Raag detection using Hidden Markov Models

Dhruva Kumar

Abstract— **The task of *raag* detection is attempted using Hidden Markov Models (HMM). Every *raag* is characterized uniquely by certain melodic transitions. HMM is used to learn these unique transitional probablities for every *raag*. By using only the pitch contour as a feature, HMM works quite well by identifying a test set of 16 samples spanning across different modalities with an accuracy of 81.25%.**

I. INTRODUCTION

A *raag* is a melodic concept in Indian Classical Music (ICM) on which most of the songs from the ICM domain are based on. Technically, a *raag* is a combination of a set of melodic notes played or sung in a specific way. The melodic notes and the transition between these notes are defined for every *raag* and uniquely characterize them. These chararcteristic sequences are called *aaroh* (ascending sequence), *avroh* (descending sequence) and *pakad* (a succinct sequence which contains both the ascent and descent). Based on top of these technical rules for the given *raag*, a *raag* performance consists of various ornamental embellishments and nuances like *meend* (glissando), *kan-swar* (grace notes) and *gamaks* (a percussive voice production technique) which are used to bring out the essence and flavor of the *raag*. Every *raag* performance is associated with a *rasa* or an emotion which is brought out vividly through these nuances. Certain notes are held longer and are emphasized more than the others. These are called *vadi* (most important) *swar* (note). The second most important note is called the *samvadi swar*.
The melodic notes of a *raag* are based on the chromatic scale and a *raag* generally consists of 5 to 12 chromatic pitches. Most of the *raag* performances are based on the three octaves C3 to B5 (*mandra*, *madhya* and *taar saptak*) and this range is considered for the quantization of pitch features.
Every *raag* performance has a drone tone (*tanpura*) played continuously in the background. The pitch of this drone tone is the tonic frequency set by the musician for the recording.

The present work uses Hidden Markov Models in an attempt to learn the transitional probabilites between the melodic notes which uniquely defines a *raag* as described previously.

II. MOTIVATION

Identifying a *raag* is important for musicians in finding the underlying key for every musical piece. The task of identifying a *raag* is currently done by trained classical musicians with 10 to 15 years of experience.
Identifying a *raag* is also useful for musical therapy. By understanding which type of *raag* the person likes to listen to and therefore the emotions or *rasa* associated with it, one can get an insight into the person's psychological experience. It can also be used in music recommendation systems where similar songs can be generated in the playlist. Here, the similarity index is the underlying *raag* or the underlying mood of the person at the time.

III. RELATED WORK

Automatic *raag* classification has been discussed and attempted previously. Various methods have been used like pitch class distribution (PCD), HMM, finite automata modeling of the musical rules, pitch class string formation.
[3] uses PCDs and pitch-class dyad distributions (PCDDs) as features derived from pitch contours. PCDDs are essentially bi-grams of pitch classes. The pitch contour was detected using the Harmonic Product Spectrum (HPS) algorithm. The dimensionality of the features was reduced to 50 from 156 using PCA and a Multivarite Normal (MVN) classifier was applied. An accuracy of 94% was was achieved on a test set of 17 samples. 17 *raag* classes were used. GTraagDB [2] was used as the dataset.
[4] uses HMM and a string matching algorithm to detect *raags*. The pitch contour was used as the feature which was detected using the autocorrelation metod. Only 2 *raag* classes were used and an accuracy of 77% was achieved on test set of 31 samples. An improved performance of 10% was achieved with an additional stage that involved identifying catch phrases called *pakad* in the test sequences. The dataset used was collected manually.

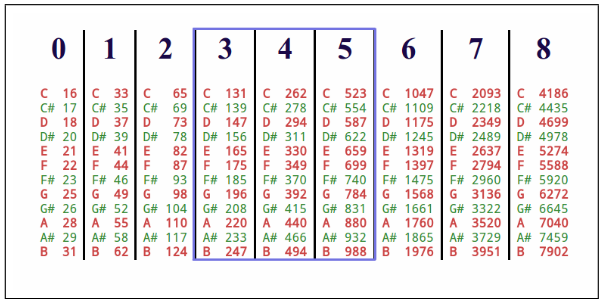
IV. METHOD

A. Dataset

A subset of GTraagDB [2] was used for training and testing. GTraagDB consists of 31 raags of duration ranging from 3 minutes to 60 minutes. The recordings consists of solo vocals and solo instruments. The percussions and the drone typically heard in *raag* performances have been removed from the recordings. Each raag is also annotated with its tonic frequency.

