MODELING CARNATIC RHYTHM GENERATION : A DATA DRIVEN APPROACH BASED ON RHYTHMIC ANALYSIS

Carlos Guedes
New York University Abu Dhabi
carlos.guedes@nyu.edu

Konstantinos Trochidis
New York University Abu Dhabi
kt70@nyu.edu

Akshay Anantapadmanabhan
Independent Musician
akshaylaya@gmail.com

ABSTRACT

In this paper, we present a data-driven approach for automatically generating South Indian rhythmic patterns. The method uses a corpus of Carnatic percussion compositions and grooves performed in adi tala – a uniform eight-beat cycle. To model the rhythmic structure and the generation process of phrasings that fit within the tala cycles, we use a set of arithmetic partitions that model the strategies used by professional Carnatic percussionists in their performance. Each partition consists of combinations of stroke sequences. This modeling approach has been validated in terms of the groupings used in this music idiom by direct discussions with Carnatic music experts. Two approaches were used for grouping the sequences of strokes into meaningful rhythmic patterns. The first is based on a well-formed dictionary of pre-recorded phrase variations of stroke groupings and the second one on a segmentation algorithm that works by comparing the distance of adjacent strokes. The sequences of strokes from both approaches were later analyzed and clustered by similarity. The results from these analyses are discussed and used to improve existing generative approaches for modelling this particular genre by emulating Carnatic-style percussive sequences and creating rhythmic grooves. The creation of these tools can be used by musicians and artists for creative purposes in their performance and also in music education as a means of actively enculturing lay people into this musical style.

1. INTRODUCTION

There is an increasing interest in developing computational strategies for the analysis and understanding of non-Eurogenetic music. Work by [1], [2], [3] and [4] in non-Eurogenetic music constitute some of the earlier examples in this area. The goal of this paper is to develop expert systems that can reliably generate music in this style of Indian Classical music, envisioning a contribution on two levels: 1) the creation of tools for lay audiences to interact with musical styles beyond the Western ones; and 2) the automatic generation of unlimited amounts of data for training machine learning algorithms. By building applications that can recreate these musical styles we hope to create innovative tools for interaction with musical heritage that go beyond passively listening to the music. Earlier work [5] used an n-gram approach for modeling Carnatic percussion generation. N-gram transition probabilities up to a five-gram were estimated by counting the frequency of the strokes in the training corpus. The size of n-grams was set to up to a five-gram to test how past information and size of accumulated memory could affect and change the generation process. The generation process used these data to generate new stroke events sequentially. The main drawback of this method was that it failed to successfully capture the long-term structure and grammar of this particular idiom and being only successful in capturing local and short term phrasing. In our present work, we aim to overcome these issues by introducing a new data-driven approach of modeling the tala cycle based on a set of arithmetic partitions which capture reliably the rhythmic structure of the tala. We also implement two novel grouping methods for stroke segmentation into syntactic valid phrases using a grouping algorithm that works by comparing the distances of adjacent strokes and a dictionary of pre-recorded phrase variations and stroke groupings of phrases for this particular idiom. Based on this analysis, we developed an application that improves the generation of South Indian rhythms and enhances the interaction of the user by adopting data visualization techniques during the generation. The paper is organized as follows: this section presents an introduction and background research on music generative applications and work on modeling Carnatic rhythm generation while section 2 presents background information on the rhythmic structure in Carnatic music. Section 3 describes the proposed approach, dataset and methods while section 4 discusses the rhythmic grouping algorithm and the dictionary of pre-recorded phrases. Section 5 describes the clustering analysis of the data. The generation process and application are provided in section 6. Finally, discussion and future work are drawn in section 7.

