GAMAKA SYNTHESIS FOR KALPITHA SWARAS IN CARNATIC MUSIC

Raghavasimhan Sankaranarayanan
Georgia Institute of Technology
Center for Music Technology

Gil Weinberg
Georgia Institute of Technology
Center for Music Technology

ABSTRACT

Gamaka (note ornamentation) is an essential element of Carnatic music. Earlier works in computer-generated gamakas focused on developing mathematical models of each gamaka, which fails to capture the intricate changes in pitch and thus does not sound natural. To address this challenge, this work approaches the synthesis of gamaka for kalpitha swaras (composed notes) in Carnatic music using a data-driven system. The model uses masked latent space representation in an auto-encoder architecture with features extracted using convolutional layers. It takes as input, the pitch contour extracted from symbolic data to generate a pitch contour with gamaka information embedded in it. The model is successful in synthesizing gamaka with nuances that closely follow the ground truth.

1. INTRODUCTION AND RELATED WORK

Carnatic Music is a system commonly associated with the southern part of India. The music form uses micro-tonal variations, where swaras (musical notes) of phrases are performed with ornamentation called 'gamakas' [1]. In Carnatic music, the gamakas are integral parts of asserting a raga [2]. In this work, we concentrate on tempo-based Kalpitha Swaras (Composed Notes).

Automatically synthesizing gamakas given plain notes has various applications in fields such as Robotic Musicianship, Music Education, virtual instruments, and others.

The main contributions of this work are, a dataset of note annotated Varnams in different tempos detailed in section 2.1 and GamakaNet - A novel Masked Latent Space architecture to generate gamaka given kalpitha swaras (in MIDI [3]) in the equal temperament scale.

Ashtamoorthy et al. [4] concentrated on rule-based gamaka generation with flute synthesis which required the gamakas to be manually provided. The Gaayaka software [5] could synthesize automatic gamakas using a custom musical representation. However, the work is not scalable since each raga requires a definition file that explicitly provides a mathematical model of gamaka for each note transition. Sasindran et. al [6] use wavelets to model a subset of gamaka types. This approach also requires manual specification of gamakas. MellisAi [7] employs a data-driven approach, which features an LSTM-based architecture to model music generation while using mathematical models to generate the gamakas.

2. METHODOLOGY

2.1 Dataset

We recorded 5 concert instrumentalists with at least 15 years of experience in Carnatic music. The instruments include violin, keyboard, and veena - an Indian plucked lute. These instruments were chosen so as to make fundamental frequency (f_0) tracking and annotating easier. Varnams are compositions with a concrete structure without deviations between different teaching lineages. We recorded 2 varnams in raga Abhogi and Kalyani. These ragas have largely varied sets of notes. Each varnam was performed in 3 different tempos - 80, 120, and 160 beats per minute (BPM) to account for 30 recordings in total. All the samples are 16-bit recorded at 44.1KHz. The recordings are monophonic without a Tanpura [8] accompaniment to minimize errors in f_0 tracking. Each recording was manually annotated with the intended MIDI notes We used pYIN [9] and CREPE [10] for f_0 tracking. We plan to make this dataset available to the research community.

2.2 Data Pre-processing

The MIDI notes were converted into f_0 contour and normalized for octave and the tonic of the instrument and performance respectively. The resulting data is centered around 1.0 with 0.0 representing un-voiced regions. We performed zero-phase low pass filtering to remove high-frequency noise in f_0 tracking. We augmented the data by varying the tempos from $[-14, +14]$ BPM from the original. We then block the data with a size of 200 with 80% overlap and concatenate the raga-id as one-hot vector.

2.3 Architecture design

We designed an auto-encoder architecture with a masked latent space representation to model different gamakas that are tied to different notes and ragas.

By using a masked latent representation, we were able to isolate the key features that distinguish the gamakas of each raga while at the same time, retaining and sharing the common features using the skip connections [11] and shared weights of the encoder and decoder between different ragas. Skip connections also aided in the training to converge faster. There are N masked latent vectors that correspond to N ragas in the dataset. Only the vector that corresponds to the raga is unmasked during training and inference. The encoder has 6 convolutional layers with skip connections after every other layer with ReLU activations. The decoder is identical but with transposed convolutions instead. Dropout and batch normalization layers were used to regularize during training.

2.4 Training

We pre-trained the model on just the pitch contour data from the recordings (i.e. treating it as a denoising problem) so as to condition all the layers. We further pre-trained our model on the pitch contour data from the saraga dataset [12] for raga Abhogi and Kalyani.

For training, the input data is the pitch contour obtained from the MIDI annotations and the target is the pitch contour extracted from the recordings. 20% data was split before pre-processing for validation and testing. We trained the model for 2000 epochs, optimizing using Stochastic Gradient Descent (SGD) with a starting learning rate of 0.001 and a momentum of 0.99. We used the Mean Absolute and Squared Error (MASE) as the loss function which is MSE + MAE. Since MAE is stronger with values < 1 and MSE is stronger otherwise, it helped the model learn both domains equally and converge faster. We used Early stopping and LR scheduler to optimize training.

3. RESULTS AND DISCUSSION

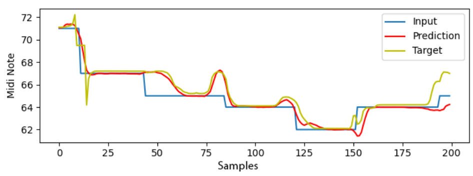

Figure 1. Pitch contour of phrase DMGRSRG

Figure 1 shows an excerpt of phrase in Abhogi [1]. It can be seen that the generated pitch contour closely matches the target.

