Problem Statement

We are given audio which is a periodic signal comprising of stick beating sound and the human voice (called as bols/syllables). There were no musical instruments used. We need to recognize and classify the audio. In the data set there are 6 set of audios which have 15 distinct audios.

2 Motivation

- Understanding the underlying semantics of performing arts like dance or music is a challenging task. Capturing and analyzing their multimedia content is useful for preservation of cultural heritage, to build recommendation systems, to assist learners use tutoring systems etc.
- India has a rich tradition of classical dance. Bharatanatyam is one of the eight Indian classical dance forms.

In Bharatanatyam the dancer performs in sync with a specific form of structured rhythmic music, called Sollukattu. Sollukattu is generated by instrumental strikes along with vocal utterance. It comprises of :-

- Stick Beats: A Full Beat or 1-beat defines the basic unit of time. It is an instance on time scale generated by instrumental strikes.

- Bols: It is the vocal utterance. It may accompany a stick beat.
- Silence Beats: It is the period when there is nothing happening ,i-e, everything is silent.

The video part of the Bharatnatyam is called as Adavu.

3 Previous Work

NAME OF THE RESEARCH PAPER	YEAR OF PUBLICATION	JOURNEL NAME/ CONFERENCE	DATASET
Neural Network Based Indian Folk Dance Song Classification Using MFCC and LPC	2017	International Journal of Intelligent Engineering and Systems	A collection of 40 songs is prepared for each folk dance. Proposed approach is evaluated using 160 popular songs.
Audio feature extraction and analysis for scene segmentation and classification	1998	Journal of VLSI signal processing systems for signal, image and video technology	Audio clips from TV programs containing five classes. news reports, weather forecasts, TV commercials, live basketball and football games.The training data set has 400 clips from each scene class and the remaining 1000 clips in each class form the testing data set
Singing Voice Detection in North Indian Classical Music	2008	National Conference on Communications (NCC)	Music starts at a low tempo where only voice and Tanpura are present. The table strokes initially are widely apart in time but later become rapid. There are seven different North Indian classical vocal performances spanning 23 minutes in total.

Figure 1: Table 1

Table 1 compares three Research papers which are related to our work

In the next page we will see another table briefly describing the solution provided by them to their problems.

Name of the Paper	Brief Description of Solution
Neural Network Based Indian Folk Dance Song Classification Using MFCC and LPC	13 MFCC features and 13 LSP features are extracted from each segment. They have selected frame size of 15 ms and frame shift as 10 ms. Vector is generated by combination of 13 MFCC coefficients with 13 LPC coefficients. Then the dimensions of the feature vector is reduced from (25x499)x26 to 2000x26. 3 different classifiers are used namely, kNN, Naive Bayesian method and Neural Networks and based on majority voting label is assigned.
Audio feature extraction and analysis for scene segmentation and classification	The linear separability of different classes under the proposed feature space is examined using a clustering analysis. The effective features are identified by evaluating the intracluster and intercluster scattering matrices of the feature space. Using these features, a neural net classifier was successful in separating the above five types of TV programs
Singing Voice Detection in North Indian Classical Music	Spectral subtraction is used for noise subtraction. 7 features are extracted from 40ms window for each frame (with shift of 20ms). Spectral roll-off, Harmonic energy, Sub-band energy ratio 1, Sub-band energy ratio 2, Sub-band flux, Audio spectral flatness, Average sub-band energy are the 7 different features. The two features used for segmentation are sub-band flux and sub-band energy ratio. GMM was used as a classifier.Two different methods are used for classification namely, majority voting(MV) and Log Likelihood.

Figure 2: Table 2

(There are some more papers which I have studied but they have very less relevance with our problem, hence didn't mention here)

Now we will look about the Data set of our problem.

4 Dataset

Number of classes we have in the dataset are 15. These are namely *Tatta, Natta, Utsanga, Tirmana, Tei Tei Dhatta, Sarika, Pakka, Paikkal, Joining, Katti/Kartari, Kuditta Nattal, Mandi, KudittaMettu, Kuditta Tattal,* and *Sarrikkal.*

There are 6 sets of audios each which are sang by different people and in each set the same person is made to sing 3 times to get hold of the variations.

Each class has a unique bol sequence. The rules for composing a Sollukattu are defined in Bharatanatyam. The challenge, here, is to capture the rule set in the classifier and recognize an unknown Sollukattu using the trained classifier.

Now let's analyse how a particular Sollukattu :- Natta looks like . *tei yum tat tat tei yum ta* Here is the amplitude vs time plot of the same.

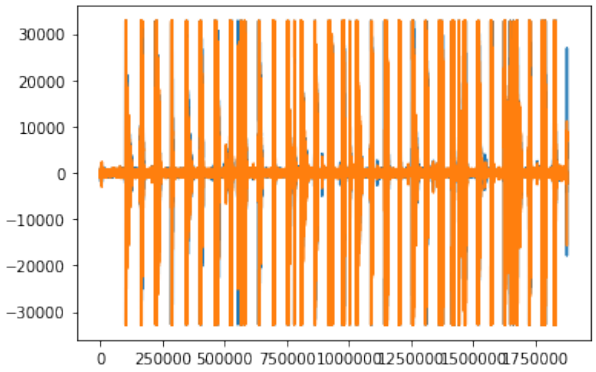

Figure 3: Amplitude vs Time graph of NATTA

We can observe that the audios are like discrete chunks/patches and we will use this feature later.

