TANSEN : A SYSTEM FOR AUTOMATIC RAGA IDENTIFICATION

Gaurav Pandey, Chaitanya Mishra, and Paul Ipe

Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur, India
{gpandey,cmishra,paulipe}@iitk.ac.in

Abstract. Computational Musicology is a new and emerging field which draws heavily from Computer Science, particularly Artificial Intelligence. Western Music has been under the gaze of this community for quite some time. However, Indian music has remained relatively untouched. In this paper, which is an illustration of the application of techniques in AI to the study of Indian music, we present an approach to solve the problem of automatic identification of Ragas from audio samples. Our system, named Tansen, is based on a Hidden Markov Model enhanced with a string matching algorithm. The whole system is built on top of an automatic note transcriptor. Experiments with Tansen show that our approach is highly effective in solving the problem.

Key words: Computational Musicology, Indian Classical Music, Note Transcription, Hidden Markov Models, String Matching

1   Introduction and Problem Definition

Indian classical music is defined by two basic elements - it must follow a Raga (classical mode), and a specific rhythm, the Taal. In any Indian classical composition, the music is based on a drone, ie a continual pitch that sounds throughout the concert, which is a tonic. This acts as a point of reference for everything that follows, a home base that the musician returns to after a flight of improvisation. The result is a melodic structure that is easily recognizable, yet infinitely variable.

A Raga is popularly defined as *a specified combination, decorated with embellishments and graceful consonances of notes within a mode which has the power of evoking a unique feeling distinct from all other joys and sorrows and which possesses something of a transcendental element.* In other words, a Raga is a characteristic arrangement or progression of notes whose full potential and complexity can only be realised in exposition. This makes it different from the concept of a scale in Western music. A Raga is characterised by several attributes, like its Vaadi-Samvaadi, Aarohana-Avrohana and Pakad, besides the sequence of notes which denotes it. It is important to note here that no two performances of the same Raga, even two performances by the same artist, will be identical.

A certain music piece is considered a certain Raga as long as the attributes associated with it are satisfied. This concept of Indian classical music, in that way, is very open.

In this freedom lies the beauty of Indian classical music and also, the root of the our problem, which we state now. The problem we addessed was "Given an audio sample (with some constraints), predict the underlying Raga". More succinctly,

Given	an audio sample
Find	the underlying Raga for the input
Complexity	a Raga is highly variable in performance

Though we have tried to be very general in our approach, some constraints had to be placed on the input. We discuss these constraints in later sections.

Through this paper we expect to make the following major contributions to the study of music and AI. Firstly, our solutions is based primarily on techniques from speech processing and pattern matching, which shows that techniques from other domains can be purposefully extended to solve problems in computational musicology. Secondly, the two note transcription methods presented are novel ways to extract notes from samples of Indian classical music and give very encouraging results. These methods could be extended to solve similar problems in music and other domains.

The rest of the paper is organized as follows. Section 2 highlights some of the uselful and related previous research work in the area. We discuss the solution strategy in detail in Section 3. The test procedures and experimental results are presented in Section 4. Finally, Section 5 lists the conclusions and future directions of research.

2 Previous Work

Very little work has taken place in the area of applying techniques from computational musicology and artificial intelligence to the realm of Indian classical music. Of special interest to us is the work done by Sahasrabuddhe et al. [4] and [3]. In their work, Ragas have been modelled as finite automata which were constructed using information codified in standard texts on classical music. This approach was used to generate new samples of the Raga, which were technically correct and were indistinguishable from compositions made by humans.

Hidden Markov Models [1] are now widely used to model signals whose functions are not known. A Raga too can be considered to be a class of signals and can be modelled as an HMM. The advantage of this approach is the similarity it has with the finite automata formalism suggested above.

