Music Information Retrieval: Raga Identification using Machine Learning

A Project Report

submitted by

PADMASUNDARI

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

in

INDUSTRIAL MATHEMATICS AND SCIENTIFIC COMPUTING

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

May 2016

DECLARATION

I declare that the work presented in this report titled **"Music Information Retrieval: Raga Identification using Machine Learning"** is the result of investigations carried out by me at Indian Institute of Technology, Madras under the supervision of Prof. Hema. A. Murthy, Department of Computer Science and Engendering and Dr. N. Narayanan, Department of Mathematics. In keeping with the general practice of reporting scientific observations, due acknowledgement has been made wherever the work described is based on the findings of other investigations. Any omission which might have occurred due to oversight or error in judgement is regretted.

Chennai - 600 036 (Padmasundari)

May 2016 (MA14M007)

THESIS CERTIFICATE

This is to certify that the thesis entitled **Music Information Retrieval: Raga Identification using Machine Learning**, submitted by **Padmasundari**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bonafide record of the project work carried out by her under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Hema. A. Murthy
Professor
Department of CSE
IIT-Madras - 600036

Dr. N. Narayanan
Assistant Professor
Department of Mathematics
IIT-Madras - 600036

Place: Chennai

Date:

DEDICATED TO

My Beloved Parents

My Revered Music Guru

My Respected Parents-in-law

ACKNOWLEDGEMENTS

First and foremost, I sincerely thank Prof. Hema A. Murthy for her constant guidance and motivation throughout the course of this project. She supported me with patience, guided me with her extensive knowledge, allowing me the room to work in my own way as a friend. One simply could not wish for a better or friendlier project advisor.

I owe my gratitude to Dr. N. Narayanan, for kindly agreeing to be my project guide from the Department of Mathematics. I thank him for allowing me to undertake and complete this work with all his support and valuable feedback.

I am indebted to every member of the music and speech groups (DON Lab and MS Lab under Prof. Hema.A.Murthy and the Speech and Vision Lab under Prof. C.Chandra Sekhar) of the Department of Computer Science and Engineering for the many useful technical discussions and all the help offered to me in the time of need, without the slightest hesitation. My special thanks are due to Mr. Krishnaraj Sekhar P.V. for spending days together to generate the dataset for my experiments.

I thank the Departments of Mathematics and ComputerScience and Engineering of IITMadras, the Institute and the CompMusic Project for providing the funding and support. I am also grateful to Prof. C. Chandra Sekhar and Dr. Anurag Mittal for tolerating me in their courses and sharing their knowledge through their enlightening class lectures.

I take this opportunity to acknowledge and thank everyone of my program(M.Tech.)

mates, hostelmates and the Institute employees who may have helped me directly or indirectly, one or more times during the course of my M.Tech. project at IIT-Madras.

Finally, my thanks and deepest regards go to my husband Mr.K.Sriram for supporting me in every way during my stay at the Institute. I specially thank his brother Dr.K.Kartik for the proof reading and providing his views to improve the manuscript of this report. Needless to say, it would have never been possible for me to complete this work without the constant support and encouragement that I received from my spouse and all family members.

ABSTRACT

KEYWORDS: Music Information Retrieval, Raga Identification, Machine Learning, Locality Sensitive Hashing

Music information retrieval (MIR) is an interdisciplinary field bridging the domains of mathematics, statistics, signal processing, machine learning, computer science, musicology, biology, and more. Some of the most common MIR tasks include finding similar audio items, music classification and archival, playlist recommendation, source separation, etc.

Rāga is a quintessential aspect of Indian classical music. There are about two hundred rāgas being currently performed in Carnatic music concerts and thousands are possible in theory. The Internet has brought a wealth of audio data and rāga identification can provide high level information about music data to become a basis for music search and archival.

Rāgas are primarily characterised by melodic time-frequency (T-F) trajectories. With hundreds of rāgas being in regular use, performing rāga identification as a machine learning task is certainly not trivial. Rāgas allow much scope for improvisation and elaboration within the bounds of certain specified norms, making the task of rāga identification difficult for machine learning algorithms.

There have been several computational approaches proposed to determine the identity of a rāga, however the techniques work only on subset(s) of rāgas and also perform poorly in terms of scalability.

The initial part of this work addresses the question of finding similar rāgas required for the verification framework and proposes to use *Locality Sensitive Hashing* (LSH) as a solution. The later part of the work includes the new proposed method for rāga identification using LSH, validating the approach and demonstrating improved performance through experiments.

Interestingly, LSH is close to how humans perceive rāgas. Experiments with 3000 rāga queries, performed on a standard dataset consisting of 100 concerts, each with 8-10 rāgas, leading to 927 items and 182 rāgas, achieves an average accuracy of 71%. Most importantly LSH does not suffer from scalability issues of the earlier approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN	Approximate Nearest Neighbour
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
LCS	Longest Common Subsequence
LCSS	Longest Common Segment Set
LR	Likelihood Ratio
LSH	Locality Sensitive Hashing
MAP	Maximum a Posteriori
MIR	Music Information Retrieval
NN	Nearest Neighbour
PCD	pitch-class Distribution
PCDD	pitch-class Dyad Distribution
RLCS	Rough Longest Common Subsequence
SVM	Support Vector Machine
T-F	Time-Frequency

CHAPTER 1

INTRODUCTION

Humans have developed abilities to distinguish pitches, melodies, harmonies, rhythms, instruments, emotions, contents, genre, song structure, etc... Retrieving such information from low level audio data is however proven to be difficult task. The retrieval of high-level musical information from the audio signal data continues to be challenging research problem in MIR.

A rāga[1] in Indian classical music has several components - primordial sound (nāda), tonal system (svarās or solfège), tonic (śruti), scale, norms of ascend (ārōha) and descend (avarōha), ornaments (gamaka) and important tones (Rao [1930]; Ayyangar [1972]; Bhagyalekshmi [1990]). A rāga may be annotated using the ascending and descending notes. However, specific ornamented notes, important tones, combinations of notes, relation between the notes, the note sequences and characteristic musical motifs, all contribute to giving a rāga its unique identity. There are hundreds of rāgas being performed currently in the concerts, and thousands are possible in theory (Ayyangar [1972]; Bhagyalekshmi [1990]). Rāga allows much scope for improvisation and elaboration within the bounds of the specified norms.

It is important to note that the location of the notes in Indian classical music is a continuum as indicated in Miron *et al.* [2011]. Owing to the highly improvisational nature of Indian music:

[1] https://en.wikipedia.org/wiki/Carnatic_raga

- no two renditions of a rāga may contain identical rendition of the same musical phrases or motifs and
- two rāgas may have the same notes and identical ārōha and avarōha templates but can be rendered very differently with unique characteristics

Rāgas are hard to be adequately defined in terms of the characteristic phrases due to the inherent subjective nature of the motifs, intra motif variabilities, subjective human perceptions of similarity and context dependency. Rāga identification as a music classification problem has interesting applications, however, has been a challenging task in the field of MIR.

The Locality Sensitive Hashing(LSH) technique is applied to locate the nearest neighbour to a given audio query for which the rāga is to be identified. The rāga of the audio query is decided using the labels of the nearest neighbour matches retrieved from the LSH query. A confidence measure is also suggested for the rāga prediction using the predictor feature scores.

1.1 Outline of the Work

The current work is primarily motivated by Dutta *et al.* [2015] where a rāga verification approach is adopted to confirm the identity of a rāga claim in Carnatic (south Indian classical) music. A rāga verification framework is suggested in the paper as an alternative to scalability. However, the approach requires a set of cohorts to be defined manually for each rāga. An attempt to automate this step of defining the cohorts for a given rāga claim has led to the proposed method for rāga identification.

LSH is a technique that can match time series that are similar, into the same bucket. Figure 3.5 depicts the block diagram of the proposed solution method for

the rāga identification problem.

1.2 Highlights of the proposed method

- New method for rāga identification in Carnatic music using LSH.
- Does not suffer from scalability issues. The method is easily scalable to a number of rāgas and repertoires.
- LSH Method is close to the human way of performing the task of raga identification.

1.3 Organization of this report

Chapter 2 provides a brief review of related works and motivic analysis. Chapter 3 describes the proposed method for rāga identification. Chapter 4 details the dataset used in this work, the experiments performed and the summary of the results. Chapter 5 includes a summary of the entire work with concluding remarks and directions for future work.

CHAPTER 2

MOTIVIC ANALYSIS

2.1 Introduction

Several prior attempts have focussed on the problem of recognition of rāgas in Carnatic and Hindustani music. Early approaches to rāga identification discussed in Section 2.2 were mostly based on note or svara transcription of the music. That is, the rāga identification problem was primarily approached as a problem of determining the notes or the scale being used in the melody. However, the concept of a rāga is much more than just a combination of notes or svarās.

In a musical work, *motif* refers to a recurring distinctive form, theme, idea or a dominant feature of the music. In Indian classical music, rāgas are often identified using typical motifs or short signature phrases or 'pakads' that contribute significantly to providing an unique identity to a rāga. The phrases are not necessarily based on the notes or scale but are basically trajectories of frequency or pitch with time. Different rāgas may have the same set of notes yet derive their unique identities from clear distinctive motifs. Some of the more recent works discussed in Section 2.3 have focussed on the time-frequency trajectory or motivic analyses of rāgas in Indian classical music.

Though rāgas are accepted to have characteristic melodic phrases or motifs, they are not limited or constrained by any combinatorial numbers or forms of the motifs appearing in them. Rāgas allow immense scope for improvisations.

The motivic patterns may vary highly with every rendition of any rāga in various aspects such as their relative position of occurence, length, notes they take, strength(energy), pitch and duration of the notes, intonations of the notes, etc.. Rāga identification methods aimed at obtaining typical motifs or signatures of rāgas are not scalable with respect to number of rāgas. Section 2.4 describes the rāga verification approach suggested in Dutta *et al.* [2015] as an alternative to scalability.

With hundreds of rāgas being in regular use, performing rāga identification as a machine learning task is certainly not trivial. The rāga verification approach adopted in Dutta *et al.* [2015] to perform the identification task requires a set of *cohorts* or similar rāgas (or rāgas with similar movements and subtle differences) to be defined for every rāga. Section 2.5 proposes the LSH method as a solution to the question of finding similar rāgas and Section 2.6 provides a quick overview of LSH. Section 2.7 concludes this chapter with some final remarks.

2.2 Early Approaches to Rāga Identification

Early computational approaches (Upadhye *et al.*[1992]; Sahasrabuddhe [1994]) to rāga recognition in Indian classical music have used finite automata to model and analyse rāgas. In Pandey *et al.* [2003], the authors use HMM and string matching approaches to classify two similar rāgas based on note transcription of the audios. The classification system in Pandey *et al.* [2003] is based largely on heuristics and imposes constraints on the input. Chordia *et al.*[2007] use pitch-class distributions (PCDs) and pitch-class dyad distributions (PCDDs) to capture rāga information and classifiers like SVM, maximum a posteriori (MAP), and Random

Forests to perform the classification. In Sridhar *et al.*[2009], authors again use audio transcription and string matching to test and identify 3 rāgas sung by 3 different singers. Other attempts (Dighe *et al.* [2013*a*,*b*]; Arthi *et al.* [2011]; Ishwar *et al.* [2012]; Krishna *et al.* [2011]; Kumar *et al.* [2014]) include use of ārōha and avarōha based template models that model the steady note transcription along with n-gram models to perform rāga identification.

However, symbolic notations have proven to be inadequate to capture the nuances in the phraseology of ragas in Indian classical music. As mentioned in the introduction to this chapter (Section 2.1), two rāgas may have the same notes and identical ārōha and avarōha templates but can be rendered very differently with unique characteristics. Figures 2.1 and 2.2 provide examples of time-frequency plots of a descending (avarōha) sequence of notes for the rāgas Tōdi and Sindhubhairavi. Sindhubhairavi is a rāga in Carnatic music adopted from the Hindustani music (Rag Bhairavi)[1]. The rāga has exactly the same svarās and ārōha-avarōha template as Tōdi (Ayyangar [1972]; Bhagyalekshmi [1990]). The rendering of the two is significantly different and gives each rāga its unique identity.