TABLE I
Raag CLASSES USED FOR TRAINING AND ITS PITCHES

Raag	**Pitches used in the raag**
Bihag	C D E F F# G A B
Darbari	C D Eb F G Ab Bb
Desh	C D E F G A Bb B
Gaud Malhar	C D E F G A Bb B
Yaman	C D E F F# G A B



0	1	2	3	4	5	6	7	8
C 16	C 33	C 65	C 131	C 262	C 523	C 1047	C 2093	C 4186
C# 17	C# 35	C# 69	C# 139	C# 278	C# 554	C# 1109	C# 2218	C# 4435
D 18	D 37	D 73	D 147	D 294	D 587	D 1175	D 2349	D 4699
D# 20	D# 39	D# 78	D# 156	D# 311	D# 622	D# 1245	D# 2489	D# 4978
E 21	E 41	E 82	E 165	E 330	E 659	E 1319	E 2637	E 5274
F 22	F 44	F 87	F 175	F 349	F 699	F 1397	F 2794	F 5588
F# 23	F# 46	F# 93	F# 185	F# 370	F# 740	F# 1475	F# 2960	F# 5920
G 25	G 49	G 98	G 196	G 392	G 784	G 1568	G 3136	G 6272
G# 26	G# 52	G# 104	G# 208	G# 415	G# 831	G# 1661	G# 3322	G# 6645
A 28	A 55	A 110	A 220	A 440	A 880	A 1760	A 3520	A 7040
A# 29	A# 58	A# 117	A# 233	A# 466	A# 932	A# 1865	A# 3729	A# 7459
B 31	B 62	B 124	B 247	B 494	B 988	B 1976	B 3951	B 7902

Fig. 1. Range of pitch scales chosen for pitch quantization as shown by the blue bounding box. Includes the middle three octaves, *mandra* (C3-B3), *madhya* (C4-B4) and *taar* (C5-B5) *saptak*

Since enough samples were not present for each raag class, only 5 raag classes were used for training. Each raag class in the training set consist of 3 recordings of the sarod and sitar. The test set consists of 16 samples of vocals and sitar.

B. Feature extraction, quantization and normalization

A pitch contour for the input recording was obtained using Praat [2]. Praat uses an enhanced autocorrelation function to estimate the most likely pitch for every window of the waveform. Post processing is done to choose the most likely pitch path across the windows. This is done to prevent octave errors.

Other popular methods for pitch detection like Yin [3] were tested for various samples in the dataset, but Praat gave the best results for most of them.

The continous pitch contour for all the recordings were then quantized to the nearest chromatic scale pitch frequency. The relation between the frequency and the chromatic scale pitch for a 88 key piano is given by the following formula:

$$f(n) = \sqrt[12]{2}^{(n-49)} \times 440Hz \tag{1}$$

where n is the key number on the piano. The 49th key on a 88 key piano is A4 and it has a frequency of 440 Hz. Since each recording is performed in a different scale, there is a need for normalization to shift the recordings to a standard scale. This is done by the annotated tonic frequency for each recording in the dataset. The quantized pitch above is normalized relative to C4 which has a frequency of 261.63 Hz. Only the middle three octaves, C3 to B5, are considered for pitch quantization since most of the raag performances are within this range. In the Indian Classical Music domain, these are called the *mandra*, *madhya* and *taar saptak*. This leads to a total of 36 quantization levels.

