4(b) Raga Recognition

Mahati Bharadwaj (IMT2014031) Email: Mahati.Bharadwaj@iiitb.org,
N. L. Amrutha (IMT2014036) Email: Lakshmi.Amrutha@iiitb.org

June 29, 2021

Contents

0.1 Introduction

The main purpose of this project is to automate the process of recognizing ragas in Indian classical music. We have focused on Indian Carnatic music. Ragas form the heart of Indian classical music. They consist of a unique set of sequences that are represented by variation in pitches and timings. Hence we have used these pitch variations and classified different snippets of songs into four different ragas. You can find the code and the input files of various ragas in this link: https://github.com/mahatibharadwaj/Raga-recognition

The four different ragas taken are Hamsadhvani, Hindolam, Mohana and Mayamalavagowla. In Indian Carnatic music there are mainly two types of ragas - Janaka and Janya. Janaka ragas are parent ragas that have all seven base notes in them. There are total 72 Janaka (or Melakarta ragas). The remaining ragas are child ragas (Janya ragas) which are derived from the Janaka ragas. We have considered the first raga that is taught in Carnatic music - Mayamalavagowla, which is one of the Janaka ragas. The Aarohana (ascending note sequence) and Avarohana (descending note sequence) of all the four ragas that we are considering are shown in Figure1.

Figure 1: Ragas with notations

0.2 Approach

We have followed two approaches:

0.2.1 Approach 1

The approach initially followed:

- Initially each song is divided into frames. Songs of 35 sec each are considered as input. Each song is divided into frames of 20 msec. Therefore there are total 1750 frames in a song.
- From each frame MFCC features are extracted and a feature vector of dimension 13 is obtained.
- The ragas Hamsadhvani, Hindolam, Mohana and Mayamalavagowla are assigned with labels 1,2,3,4 respectively. Each frame of the song is assigned its respective label. This is considered as the training dataset. This dataset is given as input to the Support Vector Machines, decision trees and a multi layered perceptron. The test accuracy is observed to be 35%, 38% and 45% respectively.

0.2.2 Approach 2

It is found that MFCC features are more appropriate to distinguish between different speakers. In our data, we have used songs of one singer and these songs are monophonic. These MFCC features almost remain same throughout the entire song because they depend mostly on the structure of the voice and not on the pattern of the song. Pitch features are more appropriate to find out the pattern or sequences in a song. These features are more appropriate for Raga recognition because ragas consist of different set of sequences. We considered only pitch features since we used songs of only one singer. The following approach is finally followed:

- Songs of length 3 mins each are taken as input.
- Each song is divided into frames of 20msec and a pitch feature is extracted from each frame. There are total 9000 frames in a song.

- For pitch feature extraction two algorithms are used. One of them is HPS algorithm. This algorithm gives more accurate results for instrumental music. Therefore another algorithm called auto-correlation is used.The dimension of pitch feature vector obtained from autocorrelation is number of frames in a song.

- The pitch feature vector of each raga is given as input to the Gaussian Mixture Models and a 36 dimensional mean vector is obtained for each raga. From the test frame pitch feature vector is obtained and the pitch feature vector is given to GMM.

- Consine similarity is calculated between the mean vectors of the training data and the testing data and based on the similarity score the frame is assigned to that raga.

0.3 Architecture

Figure2 below shows the final workflow we have followed to recognize ragas.