Further, most of the early attempts as discussed above have used only few rāgas and the dataset used is not representative of the existing variety of rāgas. This limits the significance of the results obtained. The high accuracies reported may be due to the use of limited number of rāgas or small sizes of the datasets.

[1] http://www.karnatik.com/hcragatable.shtml

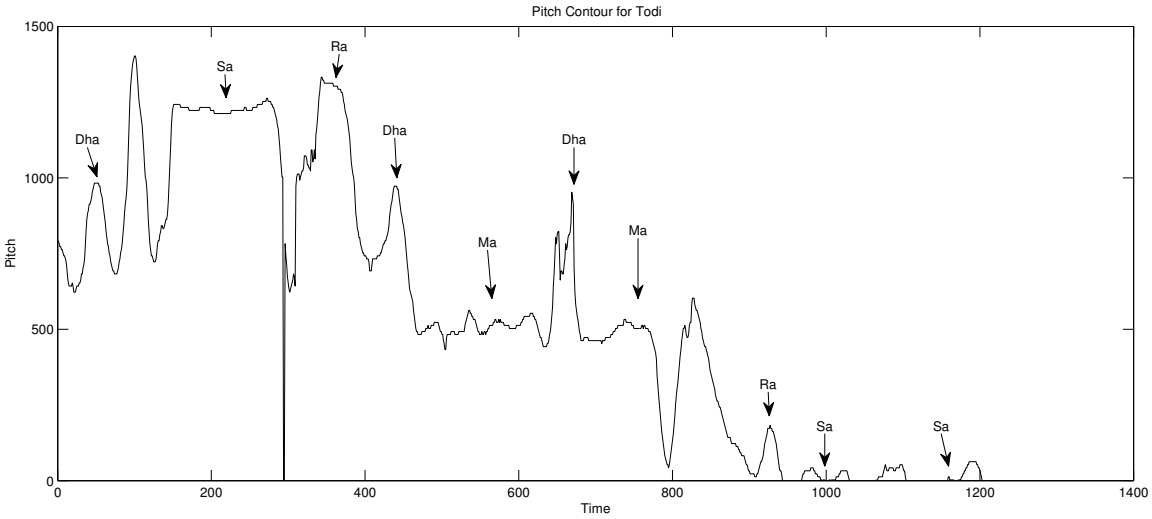


Figure 2.1: Labeled Pitch Contour for Tōdi

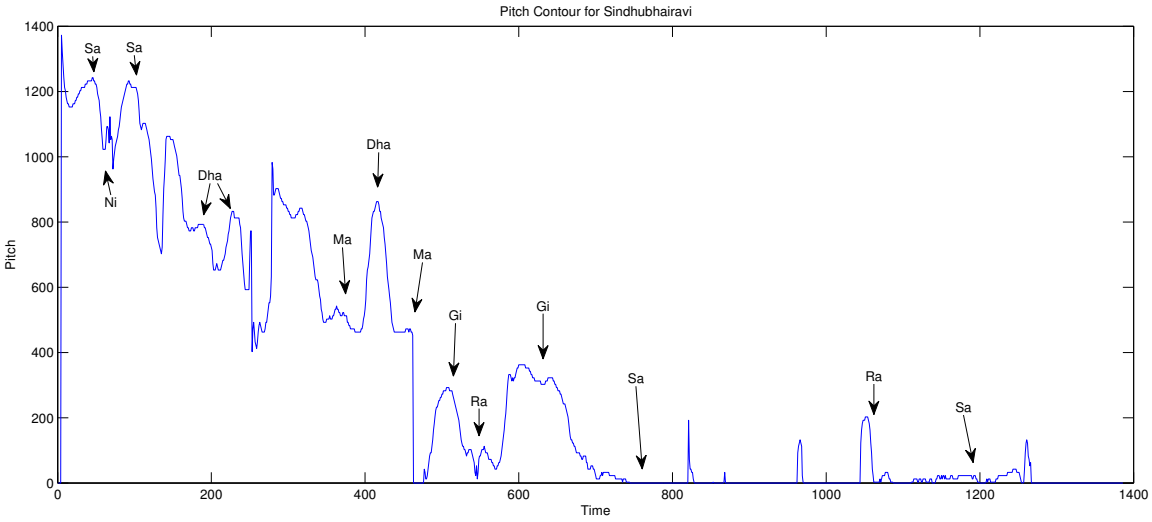


Figure 2.2: Labeled Pitch Contour for Sindhubhairavi

2.3 Motivic Analysis

Rāgas are primarily characterised by melodic time-frequency (T-F) **motif**s. Ishwar *et al.* [2012] study the relevance of analyses of motifs in understanding rāgas. Here, the authors show with examples that transcription of motifs into svaras could lead to significant loss of information. Authors conduct experiments with five set of rāgas to claim and confirm that a motif identifies a rāga uniquely. Motif

identification is performed here using Hidden Markov Models (HMMs) where each motif is represented by an HMM.

Automatic detection of recurring basic patterns from audio is of much relevance in MIR. Ross *et al.* [2012] have considered the problem of automatic detection and segmentation of melodic motifs from audio signals for Hindustani classical music. A two step approach here involves firstly identifying a limited set of candidate motifs based on rhythm cycle structure and secondly computing similarity distance of the candidate motifs from the reference template. Similarity measures traditionally used in time-series matching are shown to perform well in the context of melodic motif detection. Ishwar *et al.* [2013] and Dutta *et al.*[2014b] use Rough Longest Common Subsequence (RLCS) (Lin *et al.* [2011]), a variant of the LCS for matching and discovering motifs in ālāpanās in Carnatic music. More recent works (Dighe *et al.* [2013*a*], Dutta *et al.*[2014], Rao *et al.* [2014]; Gulati *et al.* [2015]) have conducted further systematic analyses of motifs paving way to automatic motif detection in Indian classical music.

One of the key challenges with the motif approach are those of the intra motif melodic pattern variations. Figure 2.3 illustrates for two rāgas, Tōdi and Sindhubhairavi, how two renditions of a rāga may not contain identical rendition of the same musical phrases or motifs. Another challenge is that of sensitivity to the distance measure chosen. Gulati *et al.* [2015] attempt to use various similarity measures for matching melodic motifs of Indian classical music and even compare the motif pattern variations in the Carnatic and Hindustani forms.

Also as already noted in Section 2.1, using typical motifs or signatures for rāga identification is not scalable, when the number of rāgas under consideration increases. Most of the attempts discussed above have again used only small

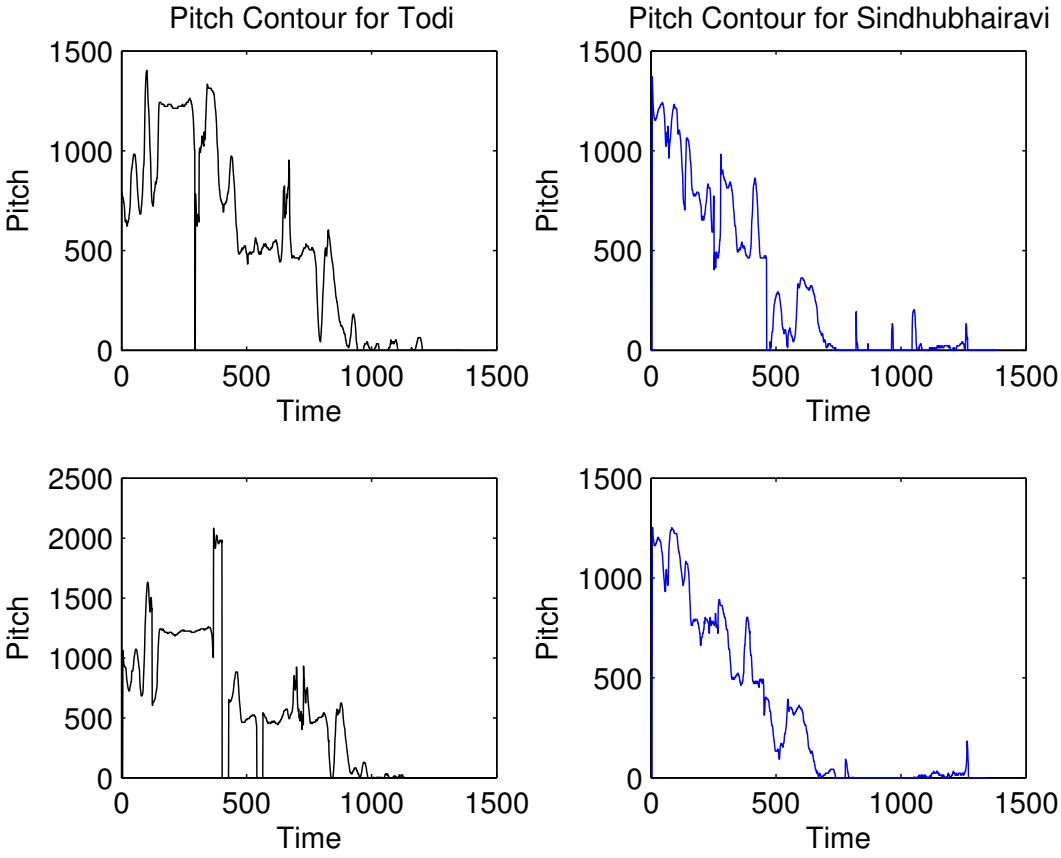


Figure 2.3: Two renditions of Tōdi and Sindhubhairavi having unidentical T-F motifs

datasets, not representative of the varieties of rāgas and repertoires available in practice currently.

2.4 Rāga Verification Framework

Dutta *et al.* [2015] take a different approach to rāga identification, close to a human listener's way of performing the task. A very common approach that listeners take to identify a rāga is to compare phrases with similar patterns occurring in compositions familiar to them in-order to ascertain the identity of the rāga. This way the listener actually performs only a verification to confirm if the phrase belongs

to a known rāga which is the rāga of the composition. Clearly, the rāgas of the candidate compositions are rāgas with similar movements and with subtle differences. Therefore, in Dutta *et al.* [2015], rāga identification problem is transformed to a verification problem as:
Given an audio clip along with a claim for the rāga's name, verify the claim to say if the audio clip belongs to the rāga or not.

In Carnatic music, *Pallavi* generally refers to the first one or two lines of lyrics set to a melody which is the dominating theme of a composition. Pallavi is usually based on the elementary melodic curve of the rāga and is repeated after each segment of the composition Nijenhuis [1974]. Dutta *et al.* [2015] uses pallavi lines (refered to as *one-liners* in the paper) of compositions to obtain typical motifs of a rāga. Thus, Dutta *et al.* [2015] compare the audio clip with the one-liners of the claimed rāga and its cohorts to perform the verification. A modified RLCS, called Longest Common Segment Set (LCSS), is used to perform the motif matching. The experiments were performed on a set of 17 rāgas.

2.5 Proposed Solution

The verification framework described in the above Section 2.4 requires the cohorts for every rāga to be defined with care by a professional musician. For the verification approach to be scalable, it is extremely important to automate this step of defining the cohorts for the rāgas. Automation could bring in more objectivity to the definition of cohorts or the notion of similarity of rāgas and improve the overall scalability of the approach.

The search for similar rāgas may be generalized as an Approximate Near-

est neighbor search problem. Nearest Neighbour methods aim at locating the exact training examples that are closest to the test example (Figure 2.4) and therefore are time consuming having linear complexity. Approximate Nearest

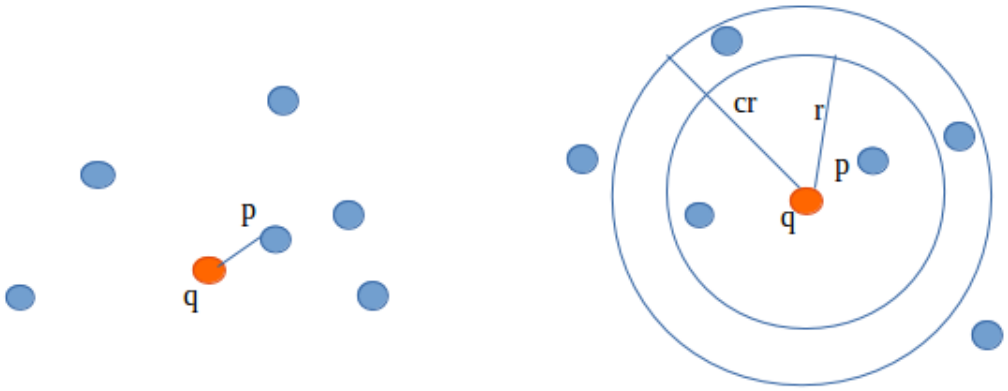


Figure 2.4: Nearest Neighbour $c-$approximate Neighbour

Neighobour(ANN) methods aim only at an approximate neighbour[2] (Figure 2.4) allowing a specified bound on the maximum approximation error. ANN methods[3] may not guarantee to return the actual nearest neighbor in every case, but find the nearest neighbors in most of the cases, with improved speed or efficient memory usage.

