

Computational Approaches for Melodic Description in Indian Art Music Corpora

Sankalp Gulati

TESI DOCTORAL UPF / 2016

Thesis Director:

Dr. Xavier Serra Casals
Music Technology Group
Dept. of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain

Dissertation submitted to the Department of Information and Communication Technologies of Universitat Pompeu Fabra in partial fulfillment of the requirements for the degree of

DOCTOR PER LA UNIVERSITAT POMPEU FABRA

Copyright © 2016 by Sankalp Gulati

Licensed under [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0](#)



The doctoral defense was held on at the Universitat Pompeu Fabra
and scored as

Dr. Xavier Serra Casals

(Thesis Supervisor)

Universitat Pompeu Fabra (UPF), Barcelona, Spain

Dr. Juan Pablo Bello

(Thesis Committee Member)

New York University (NYU), New York, USA

Dr. Emilia Gómez

(Thesis Committee Member)

Universitat Pompeu Fabra (UPF), Barcelona, Spain

Dr. Barış Bozkurt

(Thesis Committee Member)

Koç University, Istanbul, Turkey

To Papaji and Mi

This thesis has been carried out at the Music Technology Group (MTG) of Universitat Pompeu Fabra in Barcelona, Spain, from Oct. 2012 to Sep. 2016. It is supervised by Dr. Xavier Serra Casals. Work in Chapter 4, 5 and 6 has been conducted in close collaboration with Dr. Joan Serrà Julià.

Work in several parts of this thesis have been carried out in collaboration with the CompMusic team at the MTG, and the partner institutes lead by Dr. Preeti Rao (Indian Institute of Technology Bombay, Mumbai, India) and Dr. Hema A. Murthy (Indian Institute of Technology Madras, Chennai, India). A detailed list of collaborators include (alphabetically ordered) Ajay Srinivasamurthy, Ashwin Bellur, Justin Salamon, Kaustuv K. Ganguli, Ranjani H. G., Sertan Şentürk and Vignesh Ishwar. For all the music related aspects in our work pertaining to Hindustani music and Carnatic music Kaustuv K. Ganguli and Vignesh Ishwar have been consulted.

Our work has been supported by the Department of Information and Communication Technologies (DTIC) PhD fellowship (2012-16), Universitat Pompeu Fabra, and the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

Acknowledgments

When I embarked on this journey of pursuing a PhD, I thought of it as solely an intellectual pursuit, but very soon I realized that it is a way of life. A life that gets enriched by the people we engage with, and I am extremely grateful to many people for making this experience a memorable one. My heartfelt thanks to my thesis supervisor Prof Xavier Serra, whose vision of a world that nurtures different cultures led to the genesis of the CompMusic project. I thank him for giving me an opportunity to work with immense creative freedom, as well as his supervision and support throughout the dissertation. His ability to never lose sight of the big picture combined with his eye for detail, his leadership-style and magnanimity are life-lessons that shall stay with me in all my future pursuits.

I thank Joan Serrà for his constant guidance and his tireless strive for perfection, some of which, I hope, I imbibed in the process. His ideas and feedback have been instrumental in shaping my work at every step, as well as my outlook toward research work in general. His friendship has encouraged and inspired me in the good and the not-so-good times.

I would like to thank Prof Preeti Rao for her guidance during my initial years in the field, and Prof Hema Murthy for her collaboration and support. I would also like to thank Emilia Gómez and Perfecto Herrera for their valuable inputs at crucial junctures.

I thank Cristina Garrido, Sonia Espí, Alba B Rosado, Vanessa Jimenez, Lydia García, Marcel Xandrí and Jana Safrankova for always being ready to support me with a smile and helping me untangle the infinite web that bureaucracy could be.