2. RHYTHMIC STRUCTURE IN CARNATIC MUSIC

The rhythmic framework of Carnatic music is based on the tala, which provides a cyclic framework for improvisation through strategies like repetition and grouping. The tala consists of a fixed time length cycle called avartana, which can also be called the tala cycle. The avartana is divided into equidistant basic time units called aksaras, and the first aksara of each avartana is called the sama [6]. Two primary percussion accompaniments in Carnatic music are the Mridangam and Kanjira, where the Mridangam is known for its lead percussion role. The rhythmic complexities of Carnatic music are especially showcased during the percussion solo or taniavartanam, which is what we focus on in our work. In a concert, each instrument performs separately and then they trade off in shorter cycles with a precise question-answer like session, followed by a joint climactic ending. There are many forms in solo improvisation, but in the context of these experiments, three main types of characterizations were performed:

1. sarvalaghu patterns - short groove phrases that are repetitive in nature over any tala cycle (usually less than half a cycle in length)

2. korvai – multi-part (minimum two) compositions that can last over multiple cycles which are repeated three times. Each part generally adheres to the rules of arithmetic progressions.

3. pre-korvai improvisation - assuming korvais contain multiple idea phrases, these sections improvise around those particular phrases that are derived from the korvai and finally culminate into the climactic conclusion (the korvai).

All training excerpts used in the proposed generation method were performed by the Mridangam and Kanjira-drum in the context of separate solo improvisations.

3. APPROACH

3.1 Overall process

The approach we adopt in this study is given a dataset of audio recordings, initially we obtain an automatic transcription of a sequence of time-aligned events of all stroke types, their durations (IOIs) and velocities. Next, we obtain a series of groupings of strokes extracted using a segmentation algorithm that compares the IOI distances between adjacent strokes. We also use another pre-recorded dataset of groupings which include well-formed phrase variations of rhythmic patterns and were composed by percussion professionals of this genre. Next, all the patterns are indexed in terms of their duration and are kept for further analysis. We transform all textual representation of the groupings into feature vector representations by using the bags of words approach and all grouping patterns are clustered based on similarity using the k-means algorithm. To generate and synthesize the talas we model the aditala cycles as a series of arithmetic partitions that consist of combinations of groupings. Finally, an algorithm implemented in the Max programming environment is used to generate the talas by choosing different rhythmic groupings from the different clusters and concatenate them to form the partition. The block diagram in Figure 1 shows the overall approach.

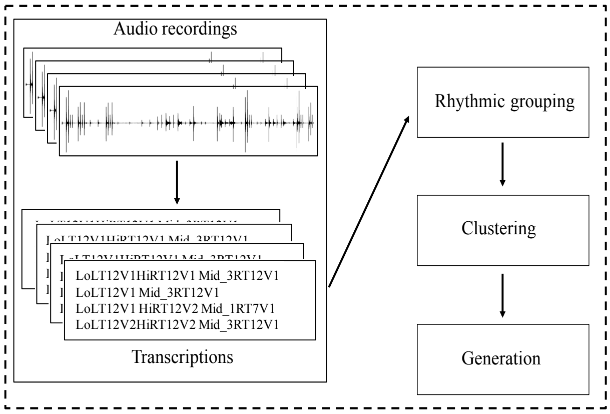

Figure 1. Block diagram of the approach

3.2 Dataset

The corpus consisted of 23 percussion solo compositions and groove patterns in aditala (8 beat-cycle) in three different tempo levels: slow (70bpm), moderate (85bpm) and fast (105bpm). The recordings were performed by a professional Carnatic percussionist with the Mridangam and Kanjira drum. All excerpts were recorded using a metronome and were manually annotated by an expert using Sonic Visualiser [7].

3.3 Encoding the strokes

For the Mridangam dataset each stroke event was coded as a string based on five registers (Lo/Mid1-2-3/Hi), the hand (left (L) or right (R)) used for initiating the stroke, the inter-onset interval (IOI) between strokes and a value indicating the velocity of the stroke. For the Kanjira drum data we used three register values (Lo/Mid/Hi), the inter-onset interval (IOI) between strokes and the velocity of the stroke. For the Mridangam strokes we allowed also the coding of two composite strokes played simultaneously with left and right hands. Although the Mridangam and Kanjira have a richer variety of registers and strokes, the reduction to three registers for the Kanjira and five for the Mridangam was a step to compromise the different stroke definition. This reduction was validated by percussion experts as a process to accurately encode the different strokes in both percussion instruments. The normalized velocity values of the strokes were obtained by computing an onset detection function by combining energy and phase information in the complex frequency domain [8], and estimating its amplitude level with a value between 0.2 and 1 according to the strength of the stroke.
Each stroke was encoded as a string containing the register, the hand which the stroke was played, velocity, and duration (IOI). For example, LoLV2T4, indicates a stroke in the Low register using the left hand on the Mridangam with velocity 0.5 and duration of a dotted quarter note. A detailed description of the encoding method is discussed in [5].