LSH is an ANN Search Algorithm which has often proved useful in retrieving similar musical items from large databases (Ryynnen *et al.*[2008]; Casey *et al.*[2007]; Cano *et al.* [2002]). LSH avoids computing the similarity of every pair of items. One general approach to LSH is to *hash* items several times, such that similar items are more likely to be hashed to the same bucket than dissimilar items are (Figure 2.5). Any pair hashed to the same bucket for any of the hashings is a candidate pair. Similarities are checked only for the candidate pairs. Thus, LSH enables to focus on similar items without comparing all the pairs; hence can perform retrieval

[2] $c-$approximate neighbour: Given test example q, find p such that $||p-q|| \leq cr$ if there exists an $r-$near neighbour p such that $||p-q|| \leq r$

[3] https://en.wikipedia.org/wiki/Nearest_neighbor_search#Approximate_nearest_neighbor

of similar items in reduced sublinear time.

Figure 2.5: Nearby points hashed to the same bin more often

LSH-based method may work effectively in finding similar rāgas and also provide a scalable solution to the problem. LSH is also easy to implement and the theoretical complexity is well studied. Challenge however is that of the choice of feature vector to represent a rāga.

2.6 Locality Sensitive Hashing

Locality Sensitive Hashing(LSH) is a randomized algorithm technique used to quickly find nearest neighbours from a large training database. The basic concept here is to hash similar data points or instances to the same bucket with high probability (Slaney*et al.*[2008]]. LSH aims to maximize the probability of a collision for similar or nearby points. If two points p and q in R^n are close together, then there is a high probability P_1 that even after a projection operation h the two points remain close to each other and if they are far apart, there is a low probability P_2 ($< P_1$) that they are nearby. Figure[4] 2.6 has two examples showing projections of two close

[4]This figure is taken from Slaney and Casey [March 2008]

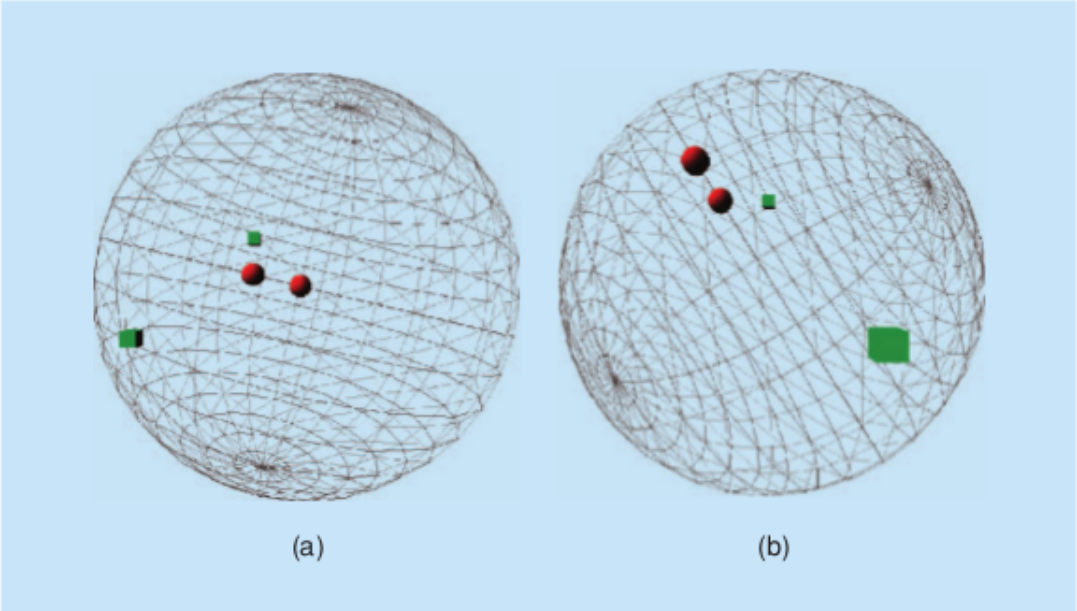


Figure 2.6: LSH: Projection

(circles) and two distant (squares) points onto the printed page. Mathematically,

$$Pr(h(p) = h(q)) \geq P_1 \; for \; ||p - q|| \leq r_1$$

$$Pr(h(p) = h(q)) \leq P_2 \; for \; ||p - q|| \geq r_2 = cr_1$$

where $c > 1$ and $Pr(h(p) = h(q))$ is the collision probability.

The idea is to start with a random projection operation $h(p) = p.x$ where x is a vector in R^n with components selected at random from a Gaussian distribution, say $N(0, 1)$. The projection is quantized to a set of hash bins as

$$h^{x,b}(p) = \left\lfloor \frac{p.x + b}{w} \right\rfloor$$

where w is the width of each quantization bin and b is a random variable uniformly distributed between 0 and w. This function h maps any data point p from high-dimensional to a low-dimensional subspace. Note that due to the linearity of the dot product, the distance between $h^{x,b}(p)$ and $h^{x,b}(q)$ is proportional to the distance

between p and q; that is

$$||h^{x,b}(p) - h^{x,b}(q)|| \propto ||p - q||$$

Keep track of the points close to the query point and consider different projections. The points that appear close to the query points in more than one projection are the ones of interest.

Consider $g = (h_1, h_2, ..., h_k)$ where $h_1, h_2, ..., h_k$ are k hash functions defined as above performing k dot products in parallel. The vectors x and b are chosen at random with replacement. It is easy to see that

- k independent projections transform p to k real numbers
- collision probabilities drop with increase in k
- small value of k increases the number of dissimilar points colliding
- probability ratio $\frac{P_1}{P_2}$ is magnified as k increases
  $\left(\frac{P_1}{P_2}\right)^k > \frac{P_1}{P_2}$ *since* $P_1 > P_2$
- increasing w increases number of buckets
- varying w effects a trade-off between the table size and the search size
  - Large table with smaller final linear search
  - More compact table with more points to consider in final search

Success is when the query point and its nearest neighbour are in the same bin in all the k dot products. Therefore, implementation algorithms use multiple such k dot product functions $g_1, g_2, ..., g_l$ to increase the chances of success and ensure good collisions. The probability of finding the true nearest neighbour increases with l.

The above idea is generalized to a metric space (M, d) as follows. A family $H = \{h : M \longrightarrow U\}$ of hash functions from a domain M to a range U is called

$(r_1, r_2, P_1, P_2 > 0)$-sensitive or locality-sensitive if for points p, q in M

$$Pr(h(p) = h(q)) \geq P_1 \; for \; d(p,q) \leq r_1$$

$$Pr(h(p) = h(q)) \leq P_2 \; for \; d(p,q) \geq r_2 = cr_1$$

The probabilities $Pr(h(p) = h(q))$ are computed over the random choice of hash functions h from H. The argument is that if the probability of a training instance p stored in the same bucket as a test instance q is high, then p is near q.

There is no guarantee that a distance measure has a locality-sensitive family of hash functions. However, Hamming distance, L_1 norm, L_2 norm, cosine distance and Jaccard distance, all have corresponding locality-sensitive hash families defined (Indyk *et al.*[1998], Leskovec *et al.* [2010]). Euclidean distance is the metric most often used in nearest neighbour search.

2.7 Summary

There have been several efforts focussed on rāga identification. Early attempts adopted transcription based rāga recognition. Systematic analyses of motifs led to better understanding of rāga characterisation and the realization that transcriptions to svaras could lead to significant loss of rāga information. Most experiments used small datasets non-representative of the size of the rāga classification problem in hand, thus limiting the significance of the performance results achieved.

In this chapter, we propose the use of LSH to obtain similar rāgas or cohorts required for the rāga verification framework. The LSH technique uses dot products

with random vectors and enables to quickly find nearest neighbours from large databases.

CHAPTER 3

RĀGA IDENTIFICATION USING LSH

3.1 Introduction

This chapter describes the overall framework of the LSH based method proposed as solution in Section 2.5. The experimental results presented in the next chapter show that the proposed approach can, in fact, quite effectively find the exact rāga, in addition to finding its cohorts. Hence, this chapter presents the proposed method as one to realize an LSH based rāga identification system.

Rāga is an essential aspect of music in the Indian classical. There does not exist a parallel to the concept of rāga and the task of rāga identification in the Western classical music. The closest music retrieval tasks that one can associate with, are the task of cover song detection i.e. determining the same songs rendered by different musicians (Casey *et al.*[2007]; Ellis *et al.*[2007]; Serra *et al.* [2008]] and the task of Query by Humming (finding a desired song using a humming clip) (Ryynnen *et al.*[2008]; Yang *et al.* [2010]). Some of these (Casey *et al.*[2007], Ryynnen *et al.*[2008]) also use LSH for the music retrieval.

A typical MIR task uses LSH just as any other machine learning technique is generally used with MIR. Section 3.2 briefly sketches how LSH is typically used with MIR.

However, before applying any machine learning approach in music information retrieval, an important step is that of determining an appropriate feature vector for

the task and extracting the same from the music audio signal. Thus, our primary challenge now is that of determining an appropriate feature vector (to represent a rāga) that may be used with LSH to quickly and approximately compare with the songs in a large database to find the matching rāga (nearest neighbour) or similar rāgas (approximate neighbour). Section 3.3 describes the data preprocessing step to extract the feature vectors. Specifically, a T-F trajectory based feature vector is proposed to be used with LSH to enable identification of rāga in Carnatic music.

Section 3.4 describes the overall framework of the proposed LSH based method. Section 3.5 includes details of the LSH python implementation used in this work to demonstarte the proposed approach. Section 3.6 summarizes the salient points from this chapter.

3.2 MIR using LSH

Figure 3.1 illustrates the general use of LSH with a typical MIR classification task. LSH is first used to index the extracted feature vectors which store the audio seg-

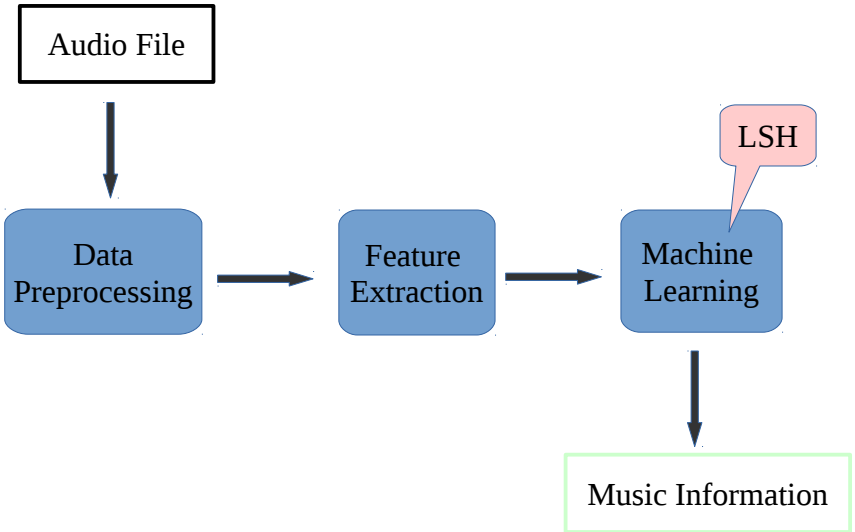


Figure 3.1: MIR using LSH

ments along with their class labels. LSH table construction is like the training phase of a machine learning task(Figure 3.2). The indexing enables efficient searching and retrieval of the matching vectors from the database.