A Pakad is a catch-phrase of the Raga, with each Raga having a different Pakad. Most people claim that they identify the Raga being played by identifying the Pakad of the Raga. However, it is not necessary for a Pakad be sung without any breaks in a Raga performance. Since the Pakad is a very liberal part of the performance in itself, standard string matching algorithms were not

gauranteed to work. Approximate string matching algorithms designed specifically for computer musicology, such as the one by Iliopoulos and Kurokawa [2] for musical melodic recognition with scope for gaps between independent pieces of music, seemed more relevant.

Other relevant works which deserve mention here are the ones on Query By Humming [5] and [7], and Music Genre Classification [6]. Although we did not follow these approaches, we feel that there is a lot of scope for using such low level primitives for Raga identification, and this might open avenues for future research.

3   Proposed Solution

Hidden Markov models have been traditionally used to solve problems in speech processing. One important class of such problems are those involving word recognition. Our problem is very closely related to the word recognition problem. This correspondence can be established by the simple observation that Raga compositions can be treated as words formed from the alphabet consisting of the notes used in Indian classical music.

We exploited this correspondence between the word recognition and Raga identification problems to devise a solution to the latter. This solution is explained below. Also presented is an enhancement to this solution using the Pakad of a Raga. However, both these solutions assume that a note transcriptor is readily available to convert the input audio sample into the sequence of notes used in it. It is generally cited in literature that "Monophonic note transcription is a trivial problem". However, our observations in the field of Indian classical music were counter to this, particularly because of the permitted variability in the duration of use of a particular note. To handle this, we designed two independent heuristic strategies for note transcription from any given audio sample. We explain these strategies later.

3.1   Hidden Markov Models

Hidden Markov models (HMMs) are mathematical models of stochastic processes, i.e. processes which generate random sequences of outcomes according to certain probabilities. A simple example of such a process is a sequence of coin tosses. More concretely, an HMM is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called *transition probabilities*. In a particular state, an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome not the state that is visible to an external observer. So states are "hidden" and hence the name hidden Markov model.

In order to define an HMM completely, the following elements are needed [1].

- The number of states of the model, N

- The number of observation symbols in the alphabet, M.
- A set of state transition probabilities

$$A = \{a_{ij}\} \tag{1}$$

$$a_{ij} = p\{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N \tag{2}$$

where q_t denotes the current state.
- A probability distribution in each of the states,

$$B = \{b_j(k)\} \tag{3}$$

$$b_j(k) \geq p\{\alpha_t = v_k | q_t = j\}, 1 \leq j \leq N, 1 \leq k \leq M \tag{4}$$

where v_k denotes the kth observation symbol in the alphabet and α_t the current parameter vector.
- The initial state distribution, $\pi = \{\pi_i\}$ where,

$$\pi_i = p\{q_1 = i\}, 1 \leq i \leq N \tag{5}$$

Thus, an HMM can be compactly represented as

$$\lambda = (A, B, \pi) \tag{6}$$

.

Hidden Markov models and their derivatives have been widely applied to speech recognition and other pattern recognition problems [1]. Most of these applications have been inspired by the strength of HMMs, ie the possibility to derive understandable rules, with highly accurate predictive power for detecting instances of the system studied, from the generated models. This also makes HMMs the ideal method for solving Raga identification problems, the details of which we present in the next subsection.

3.2 HMM in Raga Identification

As has been mentioned earlier, the Raga identification problem falls largely in the set of speech processing problems. This justifies the use of hidden Markov models in our solution. Two other important reasons which motivated the use of HMM in the present context are:

- The sequences of notes for different Ragas are very well defined and a model based on discrete states with transitions between them is the ideal representation for these sequences [3].
- The notes are small in number, hence making the setup of an HMM easier than other methods.

This HMM, which is used to capture the semantics of a Raga, is the main component of our solution.

Construction of the HMM Used The HMM used in our solution is significantly different from that used in, say word recognition. This HMM, which we call λ from now on, can be specified by considering each of its elements separately.