4. RHYTHMIC GROUPING

4.1 Rhythmic grouping based on pre-recorded phrases (solkattu)

The concept of groupings is a fundamental building-block of Carnatic music, be it melody or rhythm. In the vocal percussive form of Konakkol, the language to communicate these rhythms, groupings are also sometimes referred to as 'solkattus.' These solkattus at a very basic level are phrases whose sum of syllables map to integer numbers. For example, 'tha ki ta' maps to three, 'tha ka dhi mi' maps to four, 'tha ka tha ki ta' maps to five and so on [9]. The mapping however is not one-to-one as the musician is free to play many different n-syllable phrases to represent just one grouping. Assuming, you can fit four syllables per beat, in the eight beat cycle of adi tala, there is a possibility of thirty-two syllables to fill one cycle. One strategy to perform in this cycle, is to use a series of sollu groupings along with intermittent rests to fill the cycle. Multiple groupings can also be concatenated to form larger musically-relevant groupings. As the number of cycles grow, the possibilities of grouping combinations also increases. There are certain rules that are followed by percussionists that allow this rhythmic generation to be more musically aesthetic, rather than just a series of groupings. This approach of how groupings are grouped will be touched upon in section 6 where we discuss generation. In our study we used two approaches for grouping the sequences of strokes. The first which is described in this section is based on a well formed dictionary of pre-recorded phrase variations while the second one is based on a segmentation algorithm. Figure 2 depicts a grouping example with a duration of five pulses taken from the pre-recorded dictionary of groupings.

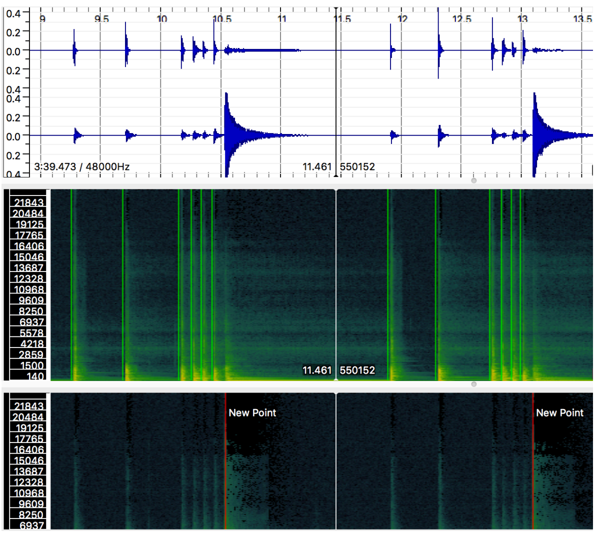

Figure 2. Example of a grouping with a duration of five pulses.

4.2 Nearest neighbor segmentation algorithm

The grouping algorithm that we used is a modified version of the mutual nearest neighbor algorithm [10]. It works based on the proximity of the strokes by measuring the distance between adjacent strokes. Strokes are grouped together if they are nearest neighbors to each other. A structural constraint of the algorithm is that every grouping has a minimum number of 2 strokes. This constraint was adopted to avoid very small groupings of individual strokes. The algorithm also works hierarchically by using a second layer to group smaller groupings into a larger one. The algorithm stops parsing and forms new groupings using a threshold that represents the largest duration that a grouping pattern can take in our case an 8 pulse duration. Figure 3 illustrates an example of grouping. The algorithm starts grouping the first two strokes in the rhythmic sequence based on the initial constraint and then compares the distances D1 and D2 between the last stroke in the grouping (blue dot) with the the next two strokes (green and red dots). D2 is less than D1 so the algorithm finds a boundary, saves the current grouping, and creates a new one. The algorithm is iterative and works hierarchically, e.g. when it finishes parsing the strokes and comparing the distances we get two layers of groupings four groupings for the first and two groupings for the second layer.