Similar feature vectors are also extracted for the user query. The query feature vectors are now matched with the indexed feature vectors to find the right class label. LSH query is like the testing phase of a machine learning task; (Figure 3.3). For each of the query feature vectors, a ranked list of class labels is retrieved using LSH. The final classification is performed using suitable performance measures derived for the MIR problem.

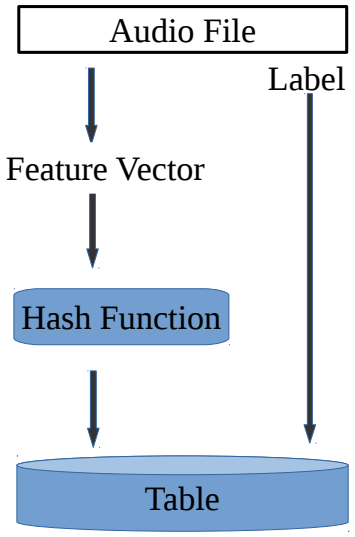


Figure 3.2: LSH Table construction

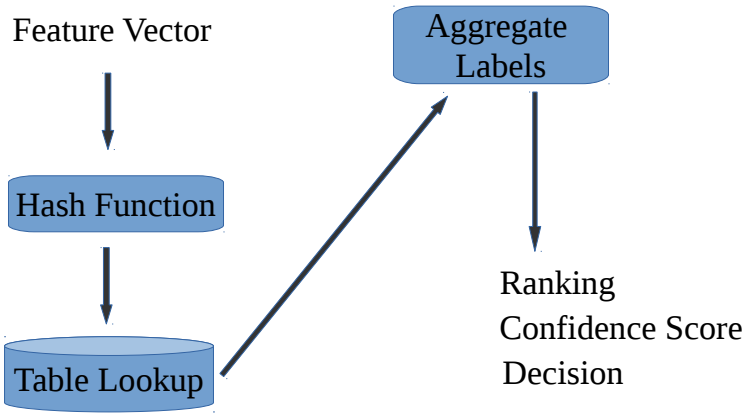


Figure 3.3: LSH Query

3.3 Data Preprocessing

Any stored recording or a query is first converted into a time-frequency trajectory or pitch contour. This task of *Predominant Melody Extraction* may be performed using any available melody transcription method (Goto and Hayamizu [1999]; Goto [2004]; Ryynanen and Klapuri [2008]; Salamon and Gmez [August 2012]; Salamon *et al.* [March 2014]).

One of the unique attributes to Indian classical music is the concept of relative pitch. The main performer is free to decide on any note as the tonic which becomes the *ādhāra shadjam* or *sā*. The pitch positions of every other svara or musical note during that rendition is derived relative to this base *sā*. Every artist has a range of about 2.5 octaves; the rendition of a rāga can span across this entire range. The choice of *sā* is therefore crucial for an artist so that the artist can exploit their voice range effortlessly. Hence the estimation of the tonic is critical to the task of rāga identification. The tonic used in a given rendition may be estimated as in Gulati *et al.* [2012] or Gulati *et al.* [March 2014]. The pitch contour of each of the database items is folded to 1 octave, normalized in pitch using the corresponding tonic estimation and converted to a uniform scale. Having obtained a tonic normalized

pitch contour, the *pitch vector*s are extracted as follows.

Rewrite a sequence of normalized pitches as a sequence of L notes $n_{1:L}$, where note $n_i =< p_i, b_i >$; b_i being the onset time(in seconds) of pitch value p_i. The offset time of the note i for $i = 1, ..., L-1$ is taken as the onset time b_{i+1} of the next note. Extract pitch vector $\mathbf{p}_i$ of length d by determining the pitch values within w-second window starting from the note onset b_i. That is, determine the pitch values on a uniform time grid $b_i + w \times j/(d-1)$, $j = 0, ..., d-1$. Do not extract any pitch vector if the window exceeds the offset of the last note, i.e., $b_i + w > e_L$. Extract pitch vectors from each database recording and index the list of $< \mathbf{p}_i, rāga >$ records.

The data pre-processing steps for the proposed approach are summarized in Figure 3.4. The pitch vector of a w-second window frame segment extracted from an audio excerpt is the basic feature used in the proposed method.

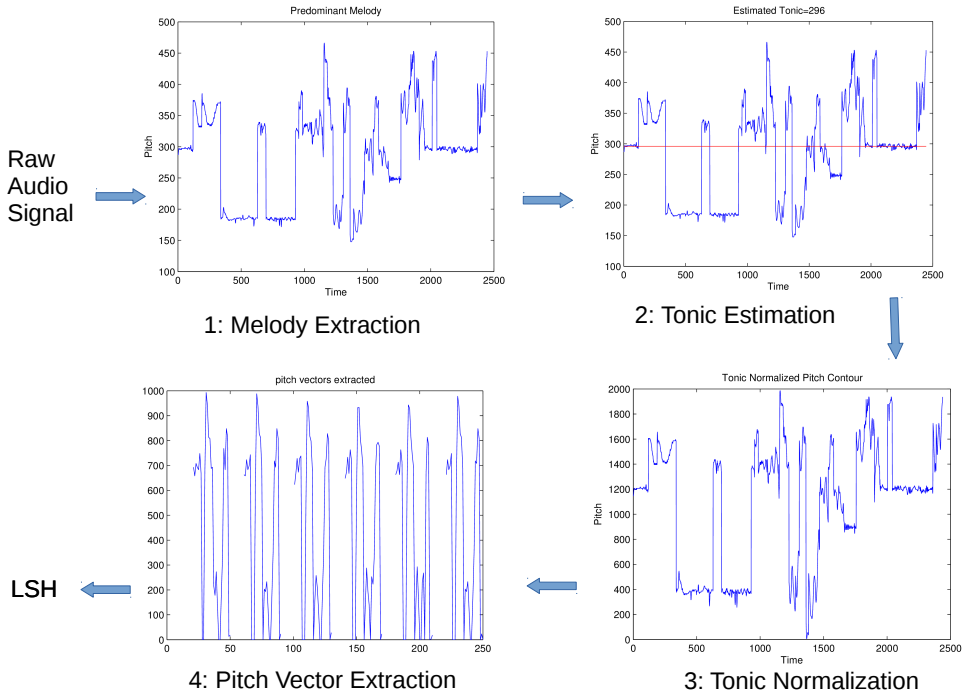


Figure 3.4: Data Preprocessing

3.4 Rāga identification using LSH

The proposed method for rāga identification may be considered as a modification of the method used by Ryynnen and Klapuri [2008] for QBH (Query by Humming). Figure 3.5 provides an overview of the method.

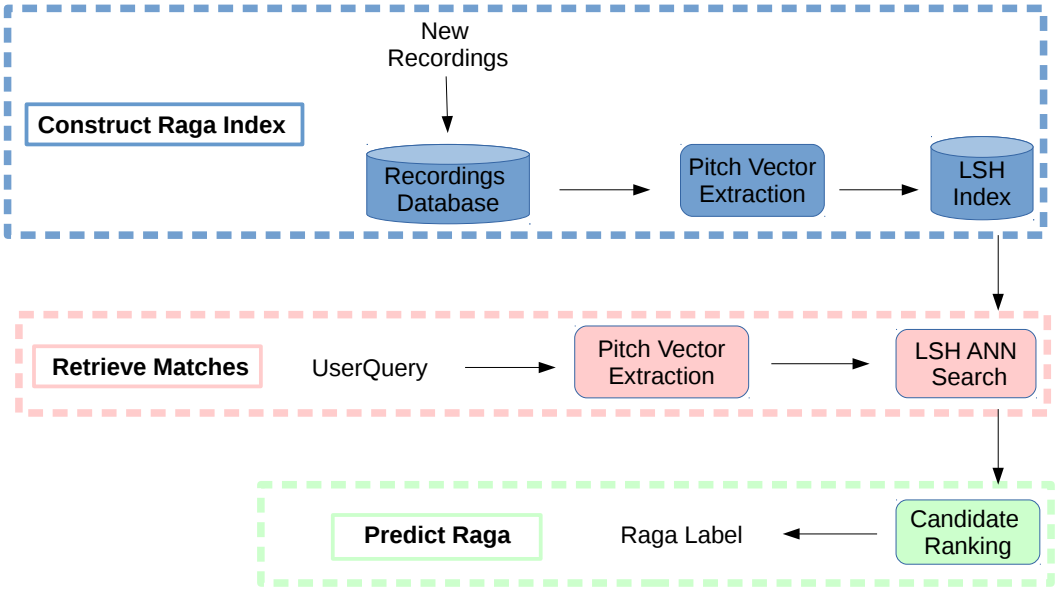


Figure 3.5: A block diagram of the proposed method

The proposed method uses LSH as in a typical MIR task, as illustrated in Figure 3.1. The pitch vectors of w-second window frame segments (as described in Section 3.4) obtained from a training set of audios along with their rāga labels are used as the templates and indexed using LSH. Each of such pitch vectors from a test set are matched against the indexed pitch vectors and queried for the top-N matches using LSH. After retrieving the top-N matches for all the pitch vectors of a given query, a list of candidate rāga labels is available for final ranking. The ranking of the rāga labels is performed by examining labels and the LSH ranks of all the

retrieved matches to arrive at the final classification of the test audio or the audio query. Different approaches as indicated in Section 4.5 were adopted to aggregate and rank the LSH query results and arrive at the label prediction.

3.5 LSH Implementation Used

There are quite a few LSH implementations available[1] publicly for use. In the current work, lshash 0.0.4 dev[2], a python implementation, has been used for demonstration. More details on the use of this package is included in the Appendix 6.1. *LSHash*[3] implements locality sensitive hashing using random projections for input vectors of given dimension.

3.6 Summary

The proposed method for rāga identification uses the pitch vectors as the basic feature. Pitch vector may be considered as a subsampling, over a uniform time scale, of the pitch contour of an audio segment. The proposed method uses LSH as any other machine learning technique. After the data pre-processing step, the pitch vectors are extracted and the LSH table is constructed with the training examples. LSH query looks up the hash table to retrieve the best matches for a given test pitch vector. Labels are aggregated to rank and make a final rāga prediction.

[1]https://www.quora.com/What-are-some-good-LSH-implementations
[2] https://pypi.python.org/pypi/lshash/0.0.4dev
[3]https://github.com/kayzh/LSHash/blob/master/lshash/lshash.py

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 Introduction

This chapter elaborates on the experiments performed to demonstrate the use of the proposed method for rāga identification in Carnatic music and potential use of the method in finding similar rāgas or cohorts. Analysis of the results obtained reveal improved performance and scalability of the proposed method.

This chapter is organized as follows. Section 4.2 details the dataset used in this work. Section 4.3 sketches the experiments performed. Section 4.4 indicates the parameters that were used with the LSH. Section 4.5 defines the proposed performance measures considered for evaluation of the classification results. Section 4.6 presents the results obtained. Section 4.7 proposes possible confidence measures that could be considered for evaluating reliability of the rāga recognitions made by the proposed method. Section 4.8 includes discussion on some interesting insights, justifications and inferences for the proposed method. This chapter concludes with Section 4.9 summarizing the strengths of the proposed method for rāga identification.

4.2 Datasets Used

The experiments were based on the audio recordings obtained from the research corpora of CompMusic (`http://compmusic.upf.edu/node/1`), a research project

funded by the European Research Council. The recordings include instrumental, vocal, live as well as studio recording excerpts. The details of the dataset collection used are as summarized in Table 4.1.

	Train	Test	Total
Number of Rāgas	182	182	
Number of Recordings	927	927	
Number of Pallavi lines	893	893	
Duration of Recordings (in minutes) from			
Initial Portions	281.7	187.8	469.5
Later Portions		123.6	123.6

Table 4.1: Dataset Used

In Table 4.1, initial portion of a recording refers to the first 30 secs of the database recording. Later portions of a recording refers to what remains after leaving out the initial 30 secs of it. *Pallavi* lines of compositions in the database recordings were obtained by first marking their start as in Sekhar *et al.* [2016] and using them for automatic segmentation of the compositions as described in Sankaran *et al.* [2015]. **DI**, **DP** and **DL** will be used to refer to the data obtained from initial portions, pallavi lines and later portions of the recording respectively. Out of the pitch vectors extracted from **DI** and **DP** for each rāga, 60% of the vectors were used as templates for indexing and remaining 40% were used for testing.

