
Ragesh Rajan M1, Ashwin Vijayakumar2, Deepu Vijayasenan1

1National Institute of Technology Karnataka, India 2Georgia Institute of Technology, Atlanta, USA

mrageshrajan@gmail.com, ashwinkv@gatech.edu, deepu.senan@gmail.com

Abstract

Gamakas, the embellishments and ornamentations used to enhance musical experience, are defining features of Karnatic Music (KM). The appropriateness of using gamaka is determined by aesthetics and is often developed by musicians with experience. Therefore, understanding and modeling gamaka is a significant bottleneck in applications like music synthesis, automatic accompaniment, etc. in the context of KM. To this end, we propose to learn both the presence and the type of gamaka in a data-driven manner using annotated symbolic music. In particular, we explore the efficacy of three classes of features – note-based, phonetic and structural – and train a Random Forest Classifier to predict the existence and the type of gamaka. The observed accuracy is ~70% for gamaka detection and ~60% for gamaka classification. Finally, we present an analysis of the features and find that frequency and duration of the neighbour notes prove to be the most important features.

Index Terms: gamaka, Karnatic Music, Symbolic Music, Random Forest Classifier.

1. Introduction

Karnatic Music (KM) is an ancient classical music system with its origins in South India. A distinct feature of KM is the usage of gamakas – graceful curves in the pitch of a note that serve as embellishments. The aesthetic appeal of a performance is largely determined by the usage of gamaka that is dependent both on the rāga, the fundamental melodic structure in KM, and the musician. The rules for assigning an embellishment to a particular note in KM is neither too open-ended nor very specific [1]. For example, certain musical phrases demand the use of specific gamakas, while certain other phrases allow some amount of flexibility in choosing gamakas. Further, KM performances are predominantly improvisational, and the usage of gamaka varies considerably across musicians. Since there can be multiple aesthetically correct variants of using gamaka and may further vary significantly across musicians, it is hard to encode or fix it as part of musical notation. Therefore, performing both composed music and improvising requires a certain amount of musical maturity.

Importance in Music Synthesis. Existing methods that explicitly model gamaka [2, 3, 4, 5] overcome the complexity involved in understanding gamaka by allowing 1) only a small fraction of gamakas that are predetermined for a particular rāga 2) using expert knowledge to construct templated gamaka in terms of pitch track, tempo and amplitude. By design, these models have multiple drawbacks including that they do not scale to a large number of rāgas and require extensive expert tuning. In the direction of better integration of symbolic music and gamaka, the use of melodic atoms has been proposed in the literature [6]. The melodic atoms include both the note and the gamaka as the building blocks. Again, similar to previous synthesis techniques, it requires experts to decide what the melodic atoms for a rāga are, as there can be innumerable melodic atoms in each rāga.

Gamakas in Karnatic Music. Although a comprehensive and fairly general understanding of gamaka has been elusive, traditional musicologists have classified gamakas into ten, typically based on the playing techniques of the musical instrument, vīna [7]. Further attempts by musicologists to adapt this to singing and more general music performance has seen a trend of increasing complexity and variance across different schools of thought. Therefore, this structured approach is largely abandoned in practical KM training, and most musicians develop a sense for the usage of gamaka by ear and experience. However, Subbarāma Dikshita, an influential musicologist and composer, documented music compositions along with a novel annotation scheme for gamaka in his seminal work, Sangita Sampadaya Pradarshini [8] in the early 20th century. While it is possible to perform the same compositions with other variations, we use this version as the gold standard and learn the nuances of gamaka in a symbolic music setting.

Related Work. Most of the existing works on gamaka prediction and classification use acoustic data for the task. Very few works have addressed the problem of predicting gamaka from annotated symbolic music. As part of expressive music synthesis, the ornamentations in Jazz Guitar has been predicted [9] using the RIPPER algorithm [10]. The extracted features were pitch and duration of the present note and neighbouring notes, chord, key and the perceptual parameters. Using a reduced feature set, an average accuracy of 70% was observed. Note that this was better than a random classifier by only 3.45%. With an increased dataset of 27 songs, classifiers such as SVM, ANN, Decision Trees and KNN were tried [11]. Interestingly, decision trees performed the best at this classification task yielding the best accuracy (78.68%), which was better than the random classifier by only 4.9%. In the context of Indian Classical Music, multiple works have addressed gamaka identification from raw audio signal [12, 13]. Other research efforts on the automatic music composition [14, 15, 16, 17] and singing voice synthesis [18] for Indian Classical Music do not address the problem of predicting gamakas from the score files.