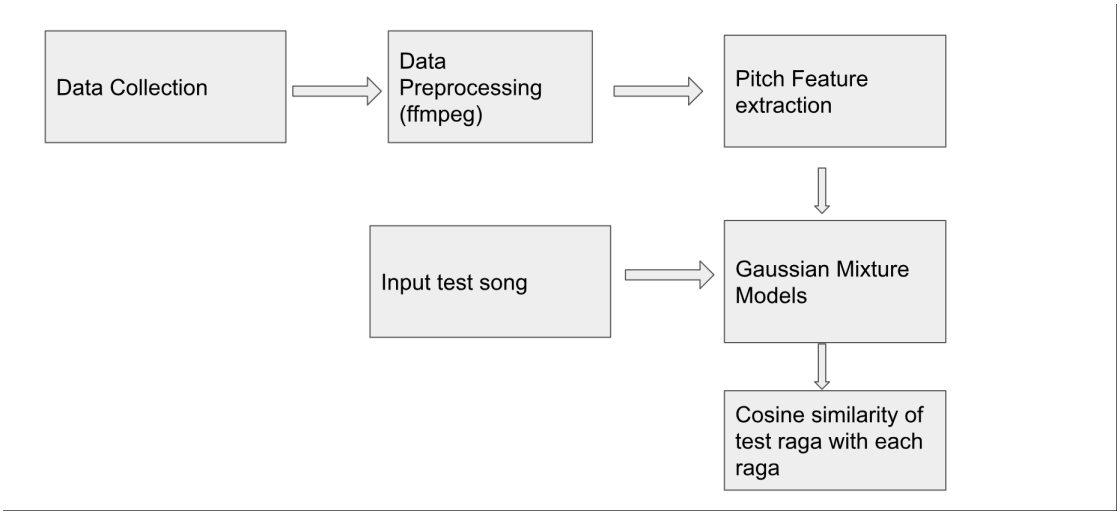

Figure 2: Workflow

0.3.1 Tools and technologies

1. Data preprocessing: ffmpeg software is used for media processing - converting songs into required format (.wav) and splitting them into training and testing. glob library in Python is used to read multiple wav files from a directory.

2. Pitch feature extraction: librosa library in Python is used for autocorrelation.

3. Gaussian Mixture Models: sklearn library in Python is used for GMMs.

4. Cosine similarity: math and numpy libraries in Python are used to calculate cosine similarities.

0.3.2 Data collection

We have considered only four ragas - Hamsadhvani, Hindolam, Mohana, Mayamalavagowla. Collected four different songs (one for each raga). All songs are in same scale. All are vocal songs and are monophonic.

0.3.3 Data preprocessing

Each song is split into training and testing. The song duration used for training is three minutes. For testing we used three snippets of each raga of twenty seconds duration each.

0.3.4 Autocorrelation of pitch

The autocorrelation of a signal describes the similarity of a signal against a time-shifted version of itself. For a signal x, the autocorrelation r is given by the below equation.[1]

$$r(k) = \sum_{n} x(n)x(n-k)$$

The autocorrelation is useful for finding repeated patterns in a signal. For example, at short lags, the autocorrelation can tell us something about the signal's fundamental frequency i.e pitch.

0.3.5 GMMs

A Gaussian mixture model is a probabilistic model that assumes all the data points generated from a mixture of finite number of Gaussian distributions with unknown parameters.The GaussianMixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models.sklearn.mixture is used for learning Gaussian Mixture Models. The parameters specified are the signal and number of gaussians i.e 36.[2] Each scale has 12 semitones which correspond to one gaussian. We considered all songs in one scale only. But in the worst case we can have notes in the scale above it and below it also. So we considered three scales with 12 semitones each which will give 36 gaussians.

0.4 Results

The following are the results obtained after cosine similarity test for each raga with every raga. Three test cases for each of them is taken and similarity score is calculated. It is observed that all ragas are closer to Janaka ragas (parent ragas) than themselves. Then they are next closer to Mohana and then Hamsadhvani. In a way, similarity between each raga is obtained.

Table 1: Results for Hamsadhvani

Test case	Hamsadhvani	Hindolam	Mohana	Mayamalavagowla
1	0.951	0.946	0.980	0.987
2	0.881	0.909	0.905	0.912
3	0.943	0.939	0.970	0.961

It is observed that Hamsadhvani is the closest to Mayamalavagowla. It is next closer to Mohana and Hamsadhvani. This is because there is only one swara (note) difference between Mohana and Hamsadhvani.

Table 2: Results for Hindolam

Test case	Hamsadhvani	Hindolam	Mohana	Mayamalavagowla
1	0.965	0.953	0.991	0.992
2	0.957	0.954	0.994	0.993
3	0.949	0.928	0.972	0.974

It is observed that Hindolam is the closest to Mayamalavagowla. It is next closer to Mohana and Hamsadhvani. It is least close to itself.

Table 3: Results for Mohana

Test case	Hamsadhvani	Hindolam	Mohana	Mayamalavagowla
1	0.950	0.947	0.988	0.988
2	0.955	0.938	0.980	0.986
3	0.950	0.948	0.981	0.989

It is observed that Mohana is the closest to Mayamalavagowla. Next closer to itself.

Table 4: Results for Mayamalavagowla

Test case	Hamsadhvani	Hindolam	Mohana	Mayamalavagowla
1	0.961	0.954	0.990	0.993
2	0.956	0.955	0.990	0.993
3	0.849	0.844	0.889	0.901

Mayamalavagowla is the closest to itself.