4.3 Experiments

Initial experiments were conducted to ascertain the best values for LSH parameters. The dataset of 30 rāgas used by Dutta *et al.* [2015] was extended to 40 rāgas and used for the purpose of fine tuning the LSH parameters. The later experiments were based on the best LSH configuration obtained through these initial experiments.

Experiments were performed varying the value of N, the configured number of LSH query match results. Different approaches as indicated in Section 4.5 were adopted to aggregate and rank the LSH query results and arrive at a label prediction. The prediction results were further analyzed to determine:

- the best performing approach for the problem
- appropriateness of the suggested confidence measure
- effectiveness of the proposed method in retrieving the cohorts

Experiments were also performed without folding (Section 3.4) to confirm that the folding of the pitch vectors to a single octave does not result in any major loss of information.

Errors in the results of a recognition system are always inevitable. Confidence measures are used to evaluate reliability of any recognition or prediction result. A good confidence measure can benefit a rāga recognition system in a variety of practical applications:

- smartly reject noise
- detect or reject out of vocabulary rāgas
- detect or correct potential recognition mistakes
- cleanup human labeling errors in a large training corpus
- guide the system to perform unsupervised learning

Therefore, supported by the experimental results, Section 4.7 suggests some confidence measures to evaluate the reliability of the recognitions made by the proposed method.

4.4 Choice of LSH parameters

The initial choice of parameters was based on Ryynnen and Klapuri [2008]. A window size of 3sec and a pitch vector dimension of $d = 20$ were chosen. The Table 4.2 provides a summary of the LSH parameters which were considered for experimentation, the various values that were tried and what worked best. The best performing window length of 4sec is in conformance with the results obtained by Dutta *et al.* [2015], where typical motifs were found to be of length 4sec. The choice for the distance metric was Euclidean and the best results were obtained with a hash size of 6 bits, meaning 64 bins and an input dimension of $d = 30$.

LSH Parameter	Values Tried	What worked best
window size (in sec)	2-10	4
vector dimension	10,20,30,40,50	30
metric	Euclidean	-
hash size (in bits)	4-8	6
No. of retrieved candidate matches N	1-5, 10	1

Table 4.2: Summary of experiments with LSH Parameters

4.5 Performance Measure

The candidate matches are ranked by examining all the matches retrieved, to obtain the final list of retrieved rāga matches. The following different approaches were considered for aggregating and ranking the candidate matches obtained from the LSH queries. Each of the following approaches were tried with different values of N. In the event of a tie while making a decision by majority, the rāga with the least LSH ranking is chosen.

- **Approach 1 :** Assign the rāga with the majority hits, counting a rāga hit only once for each pitch vector, though a rāga may have hit more than once among the N matches for the given pitch vector
- **Approach 2 :** Count all top-N hits for each of the pitch vectors from a test query and vote for the rāga by majority
- **Approach 3 :** Consider top few hits by their LSH ranking and vote by majority
- **Approach 4 :** Consider an empirically chosen cut-off threshold on the LSH ranking of the matches and then assign rāga by majority voting

Here is an illustration for the method of evaluation by the different approaches for $N = 5$. In the following example, we are given the LSH results for 10 pitch vectors of an audio query. For each of the 10 pitch vectors of the query, we have obtained the *(LSH ranking, raga label)* pair for the top 5 candidate matches.

0 bhairavi 1456 shri 1458 shri 1710 hemavati 1771 bhairavi
0 bhairavi 1529 hemavati 1642 karaharapriya 1702 kamas 1705 bhairavi
0 bhairavi 1398 karaharapriya 1520 shri 1683 kamas 1700 kamas
1520 karaharapriya 1644 karaharapriya 1648 hemavati 1655 bhairavi 1708 karaharapriya
1651 karaharapriya 1714 karaharapriya 1720 bhairavi 1941 kamas 1946 shri
1472 karaharapriya 1524 shri 1538 bhairavi 1811 kamas 1825 shri
1344 shri 1648 shri 1773 kamas 1774 shri 1787 kamas
1639 shri 1643 shri 1647 karaharapriya 1675 kamas 1705 karaharapriya
1341 shri 1937 kamas 1939 kamas 1953 abhogi 1975 kamas
1825 kamas 1894 shri 1952 shri 1958 kamas 1959 kamas

The rāga matches that we have obtained are:

bhairavi, shri, hemavati, karaharapriya, kamas, abhogi

Now count the occurrences of the rāgas as per the approach. For *Approach 3*, let us consider, say, the top 20% (that is, the top 10) of hits when sorted by LSH ranking. For *Approach 4*, let us consider a threshold of 1400 for the LSH ranking. Result of this step is summarized in Table 4.3 with the decision label obtained by the

corresponding approach being highlighted.

Approach	1	2	3	4
abhogi	1/10	1/50	0/10	0/6
bhairavi	6/10	8/50	3/10	**3/6**
hemavati	3/10	3/50	0/10	0/6
kamas	8/10	14/50	0/10	0/6
karaharapriya	6/10	10/50	3/10	1/6
shri	**8/10**	**14/50**	**4/10**	2/6

Table 4.3: Rāga counts obtained with the different approaches

4.6 Results

Table 4.4 summarizes the results of the experiments with various N and decision approaches. 4 sec frame segments extracted from initial portions of the recordings were used in this experiment. Randomly chosen 60% of the segments from each rāga were chosen for training and the remaining 40% were de-marked for testing. Results suggest that it may be enough to save only a couple of closest matches per each query vector to get the best performance.

N =	1	2	3	4	5
Approach 1	54.65	50.61	46.67	44.85	36.06
Approach 2	54.65	53.94	52.02	50.81	49.09
Approach 3	54.53	53.82	51.91	50.70	49.20
Approach 4	**67.74**	**67.33**	**66.43**	**66.03**	**65.63**

Table 4.4: Percentage accuracy for different approaches on 3000 rāga queries

The results in Table 4.4 were further analysed to measure the effectiveness of the proposed method in capturing the musical similarities as perceived by a human listener. Cohorts of a given rāga are those rāgas which have similar movements,

common characteristic svarās or phrases but have subtle differences with the given rāga (Dutta *et al.* [2015]). A list of cohorts were defined for each rāga and the LSH matches retrieved for each of the rāga queries were compared against this list of cohorts. Table 4.5 summarizes the results of the cohort analysis.

N =	1	2	3	4	5
Approach 1	79.52	77.77	76.22	74.93	69.77
Approach 2	79.49	79.13	78.15	77.99	76.91
Approach 3	79.52	81.07	81.26	81.09	80.4
Approach 4	**86.47**	**85.83**	**85.11**	**84.87**	**84.58**

Table 4.5: Rāga Analysis for different approaches

In Carnatic music, *Pallavi* generally refers to the first one or two lines of lyrics set to a melody which is the dominating theme of a composition. Pallavi is usually based on the elementary melodic curve of the rāga and is repeated after each segment of the composition (Nijenhuis [1974]). Dutta and A.Murthy [2014]; Dutta *et al.* [2015] use pallavi lines to obtain typical motifs of a rāga using a modified RLCS approach. Our experiments with the pallavi lines using the proposed method have achieved comparable results. The results of various experiments performed with datasets **DP**, **DI** and **DL** are summarized in Table 4.6.

Train Data	Test Data	Rāga Accuracy	Cohort Accuracy	No. of Queries
DI	**DI**	67.74%	86.47%	3000
DP	**DI**	74.86%	96.55%	3000
DI + DP	**DI**	**88.84%**	**98.12%**	3000
DI	**DP**	34.94%	75.87%	3000
DP	**DP**	71.49%	90.33%	3000
DI + DP	**DP**	**91.52%**	**98.41%**	3000
DI	**DL**	6.38%	64.63%	1260
DP	**DL**	7.49%	67.56%	1260
DI + DP	**DL**	**8.09%**	**71.04%**	1260

Table 4.6: Comparative Results with Approach 4 , N=1

The results indicate the rich content of rāga characteristics in the initial portions of a rāga exposition or a composition and the absence of the same in the later parts of the rendering. The later segments also being more rhythmically intensive, can be very noisy and more scale based. The recognition accuracies on the later segments **DL** and the corresponding cohort analysis results should be a testimony to this.

Figure 4.1 shows pitch vectors for two rāgas, Sankarābharanam and Tōdi, extracted from the initial portions of the recordings being typical motifs for the rāgas. As also illustrated in Figure 4.3, pitch vectors extracted from later segments of the of ālāpana or composition may often look quite similar. Figures 4.2, 4.4 provide a similar comparison for rāgas Tōdi and Nāta. In the Figures 4.1 - 4.4, the Tonic is mapped to 240 cents. Further, the improved results with **DI** and **DP** put

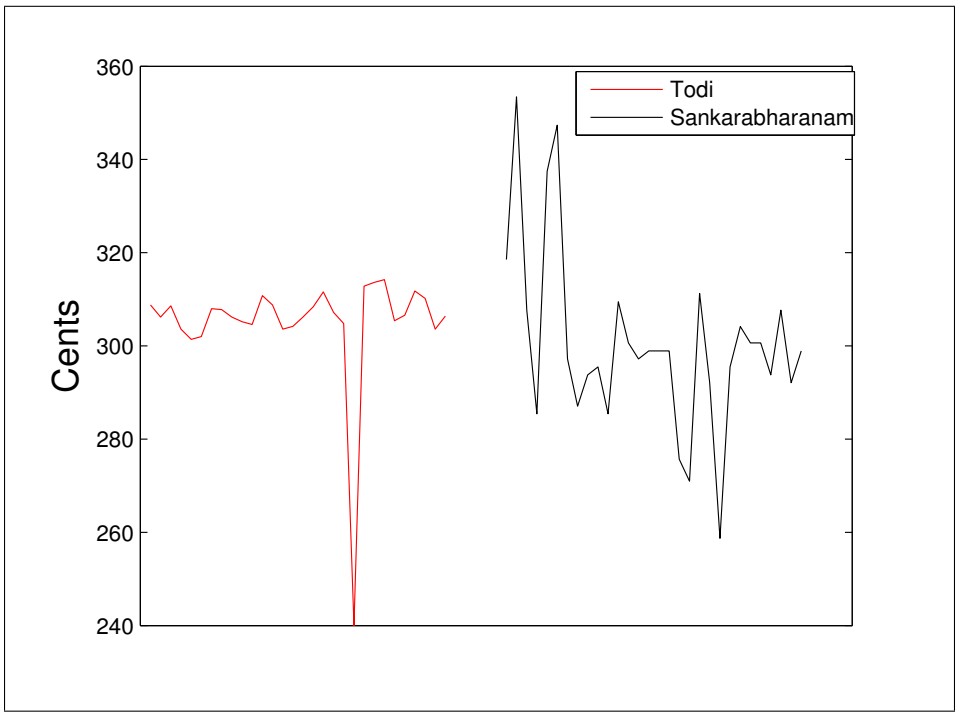


Figure 4.1: Motivic pitch vectors for Sankarābharanam and Tōdi extracted from the initial portions.

together confirm our conjecture that the pallavi lines contribute to the base set of rāga motifs and initial pitch vectors address the improvisations in performances.