- Each note in each octave represents one state in λ. Thus, the number of states, **N=12x3=36** (Here, we are considering the three octaves of Indian classical music, namely the Mandra, Madhya and Tar Saptak, each of which consist of 12 notes).
- The transition probability $A = \{a_{ij}\}$ represents the probability of note j appearing after note i in a note sequence of the Raga represented by λ.
- The initial state probability $\pi = \{\pi_i\}$ represents the probability of note i being the first note in a note sequence of the Raga represented by λ.
- The outcome probability $B = \{B_i(j)\}$ is set according to the following formula

$$B_i(j) = {}^{0,i\neq j}_{1,i=j} \tag{7}$$

Thus, at each state α in λ, the only possible outcome is note α.

The last condition takes the *hidden* character away from λ, but it can be argued that this setup suffices for the representation of Ragas, as our solution distinguishes between performances of distinct RaGas on the basis of the exact order of notes sung in them and not on the basis of the embellishments used. A small part of one such HMM is shown in the figure below:
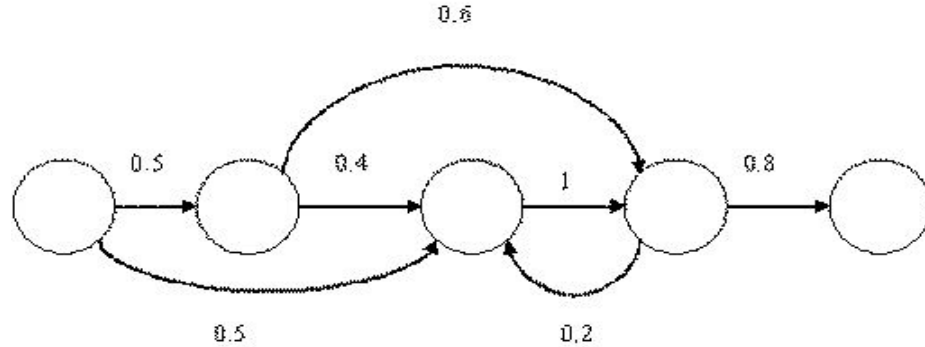

Fig. 1. A Segment of the HMM used

Using the HMMs for Identification One such HMM λ_I, whose construction is described above, is set up for each Raga I in the consideration set. Each of these HMMs is trained, i.e. its parameters A and π (B has been pre-defined) is calculated using the note sequences available for the corresponding Raga with the help of the Baum-Welch learning algorithm [13].

After all the HMMs have been trained, identifying the closest Raga in the consideration set on which the input audio sample is based, is a trivial task. The sequence of notes representing the input is passed through each of the HMMs constructed, and the index of the required Raga can be calculated as

$$Index = argmax(\log p(O|\lambda_I)), 1 \leq I \leq N_{Ragas} \qquad (8)$$

To complete the task, the required Raga can be determined as $Raga_{Index}$.

This preliminary solution gave reasonable results in our experiments (Refer to Section 4). However, there was still a need to improve the performance by incorporating knowledge into the system. This can be done through the Pakad approach, which we discuss next.

3.3 Pakad Matching

It is a well-established notion in the AI community that the incorporation of knowledge related to the problem being addressed into a system enhances its ability to solve the problem. One such very powerful piece of information about a Raga is its Pakad. This information was used to improve the performance of Tansen.

Pakad is defined as *a condensed version of the characteristic arrangement of notes, peculiar to each Raga, which when repeated in a recital enables a listener to identify the Raga being played*. In other words, Pakad is a string of notes characteristic to a Raga to which a musician frequently returns while improvising in a performance. The Pakad also serves as a springboard for improvisational ideas; each note in the Pakad can be embellished and improvised around to form new melodic lines. One common example of these embellishments is the splitting of Pakad into several substrings and playing each of them in order in disjoint portions of the composition, with the repetition of these substrings permitted. In spite of such permitted variations, Pakad is a major identifying characteristic of a Raga and is used even by experts of Indian classical music for identifying the Raga been played.

The very features of the Pakad give way to a string matching approach to solve the Raga identification problem. Pakad matching can be used as a reinforcement for an initial estimation of the underlying Raga in a composition. We devised two ways of matching the Pakad with the input string of notes in order to strengthen the estimation done as per Section 3.2. The incorporation of this step makes the final identification process a multi-stage one.