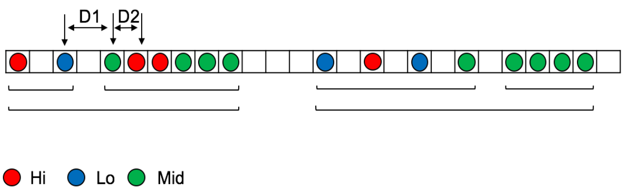

Figure 3. Rhythmic grouping of strokes based on the mutual nearest algorithm.

5. CLUSTERING

All the groupings were represented initially with symbolic notation representing the sequences of strokes. To transform the textual symbolic information to meaningful feature vectors we used the bags of words approach and extracted all bigrams of the groupings. We generated feature vectors of the groupings by counting the times each bigram occurs in a grouping. This leaded to a feature matrix, which could be further used for clustering analysis. Figure 4 illustrates the feature representation. Finally, the k-means clustering approach was used to cluster the groupings of strokes in terms of similarity.

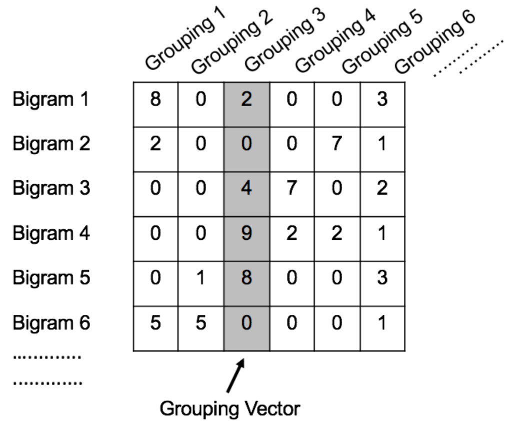

Figure 4. Feature representation of grouping patterns based on bigrams and bags of words approach.

5.1 Visualizing data using T-SNE

The next step was to convert the high-dimensional data set representing the center of the clusters from the clustering analysis into a matrix of pairwise similarities to enable the visualization of the resulting data. Traditional dimensionality reduction techniques such as Principal Components Analysis are linear techniques that focus on keeping the low-dimensional representations of dissimilar data points far apart. In our analysis, we used the so called t-distributed stochastic neighbor embedding (t-SNE), for visualizing the resulting similarity data [11]. Compared to methods discussed previously, t-SNE is capable of capturing much of the local structure of the high dimensional data, while also revealing global structure such as the presence of clusters at several scales. Figure 5 illustrates the result of the t-SNE transformation on the clusters of the groupings. A 2-dimensional axis comprising two components were used for the t-SNE analysis and the pairwise distances between the cluster centers of the groupings are plotted in the axis.

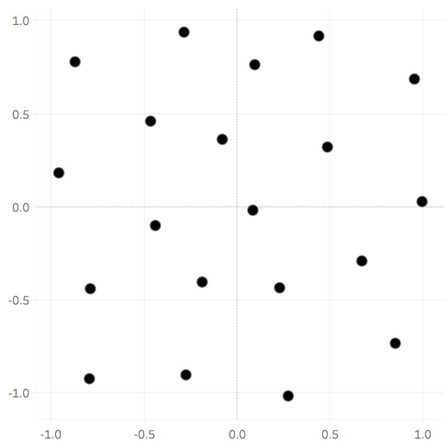

Figure 5 2D map of proximity and similarity distances between the pattern clusters of groupings

6. GENERATION PROCESS

To synthesize and generate the talas, we modelled the 8-beat aditala cycle into a series of partitions of 32 timepoints per cycle, assuming a beat subdivision in 4 parts. The templates of partitions that we adopted were validated and proposed in terms of the grammar and theory of this music idiom by direct discussions with Carnatic music expert musicians. Partitions are essentially strategies of grouping sollkattus in a musically aesthetic fashion to perform within tala cycle(s). Musically speaking, this approach is used frequently in the concept referred to as 'aradhis' or endings used in Carnatic music solos and accompaniment. An aradhi is a phrase that is repeated three times with some or no rests in between. As an example, an eight-beat cycle can be split into three sections of two beats with two-one beat rests in between. Or more specifically, three of the same eight syllable phrases (2 beats each) with two four-syllable rests separating the three groupings. The sum of these five parts will equal 32 syllables, or one-cycle of aditala. Figure 6 represents some of the partition templates that we used to generate groupings to fill one tala cycle.