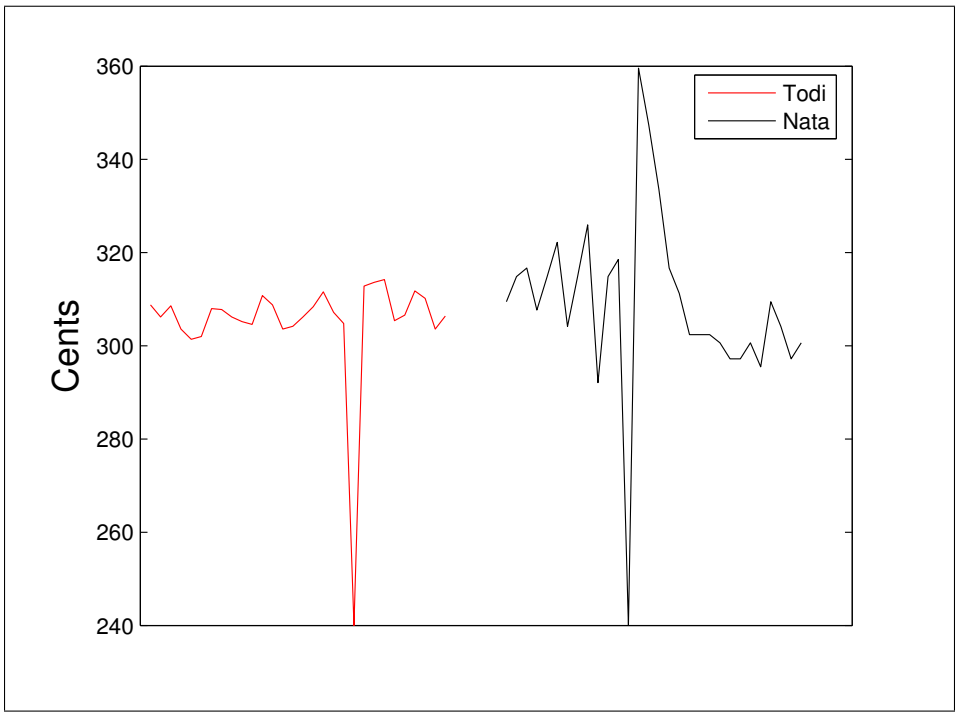


Figure 4.2: Motivic pitch vectors for Nāta and Tōdi extracted from the initial portions.

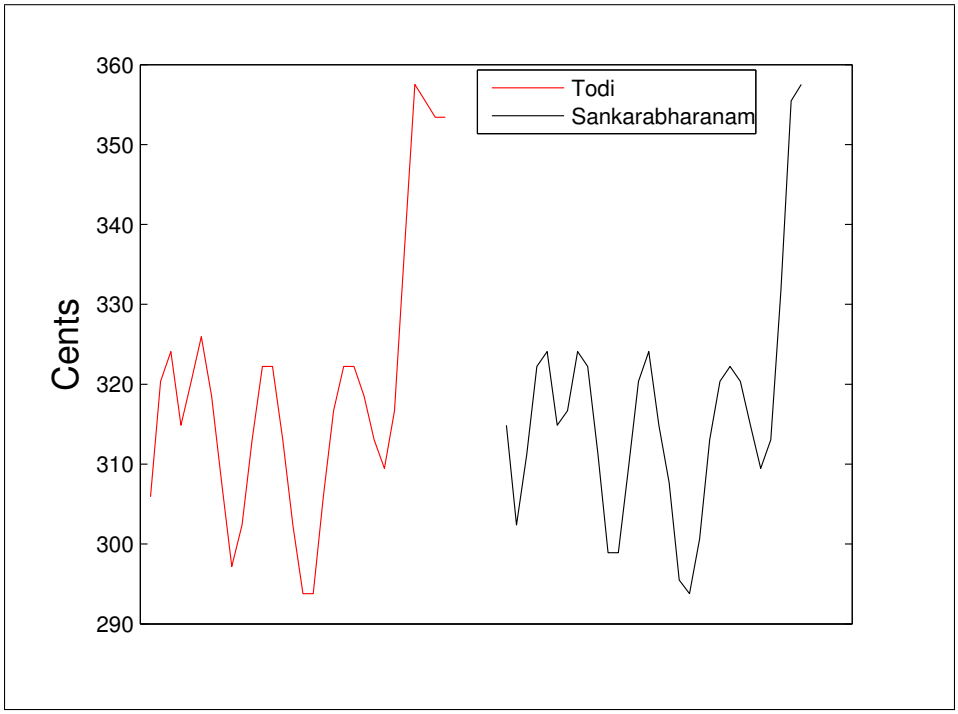


Figure 4.3: Pitch vectors of Sankarābharanam and Tōdi, extracted from later segments looking very similar.

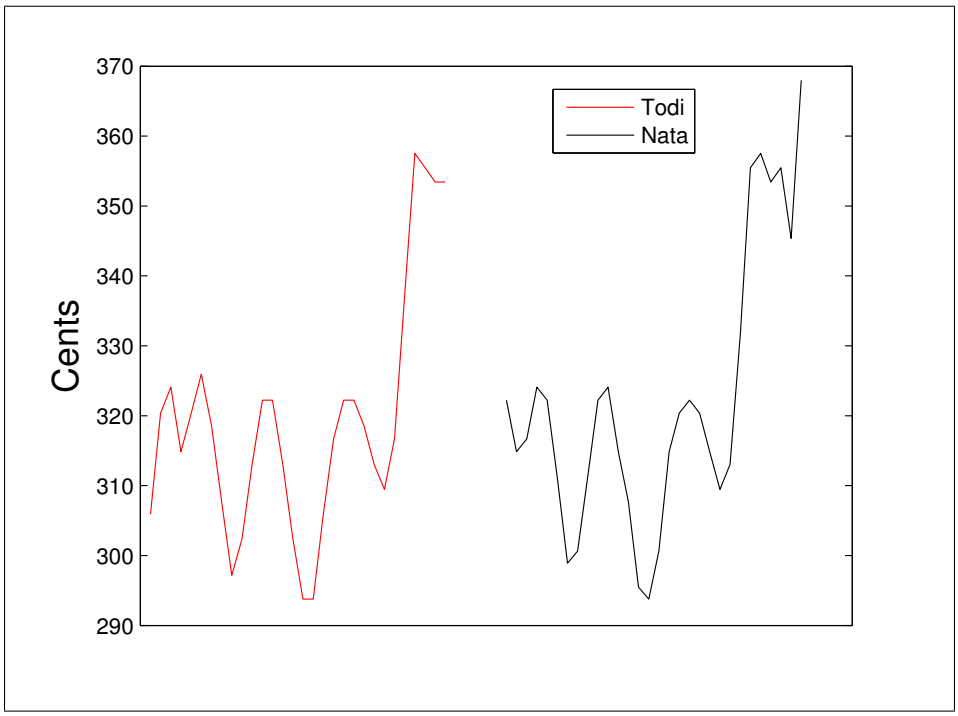


Figure 4.4: Pitch vectors of Nāta and Tōdi, extracted from later segments looking very similar.

The same experiments performed without folding to a single octave (covering about 2.5 octaves on either side of tonic) did not improve the results as is seen from Table 4.7. In fact, the performance has dropped as is expected. A rāga motif in Carnatic music generally can occur in any octave and a motif occuring in a specific octave does not become a distinguishing characteristic for a rāga in carnatic music. Therefore, folding to a single octave does not create loss of information and yields enhanced performance on rāga identification. Also it is worth noting that the difference in performance on cohort accuracy is much less than on rāga accuracy.

Train Data	Test Data	Rāga Accuracy	Cohort Accuracy	No. of Queries
DI + DP	**DI**	74.80%	91.29%	3000
DI + DP	**DP**	79.65%	93.08%	3000
DI + DP	**DL**	6.68%	60.69%	1260

Table 4.7: Comparative Results without Folding (Approach 4 , N=1)

4.7 Confidence Measure

Audio is recognized as of the predicted rāga by the top-N criteria. However, we ignore the hit-rate while making the rāga decision because it is highly variable. Clearly, a raw-decision is inadequate as a confidence measure to judge recognition reliability. However, the indicator variable, *hit-rate*, which is used in making the decision, itself can possibly serve as a good confidence measure since it represents the absolute quantitative measure of the match.

Tables 4.8 and 6.3 provide the statistics on the hit-rate for the PASSed and the FAILed tests on different datasets. $P(C|R_i)$ denotes the percentage of correctness of prediction (PASSed) in each range R_i of the hit-rate. There is a clear trend indicating greater confidence with higher hit-rate as is seen pictorially in Figures 4.5 and 6.1. Also the percentage of correctness of prediction in range R_i of the hit-rate may be seen as the posterior; hence the notation $P(C|R_i)$.

Hits Rate	Total	PASS	FAIL	$P(C\|R_i)$ %
[0,10]	0	0	0	
(10,20]	41	9	32	21.95
(20,30]	71	31	40	43.66
(30,40]	83	45	38	54.22
(40,50]	59	34	25	57.63
(50,60]	43	35	8	81.40
(60,70]	44	37	7	84.09
(70,80]	74	70	4	94.59
(80,90]	95	90	5	94.74
(90,100]	2367	2283	84	96.45
All	2877	2634	243	91.55

Table 4.8: Hit-rate stats: Testing *DP* with *DI* + *DP* for train.

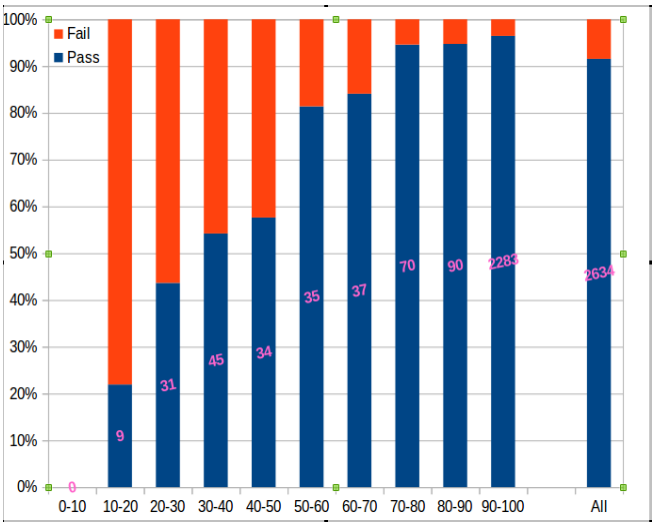


Figure 4.5: Hit-rate stats: Testing *DP* with *DI* + *DP* for train.

Another approach to measure of confidence could be to look at the likelihood ratios(LR) (Jiang [2005]). Tables 4.9 and 6.4 show LRs computed for each range of hit-rate based on tests performed on the datasets *DP* and *DI*. The ratio $\frac{P(R_k|C)}{P(R_i|W)}$ where W is the event of a wrong prediction, is the Likelihood Ratio. It is interesting to note here again a clear trend (Figures 4.6 and 6.2).The prediction may be accepted or rejected based on a threshold on the LR ratio.

Hits Rate	Total	PASS	FAIL	$P(C\|R_i)$ %	$P(R_i\|C)$	$P(R_i\|W)$	LR
[0,10]	0	0	0		0.00	0.00	
(10,20]	41	9	32	21.95	0.34	13.17	0.03
(20,30]	71	31	40	43.66	1.18	16.46	0.07
(30,40]	83	45	38	54.22	1.71	15.64	0.11
(40,50]	59	34	25	57.63	1.29	10.29	0.13
(50,60]	43	35	8	81.40	1.33	3.29	0.40
(60,70]	44	37	7	84.09	1.40	2.88	0.49
(70,80]	74	70	4	94.59	2.66	1.65	1.61
(80,90]	95	90	5	94.74	3.42	2.06	1.66
(90,100]	2367	2283	84	96.45	86.67	34.57	2.51
All	2877	2634	243	91.55			

Table 4.9: Results of Testing *DP* with *DI* + *DP* for train

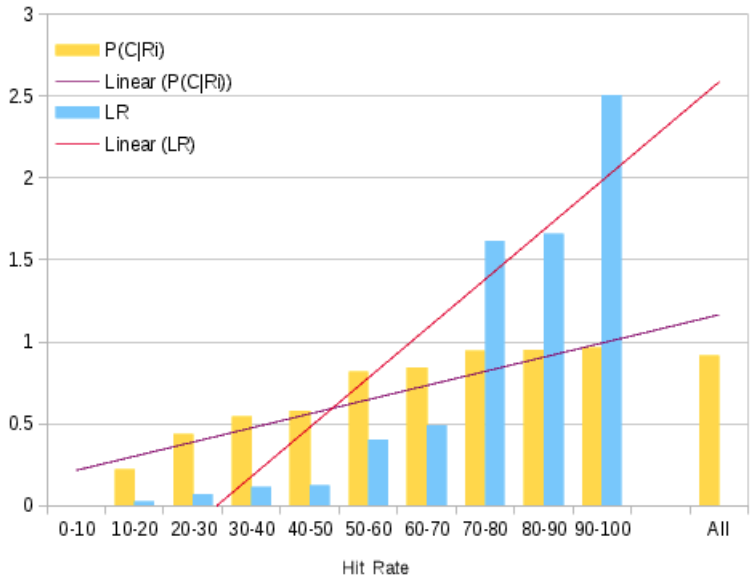


Figure 4.6: LR Trend: Testing *DP* with *DI* + *DP* for train.