δ-Occurence with α-Bounded Gaps As mentioned earlier, the Pakad has to appear within the performance of a Raga. However, it rarely appears in one segment as a whole. It is more common for it to be spread out, with substrings repeated and even other notes inserted in between. This renders simple substring matching algorithms mostly insufficient for this problem. A more appropriate method for matching the Pakad is the *δ-occurence with α-bounded gaps* algorithm [2]. The algorithm employs dynamic programming and matches individual notes

from the piece to be searched, say t, identifying a note in the complete sample, say p, as belonging to t only if:

1. there is a maximum difference of δ between the current note of p and the next note of t
2. the position of occurence of the next note of t in p is displaced from its ideal position by atmost α

However, this algorithm assumes that a piece(t) can be declared present in a sample(p) only if *all* notes of the piece(t) are present in the sample(p) within the specified bounds. This may not be true in our case because of the inaccuracy of note transcription (Refer to 3.4). Hence, for each Raga I in the consideration set, a score γ_I is maintained as

$$\gamma_I = \frac{m_I}{n_I}, 1 \le I \le N_{Ragas} \tag{9}$$

$\quad m_I = $ maximum number of notes of the Pakad of Raga I identified
$\quad n_I = $ number of notes in the Pakad of Raga I
This score is used in the final determination of the Raga.

n-gram Matching Another method of capturing the appearance of the Pakad within a Raga performance is to count the frequencies of appearance of successive n-grams of the Pakad. Successive n-grams of a string are its substrings starting from the begining and going on till the end of the string is met. For example, successive 2-grams of the string *abcde* are *ab*, *bc*, *cd* and *de*. Also, to allow for minor gaps between successive notes, each n-gram is searched in a window of size 2n in the parent string. Based on this method, another score is maintained according to the formula,

$$score_I = \Sigma_n \Sigma_j freq_{j,n,I} \tag{10}$$

$freq_{j,n,I} = $ number of times the jth n-gram of the Pakad of
$\qquad\qquad$ Raga I is found in the input.
This score is also used in the final determination of the underlying Raga.

Final Determination of the Underlying Raga Once the above scores have been calculated, the final identification process is a three-step one.

1. The probability of likelihood $prob_I$ is calculated for the input after passing it through each HMM λ_I and the values so obtained are sorted in increasing order. After reordering the indices as per the sorting, if

$$\frac{prob_{N_{Ragas}} - prob_{N_{Ragas}-1}}{prob_{N_{Ragas}-1}} > \eta \tag{11}$$

$\qquad$ then,

$$Index = \mathbf{N_{Ragas}}$$

2. Otherwise, the values γ_I are sorted in increasing order and indices set accordingly. After this arrangement, if

$$prob_{N_{Ragas}} > prob_{N_{Ragas}-1}, \text{ and } \frac{\gamma_{N_{Ragas}} - \gamma_{N_{Ragas}-1}}{\gamma_{N_{Ragas}-1}} > n_\eta$$

then,

$$Index = \mathbf{N_{Ragas}}$$

3. Otherwise, the final determination is made on the basis of the formula, where K is a predefined constant

$$Index = argmax(\log p(o|\lambda_I) + K * score_I), 1 \le I \le N_{Ragas} \qquad (12)$$

This three-step procedure is used for the identification of the underlying Raga in the latest version of Tansen. The three steps enable the system to take into account all probable features for Raga identification, and thus display good performance. We discuss the performance of the final version of Tansen in Section 4.

3.4   Note Transcription

The ideas presented in subsections 3.2 and 3.3 were built on the assumption that the input audio sample has already been converted into a string of notes. The main hurdle in this conversion with regard to Indian classical music is the fact that notes are permitted to be spread over time for variable durations in any composition. Here, we present two heuristics based on the pitch of the audio sample, which we used to derive notes from the input. They are very general and can be used for any similar purpose.