5 Abstract Approach

A brief outline of the approach that we are going to follow is :-

- Split each song of each class to its bols .
- Label each bol with its corresponding class.
- Now each data point of a dataset consists of a sequence of bols labeled by its class.

We will use this data to train our model

6 Splitting a song to its *bols*

We will split the single audio file into multiple audio files such that they contain neither silence beats nor stick beats . The approach we will follow :-

- Split the audio based on silence beats. So now we are left with small audio packets which may be pure stick beat, pure bol or a mixture of stick beat and bol.
- The packet containing only stick beat is unnecessary for us and needs to be removed.
- This can be done by checking the time period of the packet. Stick beats occur for a very small duration of time as compared to bols, hence we can remove the stick beats by keeping threshold on the time duration.

For splitting with respect to silence bits we will use the *split_on_silence* function from the *pydub* library which is as follows

.

$$chunks = split\_on\_silence(song, min\_silence\_length, silence\_threshold)$$

Where song is the audio file, min_silence_length is the silence length we require to consider it as silence, silence_threshold is the dB value which defines silence. All the small parts of the original audio will be stored individually in chunks

Now from the set of all chunks we need remove the chunk containing only stick beat.

Lets analyse the amplitude vs time curve for any Sollukattu. Eg:- Tattimettu

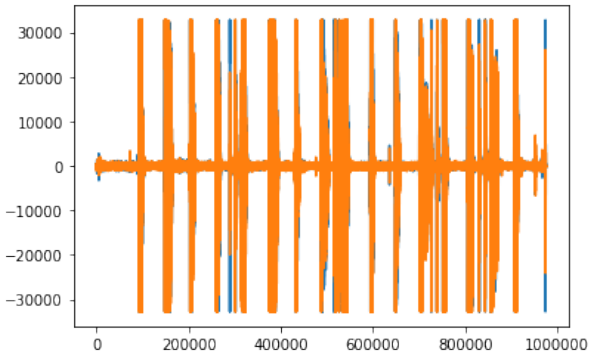

Figure 4: Amplitude vs Time graph of TATTIMETTU

We can see the chunks of audio packets in that audio. The chunk having smaller time periods are the stick bits and the bigger ones are the bols. Now we need to device a method to separate them .

We will find a threshold of time period such that all the chunks having a time period less than that threshold will be categorised as stick beats and will be removed from the data sequence.

How to get the threshold ?

We will calculate the mean of the time periods of the chunks and remove all the chunks having time period less than that of the mean (stick beats).

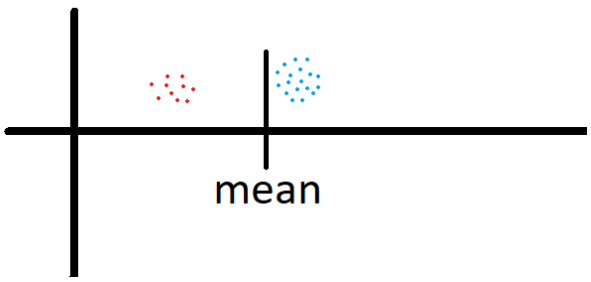

Figure 5: Beat Separation Technique

- It is obvious that this solution is prone to errors as the differentiating mechanism is vague.
- We can get more better results if use **clustering algorithms** like K-means clustering through which we can separate the clusters in a much better way and remove the cluster having smaller mean.

So now, we have splitted the song into sound signals containing only bols or bols+stick beats .

Now we should convert the signal into a form (representation) which is suitable for training purposes, which implies we should extract features from the signal.

We will use the following features :-

- Zero Crossing Rate
- Spectral Centroid
- MFCC - Mel Frequency Cepstral Coefficients

We will generate a high dimensional feature vector by combining them.

Zero Crossing Rate :- 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.

Spectral Centroid :- It indicates where the centre of mass for a sound is located and is calculated as the weighted mean of the frequencies present in the sound.

MFCC :- The Mel Frequency Cepstral Coefficients(MFCCs) of a signal are a small set of features (usually 10-20) which concisely describe the overall shape of a spectral envelope.

7 Training

Now we have chunks of audio music labeled with the class which it belongs to and containing either bols or a mixture of bols and stick beats.

We can have two methods for training our model :-

- Not preserving any sequence. Here we can use Machine Learning algorithms like XGBoost.
- Preserving the sequence of the bols. Here we should use structures like LSTMs or HMM.

8 Using Machine Learning Algorithms

- The perks of using this method are, since our original data set is small Machine Learning algorithms would be inefficient.
- By using features of each *bol* as a datapoint, number of datapoints are more and hence algorithms would do good.
- But the disadvantage is that we are not taking care of the ordering of bols in each Sollukattu .

The algorithm which we would use is XGBoost which is known to give good results irrespective of dataset. Later we can change the algorithm if we find any better one !

9 Using LSTM (Long Short-Term Memory)

Long Short-Term Memory (LSTM) Networks are a type of Recurrent Neural Network capable of learning order dependency in sequence prediction problems.

The steps we will follow to make our model is:

- Since the sequence of feature vectors is different for different audio clips, we must make all of them same for implementing LSTM on them.
- For that we will take LCM of number of bols in each song for all the songs and multiply all the audios to that length.
- By using this method we wont lose any information.
- We will use the Keras library for implementing LSTM.

The disadvantage is that since the data we have is less, model may not be trained properly.