Tonic Independent Temporal Music Pattern Recognition in Indian Classical Music

Anonymous Author(s)
Affiliation
Address
email

Abstract

A vital aspect of Indian Classical music (**ICM**) is the *Raag*, which serves as a base on which improvisations and compositions (**IAC**) are created and presented. While every Raag represents a unique emotion, it allows a musician to explore and convey his interpretation of the lyrical content of a song. Although many works have explored the problem of classification of Raag, they have several short comings owing to the fact that they assume a prior knowledge of the *tonic* of the audio or is dependent on a preprocessing technique that identifies the tonic. In this work we introduce **1)** a novel data augmentation technique leveraging an inherent aspect of ICM that the semantics of IAC are only dependent on the relative position of notes with respect to the tonic and not the tonic itself **2)** Convolutional Neural Network based approach to build a robust model that can classify Raag independent of the tonic.

1 Introduction

ICM is an advanced and complex form of classical music. Carnatic and Hindustani classical music, which are its two primary branches, have evolved a great deal over a period of more than 500 years. A critical component of any Indian Classical Music Concert is the "Manodharma" or the "Spontaneous improvisation" which is an extremely complex task. The difficulty arises partly due to it being extemporaneous in nature, but more importantly because the musician has to balance the melodic and the entertainment aspects equally while being creative at the same time.

A *Raag* can be defined as a pattern of notes having characteristic embellishments, rhythm and intervals. The notion of Raag has similarities to the concept of *scale* in Western Classical Music in that it defines the note progressions allowed during ascent and descent of notes. Every note used in a given raag is relative to the base note called the *Shadaj* or the *tonic*. Gamaka and Sangathi are features that are unique to a Raag and this is what differentiates a Raag and a scale. Gamaka is a complex version of glissando that enables a musician to express the same progression of notes in multiple ways, due to which two Raags that have a similar set of notes may sound completely different. A sangathi can be defined as a short progression of notes with or without the using gamaka. If we assume a Raag to be the depiction of a musician's emotions as the theme of a painting, then a Sangathi can be assumed to the brush strokes used to create the Raag, while the painting itself is the composition or improvisation. The Gamaka and Sangathis together define the grammar for a Raag. Since Gamaka and Sangathi are the building blocks of a Raag, the key to identify a Raag lies in identifying the characteristic Gamaka and Sangathi of a Raag.

In short, if we imagine Raag to be some kind of distribution, small sequences(or subsequences) are sampled from the same repeatedly and organized in some particular order which would make sense aesthetically and musically and hence results either in a composition or an improvisation. Since every subsequence is nothing but temporal data, a musician is essentially creating a larger temporal sequence that is comprised of the smaller temporal subsequences. To add to this we have another dimension, the tonic. Hence we could treat the problem at hand as a kind of a spatio-temporal sequence, where the spatial does not exactly refer to Cartesian co-ordinates, but it refers to an extra dimension created due to the tonic.

As mentioned before, every note in a Raag is defined relative to the tonic. Hence identifying the tonic becomes very crucial for the task of Raag classification. From an ICM standpoint, the knowledge of the Raag is necessary to identify the tonic and vice versa. Furthermore, the tonic is decided based on the performers preference. This hence results in the selection of a non standard note as a tonic, for instance a note that is halfway between the notes D#4 and E4. As a result, building a model that is capable of identifying the Raag, irrespective of the value of the tonic is very challenging. Also, since every performer has their own way of expressing the same Gamaka, at the same or different speeds, often combining multiple ones to form complex patterns, adds to the complexity of this task. In this work we present a novel approach using CNN to build a robust model that is capable of identifying the Raag independent of the tonic. To the best of our knowledge, we are the first to use a CNN based approach to solve this problem

2 Related Work

Raag identity and characteristics is well studied in Indian Music Theory. There is vast literature among the music community as to what features are key in identifying Raags. In the Carnatic Music System the Parent Raag scheme (known as Janaka Raag scheme in ICM) defines set of 72 Raags from which all the other Raags maybe derived. A detailed analysis on the theoretical aspects of a Raga is presented in [7]. Another interesting work that describes the characteristics and features of a Raag which make them similar/dissimilar is presented in [2].

[4] describe the recognition of Ragas using pitch class and pitch class dyad distributions. Although they were able to achieve 75 percent accuracy on an unseen dataset, the model is not fully robust as it assumes prior knowledge of the tonic. [8] use a non-linear svm based approach where the similarities in the audio samples is represented using a combination of two kernels. [9] use a combination of hidden Markov Models, string matching algorithm and automatic note transcription is their model. The authors assume that the audio is monophonic. They also make an assumption on the the tonic of all the audio samples being G. [10] recover characteristic features of the Raga and feed it to a Neural Network to perform the classification. We note that the features used (arohana and avarohana and set of notes) here, although essential, are not good enough to build a powerful classifier. They also assume the prior knowledge of the tonic of the audio.