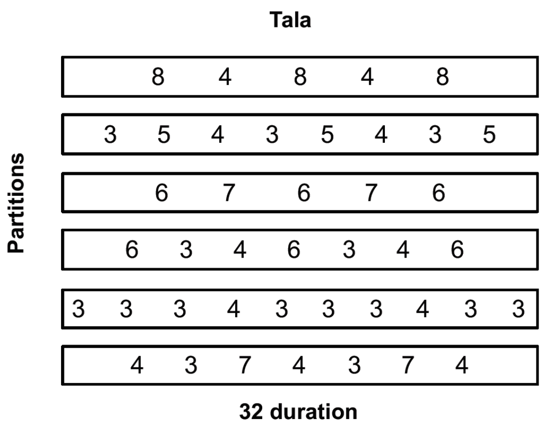

Figure 6. Partition templates of aditala cycles.

By analyzing, segmenting and clustering the different rhythmic groupings based on their duration and similarity we were able to have an index of groupings with different durations for each cluster. In order to generate talas of a certain arithmetic partition we used groupings from the same cluster and concatenate them to form the partition tala cycle. By having a large number of groupings in a cluster indexed by their durations we could generate rhythm in tala cycles with different variations of a specific partition using combinations of groupings.

6.1 Carnatic music generation

The results from the analysis were used to develop a generative model that creates rhythmic endings based on these data. The model was implemented as a Max patch that uses as inputs the arithmetic partitions, the coordinates of the clusters and the groupings and their durations in each cluster. This tool not only synthesizes the results from the analyses but it can be also used as a computational application for creative and learning exploration of these rhythms. This latter aspect is of particular interest as it provides the gateway to develop software applications for automatic rhythm generation in non-Eurogenetic music styles. Figure 7 depicts a screenshot of the Max patch. The user can interact with each cluster by travelling in the 2D space and generate talas of preference based on a set of template partitions in various tempo of choice. Clusters with enough groupings to generate the specific partition of talas are represented with black dots while clusters with insufficient number of groupings are grey. This offers a greater intuition and interactivity to the user. The user can track the position of the tala that is been playing by having a visual display of the playback of the tala inside the cycle. Users can increase the variation of the groupings, used to form a certain tala partition by increasing the circle of the cluster chosen. They can also remove clusters they do not like or can form new arithmetic partitions of groupings and generate new combinations to fill the tala cycle by adding a new partition strategy.