4.8 Discussion

Rāga is an essential aspect of south Indian classical music, in the respect that any concert item or a musical form or composition certainly has an associated rāga element to it, the rhythm or tāl̇ā component may be optional though. Generally, a performer begins a rāga exposition (*ālāpana*) or any composition in the rāga, with a quick portrait of the rāga so as to convey its identity to the listeners and the accompanists before s/he dwells into a detailed elaboration. In other words, an artist initially defines the framework where the identity of the rāga is first established and further explorations are performed within the framework of the given context. The proposed method exploits this while considering the first few seconds of an ālāpana or composition.

Phrases are melodic motifs that collectively give a raga its asthetic form (Krishna

[2013]). A listener approaches the task of rāga identification primarily by matching musical phrases to already known familiar phrases or motifs and seldom resorts to actually recognizing or transcribing to the svarās or notes or their sequences appearing therein. Also rāga identification is a difficult task even for human listeners, and it takes years for them to acquire the skills for a large corpus. Owing to the improvisational nature of Indian music, the challenge in the phrase or motif matching approach is the intra motif variabilities as mentioned earlier in Chapter 2, context dependency and no possible precise adequate definitions for ”phrases”. Figures 6.3 provide example of pitch contours of 4 different pallavi lines from Tōdi. These make the task of rāga identification difficult also for machine learning. The success of the proposed method in rāga identification may be attributed to its closeness to the human approach to identifying rāgas.

It is also important to note that the choice of 4secs duration for the training templates clearly match the observations of Dutta and A.Murthy [2014]. The observations (refer to Tables 3 & 4 of Dutta and A.Murthy [2014]) on the average durations of similar patterns found across *one-liners* and typical motifs of a rāga well support our choice of 4sec for the frame segment duration in the proposed method. The LSH parameter value of 30 for the pitch vector dimension may also be related to the number of svarās or notes rendered in 4secs with a medium tempo by a performing vocalist.

Further, the proposed method performs the best with LSH hash size of 6 bits (=64) suggesting an interesting interpretation of the LSH hashing. A maximum of 72 seven-note scales may be formed as theoretical combinations based on twelve semitones in an octave (Ayyangar [1972]). A rāgas in practice may or may not contain ALL the seven svarās in it. Note that the hash size of 64 is (as a power of

2) the closest to this number 72.

4.9 Summary

Finally a precis on the strengths of the proposed method:

1. The method is easily scalable to a number of rāgas and repertoires. The scaling may be as simple as adding a composition in a new rāga to the database, extracting audio segments in the desired form and having them indexed.

2. No assumptions have been made with respect to the musical form of the audio snippets. The query and the training snippets may be a portion of any form[1] of musical composition which embraces the basic elements of rāga like *ālāpana*, *viruttam*, *varnam*, *kriti*, etc... (Ayyangar [1972]).

3. There are about 150-200 rāgas that are generally performed in Carnatic music concerts and the identity of a rāga is ascertained by a listener within the first few seconds of listening to an item. The proposed method works with just few seconds of the initial portions of the ālāpanas or compositions and in this respect, we are probably getting more close to human way of performing the task.

[1] https://en.wikipedia.org/wiki/Carnatic_music

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter provides the summary of the current work, some criticism and discusses the future scope of the work.

5.1 Summary

There have been prior efforts focussed on rāga recognition using computational and machine learning approaches. However, most of them were demonstrated to perform well only on a much limited non-representative dataset and suffered from scalablity issues. The current work addresses both these shortcomings of the earlier attempts.

The current work proposes a rāga identification method based on LSH. The method uses the pitch contour as the basic feature for the rāga recognition. The proposed method achieves better performance than that of the state-of-the-art. The results obtained are quite significant, having been obtained on a good representative sample in terms of rāgas and repertoires. The scalability is another hallmark feature of the proposed method.

The results of the experiments with LSH have been promising even with the use of a simple euclidean metric. Experiments show that it may be enough to save only a couple of closest matches per each query vector to get the best performance. LSH has been clearly successful in retrieving the cohorts, if not the rāga itself.

Interestingly, the best performing LSH parameters seem to relate surprisingly quite well to the results of previous experiments and facts from theory of music and current day practice. The proposed method works with just few seconds of the initial portions of the ālāpanas or compositions. In this respect, we are probably getting more close to human way of performing the task.

5.2 Criticism

Currently we are indexing all the pitch vectors extracted from the training audio segments and performing LSH. This does not allow us to take benefit of the efficiency of LSH. Consider down sizing the training set by automatic detection and use of only the musical segments bounded by 2 stationary points on the T-F trajectory.

The tonic normalization step is critical to rāga identification. The proposed method currently relies on automatic identification of tonic of a musical recording. The manual verification has revealed a good number of errors here. Correcting the tonics and training the system could also help improve performance significantly.

The current work has been carried out taking along a good percentage of labeling errors in the dataset. Correcting the labels could possibly improve the performance further.

5.3 Future Work

Most distance measures that have worked well in music retrieval do not satisfy the nice properties of a metric. On the other hand, many of the indexing methods rely

on the properties of a metric. Typke and Walczak-Typke [2010] discuss indexing techniques for non-metric distance measures of musical interest. Inspired by LSH, it may be worthwhile to explore the possible use of non-metric dissimilarity measures suited for music retrieval with other indexing methods.

Designing a good effective confidence measure to support rāga identification results will be of great importance while building a real-time rāga recognition system.

CHAPTER 6

Appendix

6.1 lshash0.0.4dev package

lshash 0.0.4dev is a fast Python implementation of locality sensitive hashing. Installing *lshash 0.0.4dev* package can be easily done using the pip utility. The dependencies required for our use (with euclidean distance as metric) was only *Numpy*(*bitarray* may be required if hamming distance is used.). Further details and quick start steps for lshash 0.0.4dev can be found at `https://pypi.python.org/pypi/lshash/0.0.4dev`.

6.2 Result of Testing *DI* without folding

Hits Rate r	Total	PASS	FAIL	$P(\text{PASS}\|r)$	
$0 \leq r \leq 10$	1	0	1	0.00	
$10 < r \leq 20$	33	4	29	12.12	26.98
$20 < r \leq 30$	30	13	17	43.33	
$30 < r \leq 40$	42	21	25	45.65	56.52
$40 < r \leq 50$	46	31	15	67.39	
$50 < r \leq 60$	20	21	3	87.50	81.13
$60 < r \leq 70$	29	22	7	75.86	
$70 < r \leq 80$	28	25	3	89.29	94.19
$80 < r \leq 90$	58	56	2	96.55	
$90 < r \leq 100$	1245	1176	69	94.46	
All	1540	1369	171	88.90	

Table 6.1: Results of Testing *DI* with *DI+DP* without folding

6.3 Results of Testing *DP* without folding

Hits Rate r	Total	PASS	FAIL	$P(\text{PASS}\|r)$	
$0 \leq r \leq 10$	1	0	1	0.00	
$10 < r \leq 20$	31	9	22	29.03	26.15
$20 < r \leq 30$	34	8	26	23.53	
$30 < r \leq 40$	42	19	23	45.24	45.35
$40 < r \leq 50$	44	20	24	45.45	
$50 < r \leq 60$	20	14	6	70.00	66.67
$60 < r \leq 70$	22	14	8	63.64	
$70 < r \leq 80$	15	14	1	93.33	84.62
$80 < r \leq 90$	24	19	5	79.17	
$90 < r \leq 100$	382	338	44	88.48	
All	615	455	160	73.98	

Table 6.2: Results of Testing *DP* with *DI*+*DP* without folding

6.4 Results of Testing *DI* with Folding

Hits Rate	Total	PASS	FAIL	$P(C\|R_i)$ %
[0,10]	1	0	1	0.00
(10,20]	33	4	29	12.12
(20,30]	30	13	17	43.33
(30,40]	46	21	25	45.65
(40,50]	46	31	15	67.39
(50,60]	24	21	3	87.50
(60,70]	29	22	7	75.86
(70,80]	28	25	3	89.29
(80,90]	58	56	2	96.55
(90,100]	1245	1176	69	94.46
All	1540	1369	171	88.90

Table 6.3: Hit-rate stats: Testing *DI* with *DI* + *DP* for train.

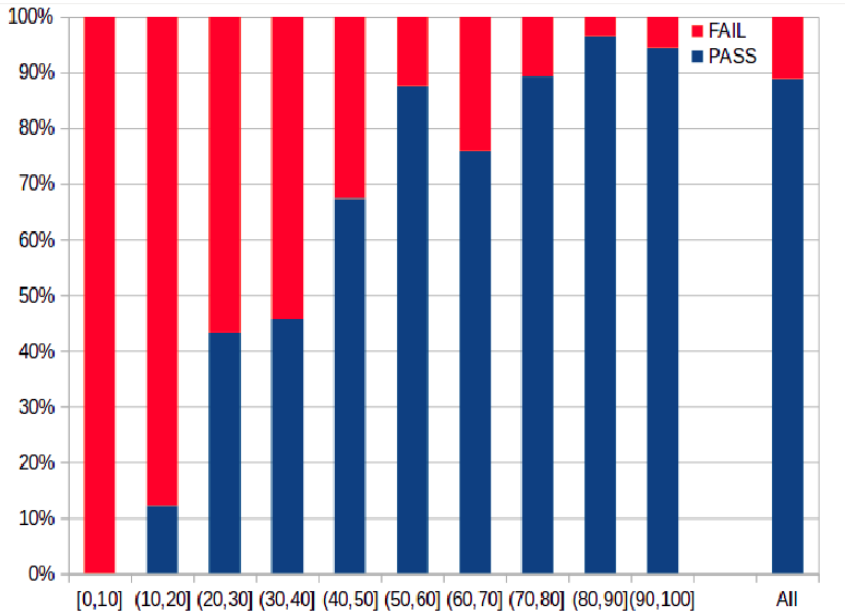


Figure 6.1: Hit-rate stats: Testing *DI* with *DI* + *DP* for train.

Hits Rate	Total	PASS	FAIL	$P(C\|R_i)$ %	$P(R_i\|C)$	$P(R_i\|W)$	LR
[0,10]	1	0	1	0.00	0.00	0.58	0.00
(10,20]	33	4	29	12.12	0.29	16.96	0.02
(20,30]	30	13	17	43.33	0.95	9.94	0.10
(30,40]	46	21	25	45.65	1.53	14.62	0.10
(40,50]	46	31	15	67.39	2.26	8.77	0.26
(50,60]	24	21	3	87.50	1.53	1.75	0.87
(60,70]	29	22	7	75.86	1.61	4.09	0.39
(70,80]	28	25	3	89.29	1.83	1.75	1.05
(80,90]	58	56	2	96.55	4.09	1.17	3.50
(90,100]	1245	1176	69	94.46	85.90	40.35	2.13
All	1540	1369	171	88.90			

Table 6.4: LR Trend: Results of Testing *DI* with *DI* + *DP* for train.