Contributions. In this paper, we describe an approach to detect the presence and type of gamaka associated with a note in KM from the annotated symbolic music. To the best of the authors’ knowledge, this work is the first one of its kind. We propose three sets of features based on the notes, lyrics and
the structure of the song. We train a Random Forest Classifier (RFC) for the detection and classification of gamakas and further, analyse the importance of these features for our task.

2. Background

In the commonly followed tradition of KM, an octave contains a maximum of 12 note positions. Unlike western music, the frequencies associated with the note positions are not fixed in KM. The note frequencies are defined relative to an arbitrary reference frequency. In this work, we denote the note positions using English alphabets and numbers. Deviating from standard convention practiced by musicians, we label the note positions across octaves using numbers, to mean octaves 1, 2 and 3 stand for lower, middle and upper octaves, respectively. For example, the 12 note positions in the lower octave are represented as $S_1$, $r_1$, $R_1$, $g_1$, $G_1$, $m_1$, $M_1$, $P_1$, $d_1$, $D_1$, $n_1$ and $N_1$. Corresponding middle octave positions are represented as $S_2$, $r_2$, ..., $N_2$.

Rāga, the fundamental melodic structure in KM, is characterized by an ordering of notes and associated note-sequences. Rāga also serves as the basis for improvisation and evoking various emotions. Different rāgas can have different number of note positions per octave. For example, the rāga named Mōhanam is a penta-tonic rāga, which has only five notes – $S$, $R$, $G$, $P$ and $D$ – per octave, and these may occur in any arbitrary order.

Gamakas account for the variability of notes in KM. They add naturalness to the music from the KM perspective. A note is considered to be properly defined only when the gamaka associated with it is properly addressed. Note that for the same note, the usage of gamaka may vary from one rāga to another. The performance of both improvisational and composed music depends on the musician’s understanding of the rāga and the usage of gamaka in that melodic framework. Sangītha Sampradāya Pradarshini [8] classifies gamakas into 15 types, out of which, some are very specific to the musical instruments.

3. Data Details

For our experiments, we consider all the songs in one of the popular rāga (Kalyāni) and its derived rāgas1 from the English translation of Sangītha Sampradāya Pradarshini [8]. We digitize the information given in this book to construct a dataset. This dataset contains the lyrics, notes, duration of notes, gamakas and other structural information of 25 songs across nine composers. The number of songs taken from each rāga is listed in the Table 1. There are 9986 notes, out of which, 4013 notes have gamakas and 5973 notes are plain notes. There are only 10 different types of gamakas in the dataset. Due to lack of training samples (less than 10), we neglect two types of gamakas and consider only eight different classes. The different gamakas considered in this paper are: Podi, Kampitham, Sphuritham, Nokku, Etra Jāru, Iraaka Jāru, Orikkar and Othukkal.

4. Experiments

Initially we explore the possibility of detecting the presence of gamaka in a note. For this purpose three different feature sets based on adjacent notes are proposed – absolute and relative frequencies, lyrics (phonetic classes like vowels, fricatives, etc.) and the structure (such as beat cycle information). We select a

---

1 A derived rāga consists only of a subset of the notes of the parent rāga

<table>
<thead>
<tr>
<th>Rāga</th>
<th>Training Set</th>
<th>Test Set 1</th>
<th>Test Set 2</th>
<th>Total Songs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalyāni</td>
<td>5</td>
<td>4</td>
<td>–</td>
<td>9</td>
</tr>
<tr>
<td>Mōhanam</td>
<td>3</td>
<td>2</td>
<td>–</td>
<td>5</td>
</tr>
<tr>
<td>Sāranga</td>
<td>2</td>
<td>2</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td>Yamunā kalyāni</td>
<td>3</td>
<td>2</td>
<td>–</td>
<td>5</td>
</tr>
<tr>
<td>Hamvīrū</td>
<td>–</td>
<td>–</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>13</td>
<td>10</td>
<td>2</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1: Number of songs in each rāga in the dataset