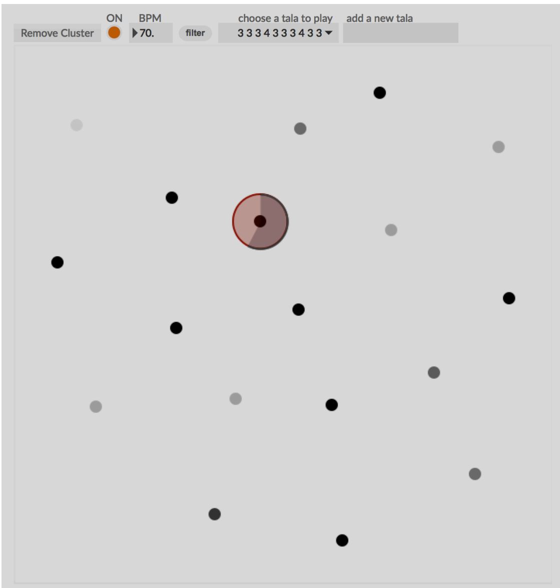

Figure 7. Max patch of Carnatic music generation application

7. DISCUSSION AND FUTURE WORK

This work presents a method for automatically generating new Carnatic style rhythmic patterns using a corpus of compositions and grooves from this genre. The approach we adopt in this study is to model the aditala cycle as a series of arithmetic partitions. Each partition is formed by concatenating different durations of rhythmic groupings and sequences of strokes. We used two approaches to group the sequences of strokes into rhythm patterns: the first uses a grouping algorithm that compares the distance of adjacent IOIs of strokes while the other uses a dictionary of pre-recorded grouping variations of solkattus composed and performed by an expert percussionist of this genre. We transformed all textual representation of the groupings into vector feature representations by using bags of words approach and all grouping patterns were clustered based on similarity using the k-means algorithm. Finally, a patch implemented in the Max environment was used to generate rhythm within tala cycles by choosing and concatenating different groupings to form particular partition durations. After discussions and initial evaluation of the system with Carnatic percussion professionals it can be concluded that the system can successfully generate Carnatic style rhythmic patterns based on the original data. A drawback of the system lies in the fact that sometimes even though the groupings/solkattus are musically meaningful, they are not always grammatically valid and context of using the groupings with rule-based constraints is yet to be explored. Grammatical invalidity is mainly caused due to the existing rhythm grouping algorithm that we use for grouping and the k-means clustering which classifies the groupings based on similarity. These techniques might not be able to capture entirely the strategies of Carnatic percussionists in terms of grouping and categorization of the strokes into phrases. To improve the current methodology and tackle this problem we aim to manually annotate all the different groupings of strokes and possible combinations with the help of expert percussionists of this genre in the entire corpus. We will also want to extend this work by discussing with experts on potential methodologies to manually cluster the groupings based on similarity as opposed to k-means clustering. This will allow us to have a more precise categorization of groupings which we can later use in our generations. Future work will evaluate the effectiveness of the method by conducting a large scale perceptual study using a group of professional Carnatic musicians comparing ratings between machine and human composed excerpts. We will also like to extent our work and use similar methodologies to generate other non-Eurogenetic rhythmic styles, with a particular emphasis on rhythms from the Gulf.

8. REFERENCES

[1] Serra, X.: A Multicultural Approach in Music Information Research. In Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR), Miami, pp. 151–156, 2011

[2] Srinivasamurthy, A., Holzapfel, A. & Serra, X.: In search of automatic rhythm analysis methods for turkish and indian art music. Journal of New Music Research, 43:94–114, 2013

[3] Bozkurt, B., Ayangil, R., & Holzapfel, A.: Computational analysis of Turkish makam music: Review of state of-the-art and challenges. Journal of New Music Research, 43(1), 3–23, 2014

[4] Herremans, D., Weisser, S., Sörensen, K., Conklin, D.: Generating structured music for bagana using

quality metrics based on Markov models. Expert Systems With Applications. 42(21):424–7435, 2015

[5] Trochidis, K., Guedes, C., Anantapadmanabhan, A. CAMeL: Carnatic Percussion Music Generation Using N-Gram Models. Proc. Int. Conf. Sound and Music Computing (SMC2016), Hamburg, Germany, (2013).

[6] Sambamoorthy, P.: South Indian Music Vol. I-VI, The Indian Music Publishing House, 1998.

[7] Cannam, C., Landone, C. and Sandler M., Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files, Proceedings of the Int. Conf. on Multimedia, pp. 1467-1468, ACM, 2010.

[8] Duxbury, C., Bello, J.P, Davies, M. and Sandler M., Complex domain onset detection for musical signals, Proceedings of the Digital Audio Effects Workshop (DAFx), No. 1, pp. 6-9, 2003

[9] Nelson, D.P: Solkattu Manual: An Introduction to the Rhythmic Language of South Indian Music. Wesleyan University Press, 2014.

[10] Tousaint, G. Measuring the Perceptual Similarity of Middle-Eastern Rhythms: A Cross-Cultural Empirical Study. In Proceedings of the Fourth International Conference on Analytical Approaches to World Music. New York, 2016.

[11] Van der Maaten, L.J.P., Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research, 9(Nov):2579-2605, 2008.