0.5 Problems faced

Most of the challenges faced are in data collection and feature extraction part.

1. Had to collect only monophonic songs at the end. This is because if polyphonic music is taken all the sources need to be separated. Moreover each source has different type of pitch and MFCC features.

2. Had to collect monophonic songs only in one scale (for convenience).

Taking songs in different scales would require normalization of all pitch values of all songs to one scale or would require to construct more number of Gaussians.

3. MFCC features were not found to be useful because they remain constant for entire duration of the song and vary from singer to singer. So this was not found to be useful to tackle this kind of classification problem which involves patterns rather than the sound itself.

4. It was also found that pitch extraction is also different for vocal music and is not same as that of instrumental music and so cannot be generalized. Initially we used HPS algorithm which was appropriate for instrumental music. Later we had to shift to autocorrelation method to extract pitch features from vocal music.

0.6 Future scope

This problem has many challenges and various sub problems involved. We have taken simplest type of music and tried Raga recognition. The future scope could be:

1. Extending raga recognition to polyphonic music. Different songs with both vocal and instrumental can be taken. This will again require different type of feature extraction and normalization techniques.

2. Extending for different scale of pitches. This would require pitch normalization techniques across all songs for all ragas to bring them to one scale.

3. Extending to different musicians. This would involve combining different feature extraction techniques. Extracting only pitch features would not be sufficient. MFCC extraction would be needed for different singers. Pitch features would be needed for finding out patterns. So we need to combine both these features.

4. Extending to more number of ragas. By extending to more number of ragas we can also find similarities between various Janya ragas or various Janaka ragas with themselves and also different Janya ragas with different Janaka ragas.

5. Extending to various other Machine Learning techniques. Other ML techniques could be Aarohan-Avarohana matching or pakkad matching, N-gram matrix distribution technique, etc.

6. Extending to various Deep Learning techniques like RNNs which are useful for classifying sequences or patterns.

0.7 Conclusion

It is observed that pitch features give better results than MFCC for this monophonic data with one singer. Among all the various Machine Learning and Deep Learning techniques, GMMs gave us better results, similarity between various ragas that we used. It was also observed that Janaka ragas have the best cosine similarity values. Janya ragas are closer to Janaka ragas.

0.8 Appendix

1. MFCC:
   The most commonly used feature extraction method in automatic speech recognition is Mel-Frequency Cepstral Coefficients (MFCC). MFCC mimics the logarithmic perception of loudness and pitch of human auditory system and tries to eliminate speaker dependent characteristics by excluding the fundamental frequency and their harmonics. The steps involved in MFCC are:[3]

   (a) Taking the Discrete Fourier Transform of the input signal.

   (b) The second processing step is computation of the mel-frequency spectrum.

   (c) The third processing step computes the logarithm of the signal.

   (d) In this step the cepstrum of the signal is computed. Cepstrum can be interpreted as the spectrum of a spectrum. The inverse transformation of the lower cepstral coefficients show the frequency response of the vocal tract and the inverse transformation of the higher order cepstral coefficients show the frequency spectrum of the source signal.

After performing these four steps 13 dimensional feature vector is obtained.

2. HPS algorithm:
   For musical signals the spectrum consists of a series of peaks corresponding to a fundamental frequency with harmonic components, called partials, positioned at integer multiples of the fundamental. Thus, by downsampling a spectrum several times the strongest harmonic peaks should line up. Therefore, if the spectra are multiplied per bin, each product will be small for all frequencies except at the position that corresponds to the fundamental frequency. This downsampling technique could therefore be used to estimate pitch in monophonic musical signals, and this is the key idea of HPS.

Bibliography

[1] Autocorrelation. (n.d.).
Retrieved from https://musicinformationretrieval.com/autocorrelation.html

[2] 2.1. Gaussian mixture models. (n.d.). Retrieved from http://scikit-learn.org/stable/modules/mixture.html

[3] Lutter, M. (2014, November 25). Mel-Frequency Cepstral Coefficients. Retrieved from http://recognize-speech.com/feature-extraction/mfcc

Acknowledgements

We thank Prof. Shrisha Rao for giving us an opportunity to do this project. We also thank Prof. V. Ramasubramanian for guiding us in the speech processing part of the project. We also thank Mrs. K. S. Padmaja who helped us in recording her voice in various ragas which we used as data for this project.