---

training set with 13 songs covering 4 rāgas. Two test sets are constructed. Test set 1 (seen rāga test set) is formed by 10 songs belonging to the same set of rāgas as the training set. The test set 2 (the unseen rāga test set) consists of two songs belonging to rāga Hamvīrū, which is not present in the training set. Thus it is possible to analyze the performance of the system for seen/unseen rāgas separately. Table 1 lists the number of songs in training/test sets. There are a total of 6198 notes in the training set, 3260 notes in the seen rāga test set, and 528 notes in the unseen rāga test set.

4.1. Note-based Features

Note-based features represent the frequency positions and the duration of the notes in the song. The 36 notes covering the three octaves (12 notes per octave) are sorted in the increasing order of frequency. The frequency positions of these notes are represented using numbers from 1 through 36.

We consider a context of five notes, centered around the note under consideration, for feature extraction. The frequency position of the middle note is used as a feature. In addition, the difference in frequency positions of the four adjacent notes with respect to the middle note are used as features. Features also include durations of all the five notes in the context. In addition, we have four more rāga-dependent features. Here, we consider only the notes present in the rāga of the song. For creating these features, rāga-specific frequency positions are assigned to notes. For example, in the rāga called Mōhanam, only five notes – $S$, $R$, $G$, $P$ and $D$ – are present in one octave. Thus the three octaves are represented by numbers from 1 through 15. The rāga-dependent relative positions of the four neighbouring notes are found out with respect to the middle note. Thus the 14 dimensional feature set is formed.

For example, let a set of contextual notes be $S_2$ $R_2$ $S_1$ $D_1$ $S_2$, from a song belonging to the rāga Mōhanam. Let $S_2$ in the middle of the set be the note under consideration. The middle note’s absolute frequency position can be found out to be 13. The relative position of the preceding 2 notes with respect to the middle note will be +2 (note $R_2$) and 0 (note $S_2$). Similarly, for the succeeding notes $D_1$ and $S_2$ it will be -3 and 0, respectively. From the rāga-based representation of Mōhanam, the relative frequency positions of the neighbouring notes $S_2$, $R_2$, $D_1$ and $S_2$ are 0, +1, -1 and 0, respectively.

We use a Random Forest Classifier (RFC) for detecting the presence of gamakas in the notes. The parameters of the classifier are set to the default values. In order to evaluate the performance of note-based features in detecting the gamakas, we perform a cross validation across all the 13 songs in the training set. The cross validation is performed in a round-robin fashion, by excluding each of the 13 songs from the training set and testing on the excluded song. The average accuracy obtained is 71%. From the accuracy score, it can be concluded that this
feature set contains information about the presence/absence of gamakas associated with the notes.

We also explore the use of Artificial Neural Network (ANN) classifier to evaluate the performance of note-based features. Maximum cross validation accuracy obtained using ANN is found to be 67%. We observe that the training accuracy itself is inferior to that using RFC by around 15%. This may be because of the nominal nature of the features used. Therefore we do not explore ANN classifier for the rest of our experiments.

4.2. Phonetic Features

Phonetic features contain information about the phonetic classes of the lyrics of the songs. We try to divide the lyrics into five phonetic classes, namely, vowel, fricative, stop, nasal and glide. Each phoneme in the lyrics of the songs falls into any of the five phonetic classes. To construct this feature set, we consider a context of four phonetic entities from the lyrics associated with each note. Starting and ending phonemes associated with the current note form two entities. Starting phoneme associated with the succeeding note, and the end phoneme associated with the preceding note form the other two phonetic entities. A five dimensional binary vector is formed for each of the above phonetic entities, indicating the phonetic class of that entity. Thus the feature set is a 20 dimensional binary vector.