3 Approach

3.1 Dataset and Pre processing

3.1.1 Dataset

The first step in this direction is to prepare a dataset (available for viewing here) with sufficient number of recordings containing rich musical content to enable the CNN to learn the subtleties and the nuances of the music being fed. Although there are thousands of recordings available, we observe that most of these recordings are of substandard quality for the purposes of training a NN. Hence we hand pick recordings that are of good quality both in terms of recording standards and music. The created dataset 'DBICM2' has a total length of 2+ hours, featuring 8 artists and the recordings have 10 different tonics. We create 2 sub sets from the dataset, details of which have been outlined below :

- D1 - This contains 9 recordings in 3 Raags and 2 different tonics. All recordings in this dataset was performed by the same artist. It was created as a baseline dataset to compare our model's performance with its performance on real world data.
- D2 - This contains 21 recordings in 7 Raags and 8 different tonics. All the recordings are real world examples, ie, they have been sampled from Live Recordings of 5 different artists.

Efforts have been made to ensure no two recordings of the same Raag have the same tonic.

3.1.2 Pitch Tracking

As mentioned earlier, the recordings have been hand picked such that they are of good audio quality and hence we do not require any additional efforts to improve the quality of the audio. Since the model analyzes the melody component of the audio to identify the Raag, a critical step in preprocessing is to

perform pitch tracking of the audio and hence represent the given audio as an array of frequencies. We use Praat [3] (which is an open source software for the analysis of speech and sound) and Parselmouth [6] (which is a Python API for Praat), to perform the pitch tracking of the audio.

3.1.3 Frequency to MIDI conversion

Since Indian Classical Music predominantly characterized by acoustic instruments/ vocals, the audio hence is a continuous waveform. To be able to effectively analyze a sequence of frequencies, the audio has to be discretized. For this we could convert a given frequency into the corresponding Musical Instrument Digital Interface (MIDI) Note by using the formula, MIDI Note $= 69 + 12 * log_2(\frac{f}{440})$. The issue with this approach is that frequencies in the range of 21 Hz to 4186 Hz is represented by 88 discreet levels (i.e MIDI note 21 to 108) which leads to a severe loss of information. Hence we define 10 additional levels (which are technically called cents) between two MIDI notes, thus resulting in a total of $88 * 10$ possible levels. Hence every note is now represented as a tuple (M,C) which can be read off as the MIDI note 'M' and 'C' cents above 'M'.

3.1.4 Data Augmentation

Although the process mentioned above helps our classifier in efficiently learning the nuances of music, it necessitates that we feed the classifier with data in every possible tonic in order to make the classifier independent of the tonic. To address this problem, we propose a novel data augmentation technique. For this we first obtain all possible (M,C) tuples and transform them using M*10 + C. Following this we:

- Once the audio has been converted to a sequence of (M,C) tuples, transform the same using the formula : M*10 + C. Hence every Note is represented as a single value and not a tuple.
- The maximum and minimum value in the sequence will be ≤ 109 and ≥ 21 respectively.
- Depending on the maximum value and the minimum value in the sequence, we increment (and decrement) the values in the sequence n (and m) times by a value *delta* until the maximum (and minimum) value becomes equal to 21(109)
- At every step we save the resulting sequence. This results in the creation of multiple copies of the audio across different tonics.
- The audio clips from the previous step is then split into smaller clips as mentioned in 3.1.5

When we train the model on this augmented dataset, it now has a clear idea as to how a Raag looks like in different tonics and hence becomes independent of the tonic of the audio.

3.1.5 Sub-sequencing

In this work, we try to emulate the way in which a human listener tries to discern the components of the Raag and hence identifies the Raag being presented. We randomly select a starting point in the pitch tracked audio and extracting a 500 long array (which corresponds to nearly 5 seconds of audio. We choose 5 seconds as each Sangathi is this long on an average) and use this as a one sample. The Raag of the recording will be the target for the classification. In a 3 minute long clip (All of the training samples are 3-4 minutes long), we repeatedly sample 250 times. Sampling a 5 second audio 250 times in a 4 minutes long clip will create a lot of overlaps between the samples. We observe that this helps the model to better understand the data presented to it.