C. Learning: Baum Welch

The HMM model was learnt using the Baum Welch procedure. It is an iterative EM procedure alternating between the E and M step, trying to maximize the cost function $P(O|\lambda)$ until the convergence criteria is met.

a) Initialization: The HMM model was initialized to a fully connected structure. The initial state vector π was uniformly distributed. The probability of starting from any state was equal. The transition matrix, A and the emission matrix B were randomly initialized. The number of hidden states N was chosen to be 36. The idea was that at any given time, the pitch can lie in any of the chosen chromatic scale pitch state. The number of discrete observations M was determined by the 36 quantization levels as described in the previous section.

b) Expectation: The expectation step consists of computing the probability of the observation sequence given the model $P(O|\lambda)$. This is done by calculating the forward and backward variables α and β. The scaling used was as described in [1].

- Forward procedure:
  1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1) \qquad 1 \le i \le N$$
$$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)}$$
$$\hat{\alpha_1}(i) = c_1 \alpha_1(i)$$

  2) Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}]\ b_j(O_{t+1}) \tag{2}$$
$$1 \le t \le T-1,\ 1 \le j \le N$$
$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{3}$$
$$\hat{\alpha_t}(i) = c_t \alpha_t(i) \tag{4}$$

  3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \frac{1}{\prod_{\tau=1}^{T} c_\tau}$$
$$\log P(O|\lambda) = -\sum_{t=1}^{T} \log c_t \tag{5}$$

  Equation (4) gives the value of the forward variable after scaling.
- Backward procedure:
  1) Initialization:

$$\beta_T(i) = 1 \qquad 1 \le i \le N \tag{6}$$
$$\hat{\beta_T}(i) = c_T \beta_T(i)$$

2) Induction:

$$\beta_T(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$
$$T-1 \geq t \geq 1, \ 1 \leq i \leq N$$
$$\hat{\beta}_t(i) = c_t \beta_t(i)$$

The scaling in (3) is used here.

For multiple sequences, α^l, β^l and c^l is computed for each sequence l.

c) Modification: Given the forward and backward variables, the model parameters π, A, B are updated so that the cost function $P(O|\lambda)$ is maximized. The re-estimated paramters $\overline{\pi}$, $\overline{A}$, $\overline{B}$ are as follows:

$$\begin{aligned}\overline{\pi_i} &= \text{expected number of times in state} \\ &\quad S_i \text{ at time } (t=1) \\ &= \gamma_1(i)\end{aligned}$$

$$\begin{aligned}\overline{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to } S_j}{\text{expected number of transitions from state } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T=1} \gamma_t(i)} \\ &= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(j)} \\ &= \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \hat{\beta}_t(j)/c_t}\end{aligned}$$

For multiple observations,

$$= \frac{\sum_{l=1}^{L-1} \sum_{t=1}^{T_l-1} \hat{\alpha}_t^l(i) a_{ij} b_j(O_{t+1}^l) \hat{\beta}_{t+1}^l(j)}{\sum_{t=1}^{T_l-1} \hat{\alpha}_t^l(i) \hat{\beta}_t^l(j)/c_t^l} \tag{7}$$

$$\begin{aligned}\bar{b}_j(k) &= \frac{\text{expected no. of times in state } j \text{ with symbol } v_k}{\text{expected no. of times in state } j} \\ &= \frac{\sum_{t=1, O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \\ &= \frac{\sum_{t=1, O_t=v_k}^{T} \hat{\alpha}_t(j) \hat{\beta}_t(j)/c_t}{\sum_{t=1}^{T} \hat{\alpha}_t(j) \hat{\beta}_t(j)/c_t}\end{aligned}$$

For multiple observations,

$$= \frac{\sum_{l=1}^{L} \sum_{t=1, O_t^l=v_k}^{T_l} \hat{\alpha}_t^l(j) \hat{\beta}_t^l(j)/c_t^l}{\sum_{t=1}^{T_l} \hat{\alpha}_t^l(j) \hat{\beta}_t^l(j)/c_t^l} \tag{8}$$

$\gamma_t(i)$ is the probability of being in state i at time t given the observations sequence and the model λ

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} \tag{9}$$

$\xi_t(i,j)$ is the probability of being in state S_i at time t and S_j at time $t+1$, given the model and observation sequence.

$$\begin{aligned}\xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}\end{aligned} \tag{10}$$

d) Convergence criteria: The EM steps were alternated iteratively until the convergence criteria was met. The convergence criteria was assigned as when the absolute of the difference in the log likelihood of $P(O|\lambda)$ between successive iterations was less than 0.1.