The American National Standards Institute (1973) defines pitch as *that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from high to low*. Although this definition of pitch is not very concrete, but largely speaking, pitch of sound is the same as its frequency and shows the same behaviour. From the pitch behaviour of various audio clips, we oberved two important characteristics of the pitch structure, based on which are the following two heuristics.

The Hill Peak Heuristic This heuristic identifies notes in an input sample on the basis of hills and peaks occuring in the its pitch graph. A sample pitch graph is shown in the figure below -:

A simultaneous observation of an audio clip and its pitch graph shows that the notes occur at points in the graph where there is a complete reversal in the sign of the slope, and in many cases, also when there isn't a reversal in sign, but a significant change in the value of the slope. Translating this into mathematical terms, given a sample with time points $t_1, t_2, ..., t_{i-1}, t_i, t_{i+1}, ..., t_n$

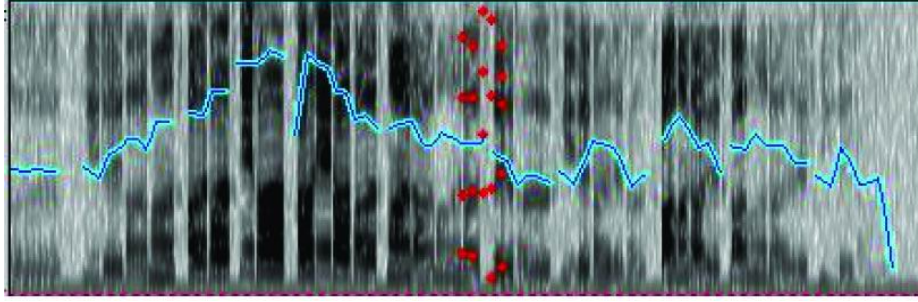**Fig. 2.** Sample Pitch Graph

and the corresponding pitch values $p_1, p_2, ..., p_{i-1}, p_i, p_{i+1}, ..., p_n$, t_i is the point of occurence of a note only if

$$\frac{\frac{p_{i+1}-p_i}{t_{i+1}-t_i} - \frac{p_i-p_{i-1}}{t_i-t_{i-1}}}{\frac{p_i-p_{i-1}}{t_i-t_{i-1}}} > \epsilon \tag{13}$$

Once the point of occurrence of a note has been determined, the note can easily be identified by finding the note with the closest characteristic pitch value. Performing this calculation over the entire duration of the sample gives the string of notes corresponding to it. An important point to note here is that unless the change in slope between the two consecutive pairs of time points is significant, it is assumed that the last detected note is still in progress, thus allowing for variable durations of notes.

The Note Duration Heuristic This heuristic is based on the assumption that in a composition a note continues for at least a certain constant span of time, which depends on the kind of music considered. For example, for compositions of Indian classical music, a value of 25ms per note usually suffices. Corresponding notes are calculated for all pitch values available. A history list of the last k notes identified is maintained including the current one (k is a pre-defined constant). The current note is accepted as a note of the sample only if it is different from the dominant note in the history, i.e. the note which occurs more than m times in the history (m is also a constant). Sample values of k and m are 10 and 8. By making this test, a note can be allowed to extend beyond the time span represented by the history. A pass over the entire set of pitch values gives the set of notes corresponding to the input sample.

As has been mentioned, the two heuristics robustly handle the variable duration problem. We discuss the performance of these heuristics in Section 4.