3.2 Network Architecture

Recently, CNNs have had a lot of success in NLP and speech recognition applications. [1] presents a concise explanation of how a CNN can be used for the task of speech recognition. Authors in [5] have used a novel VDCNN architecture based on deep CNNs to achieve improvements over-state-of-the-art on many datasets. Identifying Raags is essentially a sequence classification, with invariance to translation being a critical factor since a gamaka or sangathi can appear in any part of the audio, making it a suitable workload for a CNN. The first layer is an embedding layer with an embedding vector length of 80. The embedding layer is succeeded by a 1 dimensional convolution layer (with 30 filters, kernel size of 50, ReLU activation and Dropout), a maxpooling layer (kernel size 3), another 1 dimensional convolution layer (with 35 filters, kernel size of 100, ReLU activation and Dropout).

Dataset	Number of Raags represented	Number of tonics represented	Number of Test Instances Created	Test Accuracy
D1	3	2	4500	85.3%
D2	7	8	15000	75.7%

Table 1: Model Evaluation

This is then connected to a fully connected layer of size 50 followed by another dense layer with size equal to the number of classes (Raags) in the dataset followed by softmax activation. The network is optimized on categorical cross entropy loss using adam optimizer.

4 Model Evaluation

The steps involved in preparing the test data is similar to that of training data, only difference being, it does not require any data augmentation. We first transform the audio to obtain a sequence of (M,C) tuples. Following which we create multiple sub sequences of the same which acts as test samples. We test the model on both datasets 3.1.1. Note that there is equal representation(same number of subsequence samples) of all the Raags in the test set, hence it is a balanced classification problem.

4.1 Model Performance on D1

We first test the model on dataset D1, as it represents a simplified version of the data that model could expect to see in a real world scenario and obtain an test accuracy of 85.3% as shown in Table 1(1). This shows that the model is able to identify a good portion of the test samples correctly. We expect to see some error in prediction due to the fact that it might so happen that in the test samples (which are essentially subsequences of the original audio 3.1.5) there is only one note present for the entire length of the sample. This does cause some issue as there is not enough information for the model identify which Raag the sample belongs to. We call this issue as the prolonged note phenomenon(PNP).

4.2 Model Performance on D2

D2 contains 7 Raags with 8 tonics and the recordings have been sampled from live recordings of 5 different artists along with accompaniments. Since this is a real world example, performing the classification is extremely difficult due to the presence of 1) accompaniment as a result of which many portions of the recordings will be unclear to the model 2)PNP, which occurs more frequently in real world examples. It is important to note that the model was trained on recordings with 5 different tonics and the test set had 3 different tonics (totaling to 8 as in 3.1.1). Although the possible number of tonics in the test set is only 3, the model was unaware of the same and had to generalize over a large number of possibilities. As summarized in Table 1(1) we see that the model achieves a test accuracy of 75.7%.

5 Conclusion and Future Work

The data augmentation technique and the approach to solving this have been effective in identifying the Raag irrespective of the tonic. We observe that even though the accuracy of the model is around 75.7 % on a subsequence level, the model is able to correctly identify the Raag of an entire audio with much higher accuracy. This is because we use the predictions obtained on all the subsequences to infer the Raag of a recording, and hence even if the predictions on a few of the subsequences is wrong due to PNP or other issues, the model predicts 70-80 % of the subsequences correctly. We feel that this approach has tremendous potential and hence we are making efforts to create a larger and comprehensive dataset which will allow us to test the model in various other conditions.

References

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio,*

speech, and language processing, 22(10):1533–1545, 2014.

[2] VN Bhatkhande. Hindustani sangeet paddhati: Kramik pustak maalika vol. i-vi. *Sangeet Karyalaya*, 72, 1990.

[3] Paul Boersma et al. Praat, a system for doing phonetics by computer. *Glot international*, 5, 2002.

[4] Parag Chordia and Alex Rae. Raag recognition using pitch-class and pitch-class dyad distributions. In *ISMIR*, pages 431–436. Citeseer, 2007.

[5] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[6] Yannick Jadoul, Bill Thompson, and Bart De Boer. Introducing parselmouth: a python interface to praat. *Journal of Phonetics*, 71:1–15, 2018.

[7] TM Krishna and Vignesh Ishwar. Carnatic music: Svara, gamaka, motif and raga identity. In *Serra X, Rao P, Murthy H, Bozkurt B, editors. Proceedings of the 2nd CompMusic Workshop; 2012 Jul 12-13; Istanbul, Turkey. Barcelona: Universitat Pompeu Fabra; 2012.* Universitat Pompeu Fabra, 2012.

[8] Vijay Kumar, Harit Pandya, and CV Jawahar. Identifying ragas in indian music. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 767–772. IEEE, 2014.

[9] Gaurav Pandey, Chaitanya Mishra, and Paul Ipe. Tansen: A system for automatic raga identification. In *IICAI*, pages 1350–1363, 2003.

[10] Surendra Shetty and KK Achary. Raga mining of indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, 1(1):362, 2009.