We use the same set up as in the case of note-based features for conducting experiments to evaluate the performance of phonetic features. The average accuracy is observed to be 56.74%. It seems that the lyrical information does not play a major role in detecting the presence of gamaka in a note.

4.3. Structural Features

These features contain information about the structural elements of the song. In KM, a song is divided into different sections such as pallavi, anupallavi, charanam, swaram, etc. Pallavi, the beginning segment of the song, often gets repeated after every other segment. Pallavi is followed by anupallavi, charanam and swaram. We add 4 binary features to represent the section to which the current note belongs. We also consider other structural elements such as the beginning and end points of beat cycles and sub-beat cycles, the notes starting ‘off-beat’, etc. There are 12 such classes, and a 12-dimensional binary vector indicates the presence of these classes associated with each note. After conducting the experiments on the training set, the average accuracy using the structural features is found to be 57.86%. The accuracy is better than that of the phonetic features by 2.53%. But it is inferior to the accuracy obtained using note-based features by 12.14%. It can be inferred that the structural features do not contain as much information as in the case of note-based features to distinguish between gamakas and non-gamakas.

4.4. Feature Combination

In this experiment, we aim to test the amount of additional information that can be provided by the structural and phonetic features, when used along with the note-based features. We combine all the features to make a 46 dimensional feature vector. With these features, we conduct the experiments on the training set using the RFC. The accuracy obtained after cross validation is 70.13%. This result is inferior to the one obtained for note based features alone. To analyze this, we look at the training accuracy with the same experimental conditions. The training accuracy is found to be 98.6%. The wide gap between the train and cross validation accuracies indicates that there is an over-fit.

The over-fit problem can be reduced by employing regularization. We perform regularization by increasing the number of estimators and/or increasing the minimum number of samples required to split a node. We perform a grid search of these hyper–parameters jointly to get the maximum training cross validation accuracy. Figure 1 shows an example plot of accuracy versus different values of number of estimators.

We use the same set up as in the case of note-based features for conducting experiments to evaluate the performance of phonetic features. The average accuracy is observed to be 56.74%. It seems that the lyrical information does not play a major role in detecting the presence of gamaka in a note.

Another option to tackle the over-fit problem is feature selection. We perform feature selection based on the feature importance parameter of RFC. The importance of a feature is computed as the reduction of information when a node is split using this feature [19]. The variation of cross validation accuracy with respect to the number of selected features is illustrated.
in Figure 2. 17 best features having maximum information re-
duction are selected. All the 14 note-based features happen to
be in the selected features. This shows their importance in de-
tecting gamakas. From the other two sets of features, only two
phonetic features and one structural feature are selected. Hence
it can be concluded that in our experiments, the structural and
phonetic features do not have much of the information regard-
gamakas. The 17 selected features are listed in Table 3.

To evaluate the performance on the test data, we use the op-
timized hyper-parameters and reduced feature set. The results
obtained for different test sets are listed in Table 4. RFC per-
formance is better than the random classifier (classification by
chance) by around 10%. It is also clear that the test accuracies
obtained for the seen rāgas do not deviate much from the cross
validation accuracies. But in the case of unseen rāga, there is a
reduction in test accuracy, especially with the selected features.
This may be because of the fact that, the features selected are
tuned only for the seen rāgas.

Table 3: Selected features

<table>
<thead>
<tr>
<th>Note_Post1</th>
<th>Note_Rāga_Post1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note_Post2</td>
<td>Note_Duration_Post1</td>
</tr>
<tr>
<td>Note_Pre1</td>
<td>Note_Duration_Pre1</td>
</tr>
<tr>
<td>Note_Current</td>
<td>Note_Duration_Post2</td>
</tr>
<tr>
<td>Note_Rāga_Pre1</td>
<td>Note_Duration_Pre2</td>
</tr>
<tr>
<td>Note_Pre2</td>
<td>Charanam</td>
</tr>
<tr>
<td>Note_Rāga_Pre2</td>
<td>Prev_Note_End_Stop</td>
</tr>
<tr>
<td>Note_Duration_Current</td>
<td>Post_Note_Begin_Stop</td>
</tr>
</tbody>
</table>