3.1 Latent Space Visualization

Figure 2 shows the latent space of GamakaNet. We used Principal Component Analysis (PCA) [14] to project the latent vector into 3d space.

The latent vectors learn distinguishable information for the two ragas. Blue points being spread out could be because Kalyani is a melakartha raga [15] with a lot more gamaka variations while Abhogi is a pentatonic raga.

3.2 Tempo dependent gamaka

Figure 3 shows the generated pitch contours in 3 tempos. The values are shifted and re-sampled for readability.

[1] See [13] for Carnatic music notations

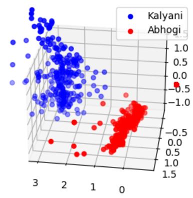

Figure 2. Latent Space vectors of Abhogi and Kalyani

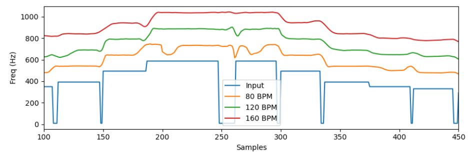

Figure 3. Pitch contour generated in 3 different tempos

We notice the complexity of gamakas getting lower as the tempo increases which is similar to how a musician would handle gamakas at different tempos.

3.3 Music Synthesis

The audio samples are synthesized using a sine wave. The equation for the phase ϕ from interpolated f_0 values at nth sample is given by

$$d\phi(n) = -n\, d\omega \quad (1)$$

$$\implies \phi(n) = \phi(n-1) - 2\pi n(f_n - f_{n-1}) \quad (2)$$

where ω is the angular frequency.

When analyzing the generated plots as shown in figure 1, we notice that the synthesized pitch contours align closely in the middle of the frame and deviate from the expected value towards the edges. This might be because full context is available to the center parts of the frame and towards the ends, only partial information about the past / future values is available to the model. To overcome this issue, we used the overlap-add approach as used in popular algorithms like fast convolution [16] when combining synthesized frames.

4. FUTURE WORK

In this work, we explored a data-driven approach to gamaka synthesis for kalpitha swaras in Carnatic music. The system receives MIDI scores and synthesizes pitch contour for the given notes with gamaka information embedded in it.

We plan to conduct listening test with performance scoring on a held-out test set along with the Turing test [17]. We will invite 10 experienced Carnatic musicians to listen to the audio samples [2] synthesized from the synth detailed in section 3.3. We included the Tanpura in all the samples for completeness and pitch reference. We will use a Likert scale [18] for the scoring. The baseline for our tests will be the Gaayaka software mentioned in section 1.

[2] https://tinyurl.com/knyr4re9

5. REFERENCES

[1] H. S. POWERS, "MODE AND RAGA," *The Musical Quarterly*, vol. XLIV, no. 4, pp. 448–460, 10 1958. [Online]. Available: https://doi.org/10.1093/mq/XLIV.4.448

[2] J. Lochtefeld, *The Illustrated Encyclopedia of Hinduism*, ser. Gale Virtual Reference Library. Rosen Publishing Group, 2002. [Online]. Available: https://books.google.com/books?id=5kl0DYIjUPgC

[3] G. Loy, "Musicians make a standard: the midi phenomenon," *Computer Music Journal*, vol. 9, no. 4, pp. 8–26, 1985.

[4] A. Ashtamoorthy, P. Prasad, S. Dhar, and D. Vijayasenan, "Frequency contour modeling to synthesize natural flute renditions for carnatic music," in *2018 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2018, pp. 172–176.

[5] M.Subramanian, "Carnatic music - automatic computer synthesis of gamakams," *Sangeet Natak*, 2009. [Online]. Available: https://carnatic2000.tripod.com/indexPr1.html

[6] Z. Sasindran and S. S. Garani, "Modeling ornaments in carnatic music signals via wavelets," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 1447–1450.

[7] N. H. Kumar, P. Ashwin, and H. Ananthakrishnan, "Mellisai-an ai generated music composer using rnn-lstms," *International Journal of Machine Learning and Computing*, vol. 10, no. 2, 2020.

[8] M. R. Clayton, "Observing entrainment in music performance: Video-based observational analysis of indian musicians' tanpura playing and beat marking," *Musicae Scientiae*, vol. 11, no. 1, pp. 27–59, 2007.

[9] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014, pp. 659–663.

[10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[11] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[12] A. Srinivasamurthy, S. Gulati, R. Caro Repetto, and X. Serra, "Saraga: Open datasets for research on indian art music," *Empirical Musicology Review 2021; 16 (1): 85-98.*, 2021.

[13] S. Dikshitar, *Sangita Sampradaya Pradarsini*. The Music Academy, 1973, vol. 1. [Online]. Available: http://www.ibiblio.org/guruguha/ssp_cakram1-4.pdf

[14] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[15] R. Sridhar and T. Geetha, "Raga identification of carnatic music for music information retrieval," *International Journal of recent trends in Engineering*, vol. 1, no. 1, p. 571, 2009.

[16] C. S. Burrus and T. Parks, "Convolution algorithms," *Citeseer: New York, NY, USA*, 1985.

[17] R. M. French, "The turing test: the first 50 years," *Trends in cognitive sciences*, vol. 4, no. 3, pp. 115–122, 2000.

[18] A. Joshi, S. Kale, S. Chandel, and D. K. Pal, "Likert scale: Explored and explained," *British journal of applied science & technology*, vol. 7, no. 4, p. 396, 2015.