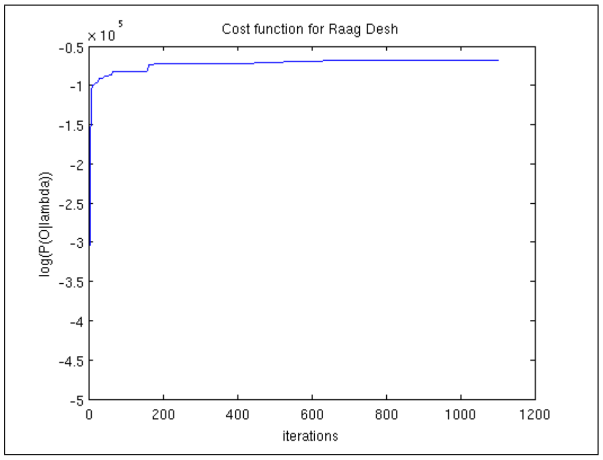


Fig. 2. The cost function: $\log P(O|\lambda)$ for *raag desh* during training until the convergence criteria was met

V. Results

Once the model parameters were learnt for each *raag* class, each test sequence was tested using the forward-backward procedure as described before. The $argmax$ of $\log P(O|\lambda)$ among the *raag* classes was chosen to be the detected *raag* for the test sequence in subject.

An accuracy of 81.25% was obtained on the test set. 13 out of 16 test samples were correctly identified. Out of the 16 test samples, 4 were solo vocal pieces and 12 were solo instrumentals. 11 out of 12 instrumental pieces were correctly identified. 2 out of 4 of the vocal pieces were correctly identified. The results are pretty good given the fact that vocal pieces were not included in the training data due to shortage of data. *Raag desh* was misclassified twice as *raag gaud malhar*. This can be explained by the fact that both the *raags* share the same set of musical notes. They however differ in the transition between the notes. This transition was not accurately learnt by the HMM model.

TABLE II

CONFUSION MATRIX ON THE TEST SET

	Bihag	Darbari	Desh	Gaud M.	Yaman
Bihag	2	0	0	0	0
Darbari	0	3	0	1	0
Desh	0	0	1	2	0
Gaud Malhar	0	0	0	4	0
Yaman	0	0	0	0	3

This might be due to the pitch detection not always being correct. Another possibility is that the *tanpura* drone which is played continuously in the background in every piece might interfere with the notes that should be identified. A solution to this would be to identify and remove the drone tone in the preprocessing stage. All the test samples were accurately identified within the second or third best guess.

VI. CONCLUSION AND FUTURE WORK

Hidden Markov Model has proved to work quite well for *raag* detection. It does a pretty good job of identifying the underlying transitional probabilites between the musical notes which is characteristic of every *raag*. The current work was done on a small dataset. The next task would be to procure a strong training set in terms of quality so that the trained model would correctly identify test sequences spanning across different modalities. This can potentially be done by explicity adding the characteristic phrases, *aaroh*, *avroh*, *pakad* and *aalap* for every raag. The weights of these training sequences can be increased compared to the others so that the model learns the melodic tranisitions better.
As described previously, the sympathetic string drone sounds played in the background can be identified and removed during the preprocessing stage so that it doesn't interfere in the model learning the transitional probabilites. Another improvement is to use a better pitch detection algorithm.

In terms of features, more features like the amplitude can be incorporated in the training phase.