Table 4: Test accuracies for different features

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Random Classification Accuracy</th>
<th>Full Features</th>
<th>Selected Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seen Rāgas</td>
<td>60.21%</td>
<td>72.47%</td>
<td>72.1%</td>
</tr>
<tr>
<td>Hamviru</td>
<td>57.57%</td>
<td>70.62%</td>
<td>67.03%</td>
</tr>
</tbody>
</table>

4.5. Gamaka Classification

In the next set of experiments, we try to classify the gamakas
associated with a note into 8 different classes detailed in Section
3. The training and test sets contain the same songs as in the
previous experiment. But for the gamaka classification exper-
iment, we consider only the notes which contain gamakas. The
plain notes are removed from the test and training sets. Notes
having gamaka are selected manually from the ground truth.
The training set for this experiment consists of 2534 notes, and
the seen and unseen rāga test sets consist of 1261 notes and
218 notes, respectively.

For this experiment, we follow the same procedure as in the
detection experiment. 22 features are selected, which yield
the best accuracy. Out of these 22 features, 17 are the selected
features from the detection experiment. Five more phonetic fea-
tures are also selected. The classification accuracies for the RFC
using two different feature sets and for different test sets are
listed in Table 5. The accuracies for the random classification
(selecting the most probable class) are also listed in the table. It
can be seen that the accuracies using RFC are better than that
for the random classification, by around 40%.

Table 5: Test accuracies for gamakas

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Random Classification Accuracy</th>
<th>RFC Full Features</th>
<th>RFC Selected Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seen Rāgas</td>
<td>22.8%</td>
<td>58%</td>
<td>38.8%</td>
</tr>
<tr>
<td>Hamviru</td>
<td>24.5%</td>
<td>58.7%</td>
<td>63.5%</td>
</tr>
</tbody>
</table>

4.5. Gamaka Classification

In the next set of experiments, we try to classify the gamakas
associated with a note into 8 different classes detailed in Section
3. The training and test sets contain the same songs as in the
previous experiment. But for the gamaka classification exper-
iment, we consider only the notes which contain gamakas. The
plain notes are removed from the test and training sets. Notes
having gamaka are selected manually from the ground truth.
The training set for this experiment consists of 2534 notes, and
the seen and unseen rāga test sets consist of 1261 notes and
218 notes, respectively.

For this experiment, we follow the same procedure as in the
detection experiment. 22 features are selected, which yield
the best accuracy. Out of these 22 features, 17 are the selected
features from the detection experiment. Five more phonetic fea-
tures are also selected. The classification accuracies for the RFC
using two different feature sets and for different test sets are
listed in Table 5. The accuracies for the random classification
(selecting the most probable class) are also listed in the table. It
can be seen that the accuracies using RFC are better than that
for the random classification, by around 40%.

Table 4: Test accuracies for different features

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Random Classification Accuracy</th>
<th>Full Features</th>
<th>Selected Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seen Rāgas</td>
<td>60.21%</td>
<td>72.47%</td>
<td>72.1%</td>
</tr>
<tr>
<td>Hamviru</td>
<td>57.57%</td>
<td>70.62%</td>
<td>67.03%</td>
</tr>
</tbody>
</table>

5. Conclusion and Future Work

In this work, we detect the presence as well as the type of
gamakas from KM notations, using a data-driven approach. We
propose three different sets of features – based on the frequen-
cies and durations of the notes, phonetic class of the lyrics and
the structure of the song. RFC is used for the detection and clas-
cification. Gamaka detection experiments on different feature
sets reveal that note-based features contain the most relevant in-
formation. Detection accuracies obtained with the full/selected
features are around 72% for the seen rāgas test set. The corre-
spending results for the unseen rāga test set are 70.62% (full
features) and 67.03% (selected features).

For gamaka classification, we use the same experimental set
up as in the detection experiments. The classification accu-
racies for the seen rāgas are around 58% for the full and se-
lected features. accuracies for the unseen rāga are around 57%
using full features, and around 64% using selected features.

Due to the limited amount of training data, we still observe
an over-fit in our experiments. We would like to extend our
work by constructing a larger dataset, for better modeling of
gamakas associated with notes.
6. References


