Tonic Independent Raag Classification in Indian Classical Music

Sathwik Tejaswi Madhusudhan
Department of Industrial Engineering
University of Illinois Urbana Champaign
Champaign, IL 61820
stm4@illinois.edu

Girish Chowdhary
Department of Agriculture and Bioengineering
University of Illinois Urbana Champaign
Champaign, IL 61820
girishc@illinois.edu

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. 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 which has Carnatic Music (CM)[16] and Hindustani Music (HM)[3], as its two primary branches. *Raag* is a basic element for compositions and the improvisations (or "Manodharma" [12]) and is defined as a pattern of notes having characteristic embellishments, rhythm and intervals. Every Raag is associated with a unique emotion. [2] conduct a study on perception of Raags and observe that an inexperienced listener is able to identify the emotion of a Raag effectively by relying only on psychological cues. Hence Raag can be used in a variety of music related tasks like organizing recordings into songs with similar emotional content, music recommendation systems [15] etc.

Every note used in a Raag is defined with respect to a base note called the *Tonic*. Features of the Raag like the arohana-avarohana(legal ascent and descent note progressions respectively) and the Gamaka are key in identifying a Raag. The Gamaka, which is unique to every Raag, 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. [11] describe and illustrate 6 popular types of gamakas in CM. [11] and [10] provide a detailed explanation of Raag in CM and HM respectively.

Recent success of Convolutional Neural Networks(CNN) in text classification applications [7] [1] show that CNNs are capable of handling temporal sequences well. The Raag identification/Classification can be viewed as a sequence classification problem, with the set of all possible notes being the dictionary. As CNNs are translation invariant, they can identify features of a Raag irrespective of the order of their appearance in the audio. In this work we introduce a CNN based approach to Raag Classification in conjuction with of a novel data augmentation technique which makes our method Tonic independent.

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

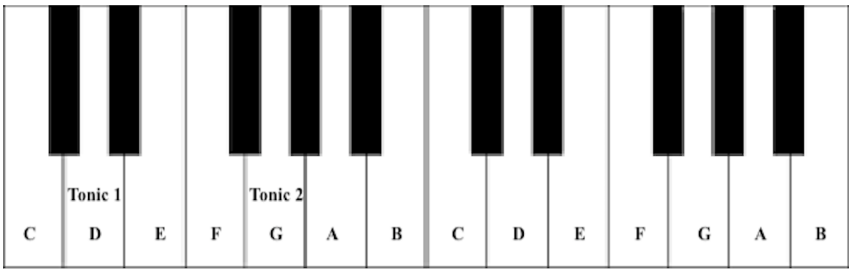

Figure 1: Above is an illustration that shows how one can arrive at two different Raags with same set of notes. If we assumed the note D to be a Tonic, then D-E-F-G-A-B-C-D becomes a raga called *Kharaharapriya* while if G is made a tonic, the sequence G-A-B-C-D-E-F-G becomes a raga called *Harikambodhi*. Combined with the Gamakas that are used both of these sound completely different.

Previous works have used machine learning based approaches to recognize Raags. [6] describe the recognition of Ragas using pitch class and pitch class dyad distributions. Their approach assumes a prior knowledge of the tonic of the audio(every recording is manually annotated with the tonic). [13] use a combination of hidden Markov Models, string matching algorithm and automatic note transcription as part of their approach. The authors assume that the audio is monophonic i.e, that the audio has only one performer(either vocals or some melody based instrument). But most of the audio recordings are not monophonic. They also make an assumption that the tonic of all the audio samples is G#. [14] recover the arohana(rule for note ascent) and avarohana(rule for note descent) and use it as a feature in the model. This is insufficient as multiple Raags have similar arohana and avarohana but the gamaka and the characteristic phrases make them sound entirely different [11]. [8] use Gaussian mixture model (GMM) based HMM using features like MFCC and Chroma.

2 Approach

2.1 Dataset and Pre processing

2.1.1 Dataset

We create a dataset (available for viewing here) with sufficient number of recordings containing rich musical content to enable the model to learn the subtleties and the nuances of the music being fed. The created dataset 'DBICM' features recordings of 8 artists and the recordings have 10 different tonics. We then use the Data Augmentation technique and the subsequencing mentioned in 2.1.4 and 2.1.5 respectively, resulting in close 300000 10 second long clips. We have ensured that none of the selected recordings of the same Raag have the same tonic. This allows us to test our model and validate if it can perform classification of Raag independent of the tonic. We create 2 sub sets from the dataset, details of which have been outlined below:

- D1 - This contains 9 recordings (6 for training and 3 testing) 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 (14 training and 7 testing) in 7 Raags and 8 different tonics. All the recordings are real world examples, i.e., they have been sampled from Live Recordings of 5 different artists.

2.1.2 Pitch Tracking

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 [5][4] (which is an open source software for the analysis of speech and sound) and Parselmouth [9] (which is a Python API for Praat), to perform the pitch tracking of the audio.

2.1.3 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 α additional levels (which are technically called cents) between two MIDI notes, thus resulting in a total of $88 * \alpha$ possible levels (we choose α to be 10). 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'.

2.1.4 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, (which may or may not be available), in order to make the classifier independent of the tonic. To address this problem, we propose a novel data augmentation technique, summarized as below: Following this we:

- Perform pitch tracking on the audio to obtain a sequence of frequencies *S* = $S_1, S_2, ..S_T$, where 'T' is the total number of time steps of the audio.
- Convert S into a sequence of (M, C) tuples as mentioned in 2.1.3 to obtain $MC = \{(M, C)_1, (M, C)_2, ..(M, C)_T\} = \{MC_1, MC_2, ...MC_T\}$
- Transform $(M, C)_i$ as $M + C$ i.e if $M = 29$ and $C = 0.1$ $M + C = 29.1$.
- Let *minv* and *maxv* be the minimum and the maximum value as observed in the above sequence.
- Select a suitable step size Δ, say 0.5, and do $MC = MC - \Delta$ until minv of $MC = 8$. (Similarly do $MC = MC + \Delta$ until maxv of $MC = 108$).
- For every decrement/increment performed above, we get a new sequence that represents how the audio would look like in another tonic. Each of these new sequences are stored.