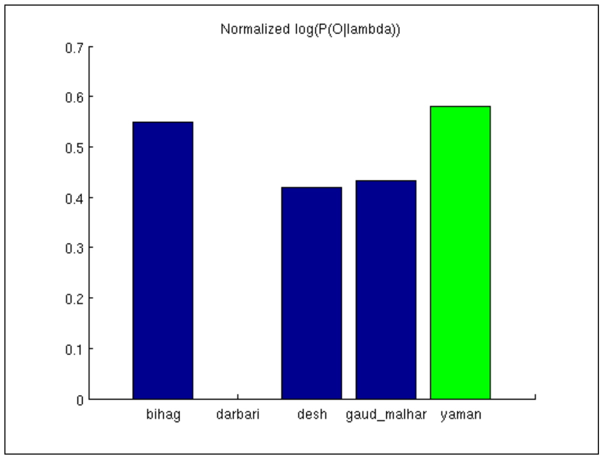


Fig. 3. The comparison of log likelihood of the test sequence (*raag yaman*) given the models. Here, it correctly predicts the *raag*. *Raag bihag* comes close since it shares the same set of notes.

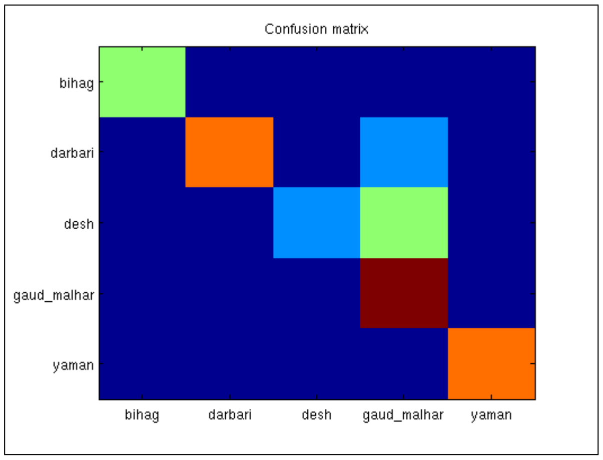


Fig. 4. Confusion matrix on the test set

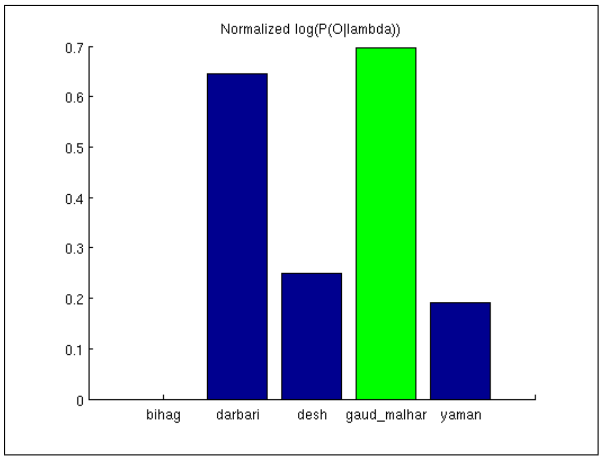


Fig. 5. *Raag darbari* being misclassified as *raag gaud malhar*

In addition to the unique sequences in every *raag*, each *raag* is also characterized by the most emphasized note, *vadi* and the second most emphasized note *samvadi*. This can be encoded as the amplitdue along with the pitch to be used as features.
In conclusion, HMM has proved to be quite a good candidate for *raag* detection and there is scope for improvement to achieve better results.

ACKNOWLEDGMENT

I would like to thank Dr. Dan Lee for the excellent lectures due to which understanding the content was a breeze. I would also like to thank Bhoram, Steve and Chris for their invaluable guidance.

REFERENCES

[1] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition": Proc. IEEE, Vol. 77, No. 2, pp. 257-286: February 1989.
[2] GTraagDB arranged by Parag Chordia, http://prernagupta.com/parag/data.html
[3] Chordia, P. 2006. "Automatic Raag Classification of Pitchtracked Performances Using Pitch-Class and Pitch-Class Dyad Distributions." In Proceedings of the International Computer Music Conference, pp. 314?321.
[4] Gaurav Pandey, et al "Tansen: A system for automatic Raga identification" IICAI, 2003