4 Experiments and Results

4.1 Test Procedures

Throughout this paper we have stressed on the high degree of variability permitted in Indian classical music. Since covering this wide expanse of possible compositions is not possible, we placed the following constraints on the input in order to test the performance of Tansen:

1. There should be only one source of sound in the input sample
2. The notes must be sung explicitly in the performance
3. The whole performance should be in the *G-Sharp* scale

These constraints make note transcription of Raga compositions much easier. Collection of data was done manually because because not many Raga performances satisfying all the above constraints were readily available. Once the required testing set was obtained, the following test procedure was adopted to test the performance of Tansen:

1. The input was fed into an audio processing software, Praat [8] and pitch values were extracted with window sizes of 0.01 second and 0.05 second. These two sets of pitch values were named p_1 and p_5 respectively.
2. p_1 was fed into the Note Duration heuristic and p_5 into the Hill Peaks heuristic, and the two sets of derived notes, namely n_1 and n_5 were saved. (For justification see Section 4.2)
3. The Raga identification was done with both n_1 and n_5. If both of them produced the same Raga as output, that Raga was declared as the final result. In case of a conflict, the Raga with the higher HMM score was declared as the final result.

Rigorous tests were performed using the above procedure on a Pentium 4 1.6 GHz machine running RedHat Linux 8.0. The results of these tests are discussed in the following section.

4.2 Results

There were three clearly identifiable phases in the development of Tansen, and a track was kept of the performance of each phase. We discuss the results of the experiments detailed in 4.1 for each of the phases separately.

Note Transcription Note transcription was the first phase accomplished in the development of Tansen. As explained in 3.4, the strategies employed for extracting the notes, namely the Hill Peaks heuristic and the Note Duration heuristic, very robustly capture the duration variability problem in Indian classical music. Since comparison of two sets of notes for a given music piece is a very subjective problem, the only method for checking the performance of these two methods was manual inspection. A rigorous comparison of notes derived

through the two strategies and the actual set of notes for the input was done by the authors as well as other individuals not connected to the project. The consistent observations were -:

1. many more notes were extracted than were actually present in the input
2. many of the notes extracted were displaced from the actual corresponding note by small values
3. the performance of the two heuristics was quite similar since both these methods are based on the same fundamental concept of pitch

Inspite of these drawbacks, the results obtained were encouraging, and the above errors were very well accomodated in the algorithms which made use of the results of this stage, ie the HMM and Pakad matching parts. Thus, the effect on the overall performance of Tansen was insignificant.

A very important point to note here is the role of the sampling rate used to derive pitch values from the original audio sample. The individual performance of the note transcription methods varies with this sampling rate as follows -:

1. The Hill Peaks heuristic is based on the changing slopes in the pitch graph of the sample. Thus, for best performance, it requires a low sampling rate since with a high sampling rate, variations in the pitch graph will be much more and too many notes may be identified.
2. The Note Duration heuristic assumes a minimum duration for each note in the performance and allows for repetition of notes by keeping a history of past notes. So, for best performance, it requires a high sampling rate so that notes are not missed due to a low sampling rate.

Plain Raga Identification The preliminary version of Tansen was based only on the hidden Markov model used (Refer to Section 3.2). For testing the performance of this version, tests were performed as detailed in Section 4.1 and the accuracy was noted, which is tabulated below.

Raga	Test Samples	Accurately identified	Accuracy
Yaman Kalyan	15	12	80%
Bhupali	16	12	75%
Total	31	24	77%

Table 1. Results of Plain Raga Identification

The results obtained were very encouraging, particularly because the method used was simple pattern matching using HMMs. In order to improve the performance, the Pakad matching method was incorporated, whose results we discuss next.

Raga Identification with Pakad Matching The latest version of Tansen uses both hidden Markov models and Pakad matching to make the final determination of the underlying Raga in the input. When tests were performed on this version of Tansen, the following results were obtained.

Raga	Test Samples	Accurately identified	Accuracy
Yaman Kalyan	15	12	80%
Bhupali	16	15	94%
Total	31	27	87%

Table 2. Results of Raga Identification with Pakad Matching

A comparison with the previous table shows that incorporation of Pakad matching, which in our case is a method of equipping Tansen with knowledge, improves the performance significantly. Addition of this additional step results in a reinforcement of the results obtained from simple HMM based matching. A closer inspection of the results shows that the performance for Raga Bhupali increases by much more than that for Raga Yaman Kalyan. This variation in individual performances is becuase the Pakad used for the former was much more established and popularly used than the one for the latter. Thus, it is important that the knowledge used to make the system more powerful must be correct.