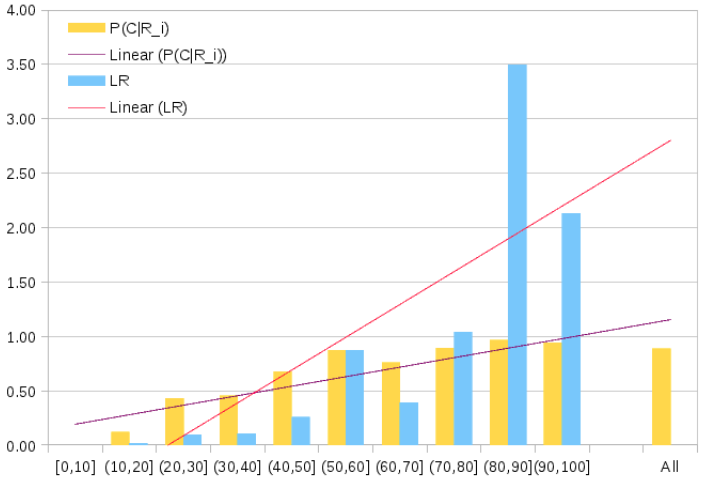
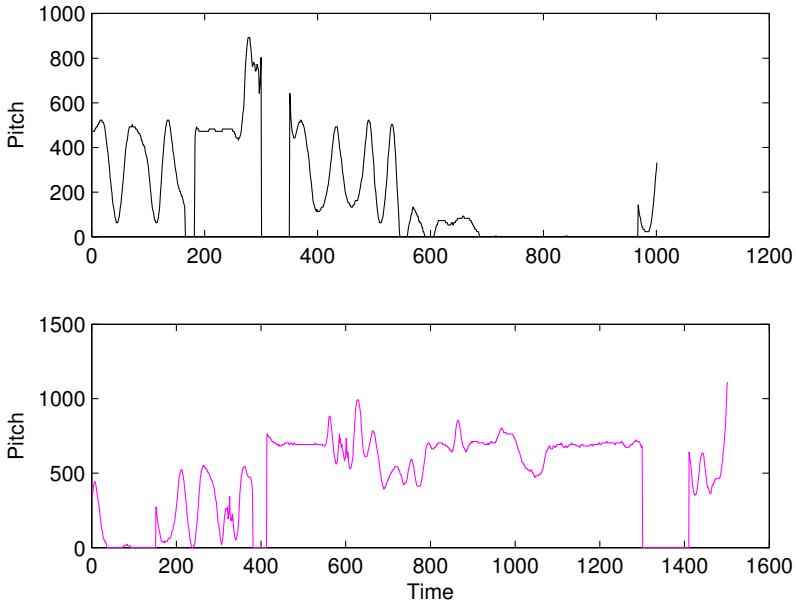
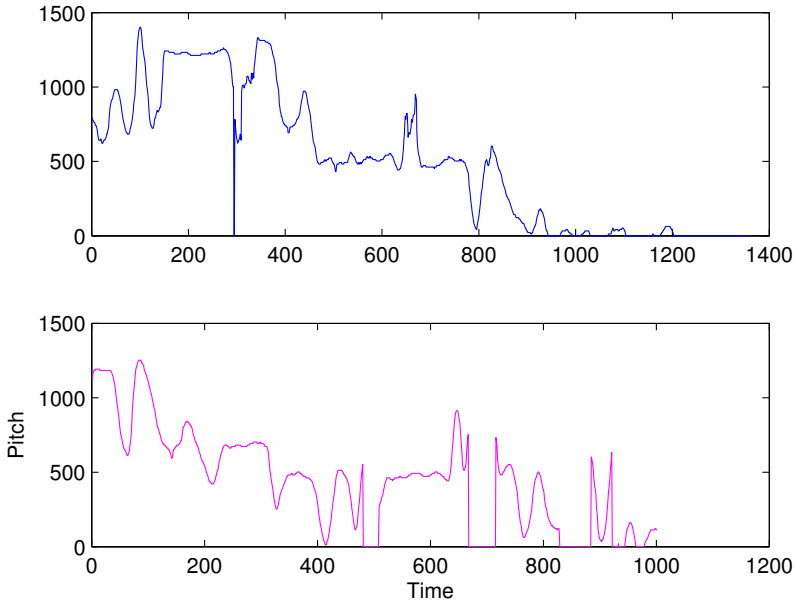


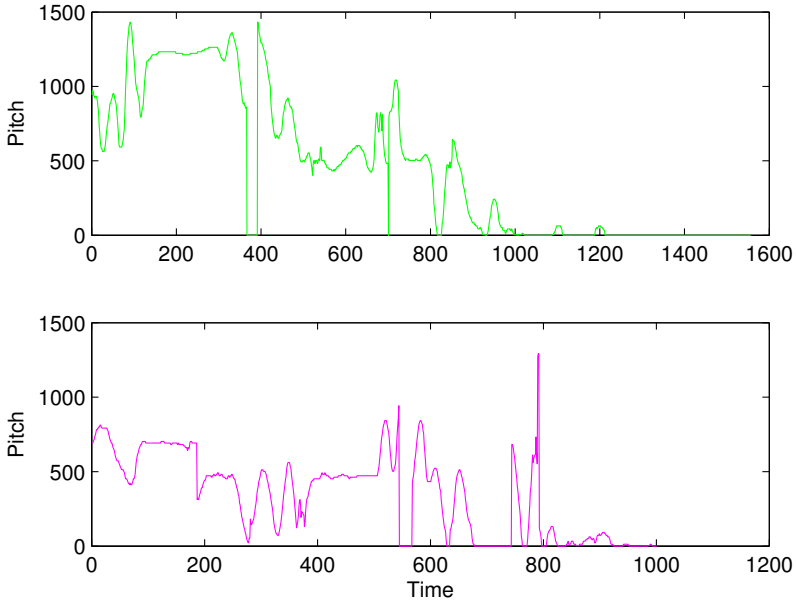
Figure 6.2: LR Trend: Testing *DI* with *DI* + *DP* for train.

6.5 Illustrations

Figure 6.3: Pitch Contours of different pallavi lines of same rāga




1500
1000
500
0
0
200
400
600
800
1000
1200
1400
Pitch
1500
1000
500
0
0
200
400
600
800
1000
1200
Time


Pitch
1500
1000
500
0
0
200
400
600
800
1000
1200
1400
1600
Pitch
1500
1000
500
0
0
200
400
600
800
1000
1200
Time

PUBLICATIONS

1. Padmasundari and Hema A. Murthy (2016). Raga Identification Using LSH. *submitted to ISMIR 2016*.

REFERENCES

Arthi, S., **H. Ranjani**, and **T. Sreenivas**, Shadja, swara identification and raga verification in alapana using stochastic models. *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2011.

Ayyangar, R. R., *History of South Indian (Carnatic) Music: from vedic times to the present*. Vipanc(h)I Cultural trust, Bombay, 1972.

Bhagyalekshmi, D., *Ragas in Carnatic Music*. CBH Publications, Nagercoil India, 1990.

Cano, P., **E. Batlle**, **T. Kalker**, and **J. Haitsma**, A review of algorithms for audio fingerprinting. *In Proceedings of International Workshop on Multimedia Signal Processing*. 2002.

Dighe, P., **P. Agarwal**, **H. Karnick**, **S. Thota**, and **B. Raj**, Scale independent raga identification using chromagram patterns and swara based features. *In IEEE International Conference on Multimedia and Expo Workshops*. 2013*a*.

Dighe, P., **H. Karnick**, and **B. Raj**, Swara histogram based structural analysis and identification of indian classical ragas. *In Proceedings of 14th International Society for Music Information Retrieval Conference*. 2013*b*.

Dutta, S. and **H. A.Murthy**, Discovering typical motifs of a raga from one-liners of songs in carnatic music. *In Proceedings of the 15th International Society for Music Information Retrieval Conference*. 2014.

Dutta, S., **P. K. Sekhar**, and **H. A.Murthy**, Raga verification in carnatic music using longest common segment set. *In Proceedings of the 16th International Society for Music Information Retrieval Conference*. 2015.

Goto, M. (2004). A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, **43**, 311–329.

Goto, M. and **S. Hayamizu**, A real-time music scene description system: Detecting melody and bass lines in audio signals. *In Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis*. 1999.

Gulati, S., **A. Bellur**, **J. Salamon**, **H. G. Ranjani**, **V. Ishwar**, **H. A. Murthy**, and **X.Serra** (March 2014). Automatic tonic identification in indian art music: Approaches and evaluation. *Journal of New Music Research*, **43**(1), 53–71.

Gulati, S., **J. Salamon**, and **X. Serra**, A two-stage approach for tonic identification in indian art music. *In Proceedings of 2nd CompMusic Workshop*. 2012.

Gulati, S., **J. Serra**, and **X. Serra**, An evaluation of methodologies for melodic similarity in audio recordings of indian art music. *In Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015.

Ishwar, V., **A. Bellur**, and **H. A.Murthy**, Motivic analysis and its relevance to raga identification in carnatic music. *In Proceedings of 2nd CompMusic Workshop*. 2012.

Ishwar, V., **S. Dutta**, **A. Bellur**, and **H. A.Murthy**, Motif spotting in an alapana in carnatic music. *In Proceedings of the 14th International Society for Music Information Retrieval Conference*. 2013.

Jiang, H. (2005). Confidence measures for speech recognition: a survey. *Speech Communication*, **45**(4), 455–470.

Krishna, A., **P. Rajkumar**, **K. Saishankar**, and **M. John**, Identification of carnatic raagas using hidden markov models. *In IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2011.

Krishna, T., *A Southern Music: The Karnatik Story*. HarperCollins Publishers India, Noida UttarPradesh, 2013.

Kumar, V., **H. Pandya**, and **C. Jawahar**, Identifying ragas in indian music. *In IEEE 22nd International Conference on Pattern Recognition (ICPR)*. 2014.

Leskovec, J., **A. Rajaraman**, and **J. Ullman**, *Mining of Massive Datasets*. Stanford University, http://infolab.stanford.edu/ ullman/mmds, http://www.mmds.org, 2010.

Lin, H.-J., **H.-H. Wu**, and **C.-W. Wang** (2011). Music matching based on rough longest common subsequence. *Journal of Information Science and Engineering*, 95–110.

Miron, M., **J. Serra**, **G. K. Koduri**, and **X. Serra**, Tuning of sung indian classical music. *In Proceedings of the 12th International Society for Music Information Retrieval Conference*. 2011.

Nijenhuis, E. T., *Indian Music: History and structure*. Tuta sub Aegide Pallas, Leiden/Koln, E.J.Brill, 1974.

Pandey, G., **C. Mishra**, and **P. Ipe**, Tansen:a system for automatic raga identification. *In Proceedings of the 1st Indian International Conference on Artificial Intelligence*. 2003.

Rao, P., **J. C. Ross**, **K. Ganguli**, **V. Pandit**, **V. Ishwar**, **A. Bellur**, and **H. A.Murthy** (2014). Melodic motivic analysis of indian music. *Journal of New Music Research*, **43**(1), 115–131.

Rao, T. S., *Studies in Indian Music*. 1930.

Ross, J. C., **T. Vinutha**, and **P. Rao**, Detecting melodic motifs from audio for hindustani classical music. *In Proceedings of the 13th International Society for Music Information Retrieval Conference*. 2012.

Ryynanen, M. and **A. Klapuri** (2008). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, **32**(3), 72–86.

Ryynnen, M. and **A. Klapuri**, Query by humming of midi and audio using locality sensitive hashing. *In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2008.

Sahasrabuddhe, H., Searching for a common language of raga. *In Proceedings Indian Music and Computers: Can Mindware and Software Meet?*. 1994.

Salamon, J. and **E. Gmez** (August 2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, **20**(6), 1759–1770.

Salamon, J., **E. Gmez**, **D. P. W. Ellis**, and **G. Richard** (March 2014). Melody extraction from polyphonic music signals: Approaches, applications and challenges. *IEEE Signal Processing Magazine*, **31**(2), 118–134.

Sankaran, S., **P. K. Sekhar**, and **H. A.Murthy**, Automatic segmentation of composition in carnatic music using time-frequency cfcc templates. *In Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2015.

Sekhar, P. K., **S. Sankaran**, and **H. A.Murthy**, Segmentation of carnatic music items using kl2,gmm and cfb energy feature. *In Proceedings of Twenty Second National Conference on Communications (NCC)*. 2016.

Serra, J., **E. Gomez**, **P. Herrera**, and **X. Serra** (2008). Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, **16**(6), 1138–1151.

Slaney, M. and **M. Casey** (March 2008). Locality-sensitive hashing for finding nearest neighbors. *IEEE Signal Processing Magazine*, 128–131.

Typke, R. and **A. Walczak-Typke** (2010). Indexing techniques for non-metric music dissimilarity measures. *Studies in Computational Intelligence*, **274**, 3–17.

Yang, J., **J. Liu**, and **W.-Q. Zhang** (2010). Fast query by humming system based on notes. *INTERSPEECH 2010*, 28982901.