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.

2.1.5 Sub-sequencing

We observe that a human while trying to identify a Raag, breaks it up into smaller subsequences as they listen to the audio and tries to classify each of these into a particular Raag and hence identifies the Raag of the audio presented. We emulate this by repeatedly sampling subsequences $MC_{sub} = \{MC_i, MC_{i+1}, ...MC_{i+1000}\}$ and training the model on the these subsequences. Note that the sampling of these subsequences is random in that 'i' is selected randomly.

2.2 Model Selection

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 [7] 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 features of a Raag can appear in any part of the audio, making it a suitable workload for a CNN. The Network Architecture has been outlined in table 1.

3 Results

3.1 Training

The model was trained using the mini batch stochastic gradient descent algorithm and adam optimizer. Owing to the small dataset size, overfitting was one of the main concerns. Multiple approaches were used to mitigate the effects of overfiting like using dropout, early stopping, batch normalization

Layer Name	Layer_Specifications
Feature Embedding	Embedding Layer (Embedding Size = 80)
Sequential1	1D Conv (kernel = 20, feature maps = 16, Relu activation), BatchNorm 1D, Dropout
Sequential2	1D Conv (kernel = 15, feature maps = 32, Relu activation), BatchNorm 1D, Dropout, MaxPool
Sequential3	1D Conv (kernel = 10, feature maps = 64, Relu activation), BatchNorm 1D, Dropout
Sequential4	1D Conv (kernel = 5, feature maps = 128, Relu activation), BatchNorm 1D, Dropout, MaxPool
Sequential5	1D Conv (kernel = 5, feature maps = 256, Relu, activation), BatchNorm 1D, Dropout
Dense	Fully Connected Layer and Softmax output

Table 1: Network Architecture

etc. A significantly high dropout rates had to be employed to help the model generalize to the test dataset. Parameters like Δ (refer 2.1.4), α (refer 2.1.3) and the subsequence length are very critical for training. As the value of Δ reduces (approach 0), the model tends to overfit and as Δ increases (approac 1) the model tends under fit, which can be rectified with more number of training epochs. The model severely underfits once the value of Δ exceeds 1.

Dataset	Number of Raags represented	Number of tonics represented	Number of Test Instances Created	Test Accuracy
D1	3	2	30000	77.1%
D2	7	8	70000	72.8%

Table 2: Model Evaluation

3.2 Model Evaluation

As a result of the subsequencing and data augmentation, the data produced has equal number of samples for each class and hence is a balanced classification problem. We thus choose accuracy as a metric to evaluate the performance of the model. Dataset D1 has audio clips representing 3 Raags in 2 different tonics. We create 70000 training instances and 30000 test instance by using the data augmentation and subsequencing techniques. The model is able to achieve a test accuracy of 77.1% on the test. Dataset D2 has audio clips that represent 7 Raags in 8 different tonics, from which 140000 training and 70000 testing instances were created. The model was able to achieve an accuracy of 72.8 % on the same.

3.3 Inference on Long Sequences and Sequences with multiple Raags

In most practical situations we are required to make predictions on an audio clip as a whole. We observe that by employing subsequencing, we can obtain multiple samples from the audio on which we use the model to make predictions. The overall Raag content of the audio can be inferred by taking a vote from the predictions made on the samples. There are numerous audio recordings where the musician uses multiple Raags sequentially. By replacing the random sampling as described in 2.1.5 with a sequential sampling,(i.e vary i from 0 to T-1000, with a suitable step size), we can obtain a series of predictions which can thus be used to describe how the Raag content of the audio changes wih time.

4 Conclusion and Future Work

We have presented a new data augmentation technique that enables a model to learn the semantics of different Raags in numerous different tonics apart from the tonic present in the audio. The model, which is a simple 5 layer deep Convolutional Neural Network, achieves an accuracy of around 72-77% on the test datasets. We believe that this approach towards Raag Recognition has tremendous potential. It will be interesting to see how a larger multi-scale model (similar to [17]) trained by employing our data augmentation technique performs with datasets with an increased number of Raags and tonics.

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] Laura-Lee Balkwill and William Forde Thompson. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music perception: an interdisciplinary journal*, 17(1):43–64, 1999.

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

[4] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the institute of phonetic sciences*, volume 17, pages 97–110. Amsterdam, 1993.

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

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

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

[8] Pranay Dighe, Parul Agrawal, Harish Karnick, Siddartha Thota, and Bhiksha Raj. Scale independent raga identification using chromagram patterns and swara based features. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–4. IEEE, 2013.

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

[10] Nazir Ali Jairazbhoy. *The rāgs of North Indian music: their structure and evolution*. Popular Prakashan, 1995.

[11] 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.

[12] Laudan Nooshin and Richard Widdess. Improvisation in iranian and indian music. *Journal of the Indian Musicological Society*, 36:104–119, 2006.

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

[14] 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.

[15] Yading Song, Simon Dixon, and Marcus Pearce. A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*, volume 4, 2012.

[16] Tanjore Viswanathan and Matthew Harp Allen. *Music in South India: the Karn† atak concert tradition and beyond: experiencing music, expressing culture*. Number Sirsi) i9780195145908. 2004.

[17] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.