The final results obtained from Tansen are very encouraging. The results obtained should also be seen in the light of the fact that both Ragas belong to the **Kalyan Thaat**, which essentially means that their note subsets belong to the same set of notes. They also, have similar Pakads and are close in structure. The fact that Tansen was able to distinguish between these two ragas with good accuracy shows that Tansen has shown initial success in solving the problem of raga identification.

5 Conclusions

In this paper, we have presented the system Tansen for automatic Raga identification, which is based on hidden Markov models and string matching . A very important part of Tansen is its note transcriptor, for which we have proposed two (heuristic) strategies based on the pitch of sound.

Our strategy is significantly different from those adopted for similar problems in Western and Indian classical music. In the former, systems generally use low level features like spectral centroid, spectral rolloff, spectral flux, Mel Frequency Cepstral Coefficients etc to characterize music samples and use these for classification [5]. On the other hand, approaches in Indian classical music use concepts like finite automata to model and analyse Ragas and similar compositions [3]. Our problem, however, is different. Hence, we use probabilistic automata constructed on the basis of the notes of the composition to achieve our goal.

Though the approach used in building Tansen is very general, there are two important directions for future research. Firstly, a major part of Tansen is based on heuristics. There is a need to build this part on more rigorous theoretical foundations. Secondly, the constraints on the input for Tansen are quite restrictive. The two most important problems which must be solved are estimation of the base frequency of an audio sample and multiphonic note identification. Solutions to these problems will help improve the performance and scope of Tansen.

Acknowledgements

References

1. L. 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. C. S. Iliopoulos and M. Kurokawa: "String Matching with Gaps for Musical Melodic Recognition": Proc. Prague Stringology Conference, pp. 55-64: 2002.
3. H. V. Sahasrabuddhe: "Searching for a Common Language of Ragas": Proc. Indian Music and Computers: Can 'Mindware' and Software Meet?: August 1994.
4. R. Upadhye and H. V. Sahasrabuddhe: "On the Computational Model of Raag Music of India": Workshop on AI and Music: 10th European Conference on AI, Vienna: 1992.
5. G. Tzanetakis, G. Essl and P. Cook: "Automatic Musical Genre Classification of Audio Signals": Proc. International Symposium of Music Information Retrieval, pp. 205-210: October 2001.
6. A. Ghias, J. Logan, D. Chamberlin and B. C. Smith: "Query by Humming - Musical Information Retrieval in an Audio Database": Proc. ACM Multimedia, pp. 231-236: 1995.
7. H. Deshpande, U. Nam and R. Singh: "MUGEC: Automatic Music Genre Classification": Technical Report, Stanford University: June 2001.
8. P. Boersma and D. Weenink: "Praat: doing phonetics by computer": Institute of Phonetic Sciences, University of Amsterdam (`www.praat.org`)

9. M. Choudhary and P. R. Ray: "Measuring Similarities Across Musical Composi-
   tions: An Approach Based on the Raga Paradigm": Proc. International Workshop
   on Frontiers of Research in Speech and Music, pp. 25-34: February 2003.
10. S. Dixon: "Multiphonic Note Identification": Proc. 19th Australasian Computer
    Science Conference: Jan-Feb 2003.
11. W. Chai and B. Vercoe: "Folk Music Classification Using Hidden Markov Models":
    Proc. Internation Conference on Artificial Intelligence: June 2001.
12. A. J. Viterbi: "Error bounds for convolutional codes and an asymptotically optimal
    decoding algorithm": IEEE Transactions on Information Theory, Volume IT-13,
    pp. 260-269: April 1967.
13. L. E. Baum: "An inequality and associated maximization technique in statistical
    estimation for probabilistic functions of Markov processes": Inequalities, Volume
    3, pp. 1-8: 1972.
14. L. E. Baum and T. Petrie: "Statistical inference for probabilistic functions of finite
    state Markov chains": Ann.Math.Stat., Volume 37, pp. 1554-1563: 1966.