I gratefully acknowledge Joe Cheri Ross, Vinuta Prasad, Shrey Dutta, Ashwin Bellur, and Ranjani HG for their collaboration and valuable inputs. I would also like to thank Vignesh Ishwar and Kaustuv K. Ganguli, my friends and gurus in Carnatic and Hindustani music, respectively, for not only collaborating with me in my work given their expertise in music but also for bringing the magic of Carnatic and Hindustani music to my life here in Barcelona through their wonderful performances both on and off stage.

I am extremely grateful to the friendships that were forged during this time in the MTG. I thank Ajay Srinivasamurthy for being an epitome of generosity, Marius Miron, for being such an amazing person and all his help and support, Sertan Şentürk and Gopala K. Koduri for being awesome conversationalists and co-workers, Rong Gong whose indefatigable spirit is a thing to emulate, Rafael Caro Repetto whose patience and calmness could make Yogis introspect, Georgi Dzhambazov (still can't pronounce his last name) for his insatiable curiosity, Swapnil Gupta for being my mu-

sic partner and stepping in to be my room-mate and helping me out, Alastair Porter for knowing everything about everything in technology and Andres Ferraro for helping me out and always being there for any technical support. This journey has been made further memorable by my colleagues Justin Salamon, Mohamed Sordo, and Frederic Font whose comments and suggestions have helped me look at my work from new perspectives.

I thank Jose Zapata, Pauli Be, Marius Miron and Ajay Srinivasamurthy, my wonderful flat-mates, who gave me a home away from home. My friends, Varun Jewalikar, Sergio Giraldo, Hector Parra, Dara Dabiri, Aluizio Oliveira, Nadine Kroher, Juanjo Bosch, and Sergio Oramas for all the fun-times and memories that we made during the last four years and for the sheer brilliance of their beings. My special thanks to Eva Arrizabalaga, whose kindness makes this world a little better everyday.

Last but not the least, I would like to thank my parents for always believing in me and being a constant source of support and strength. My siblings, Vikram and Ruchi for the joy they bring to my life, and my life-partner Shefali, for always being there for me...

Abstract

Automatically describing contents of recorded music is crucial for interacting with large volumes of audio recordings, and for developing novel tools to facilitate music pedagogy. Melody is a fundamental facet in most music traditions and, therefore, is an indispensable component in such description. In this thesis, we develop computational approaches for analyzing high-level melodic aspects of music performances in Indian art music (**IAM**), with which we can describe and interlink large amounts of audio recordings. With its complex melodic framework and well-grounded theory, the description of **IAM** melody beyond pitch contours offers a very interesting and challenging research topic. We analyze melodies within their tonal context, identify melodic patterns, compare them both within and across music pieces, and finally, characterize the specific melodic context of **IAM**, the *rāgas*. All these analyses are done using data-driven methodologies on sizable curated music corpora. Our work paves the way for addressing several interesting research problems in the field of **music information research**, as well as developing novel applications in the context of music discovery and music pedagogy.

The thesis starts by compiling and structuring largest to date music corpora of the two **IAM** traditions, Hindustani and Carnatic music, comprising quality audio recordings and the associated metadata. From them we extract the predominant pitch and normalize by the tonic context. An important element to describe melodies is the identification of the meaningful temporal units, for which we propose to detect occurrences of *nyās svaras* in Hindustani music, a landmark that demarcates musically salient melodic patterns.

Utilizing these melodic features, we extract musically relevant recurring melodic patterns. These patterns are the building blocks of melodic structures in both improvisation and composition. Thus, they are fundamental to the description of audio collections in **IAM**. We propose an unsupervised approach that employs time-series analysis tools to discover melodic patterns in sizable music collections. We first carry out an in-depth supervised analysis of melodic similarity, which is a critical component in pattern discovery. We then improve upon the best possible competing approach by exploiting peculiar melodic characteristics in **IAM**. To identify musically meaningful patterns, we exploit the relationships between the discovered patterns by performing a network analysis. Extensive listening tests by professional musicians reveal that the discovered melodic patterns are musically interesting and significant.

Finally, we utilize our results for recognizing *rāgas* in recorded performances of **IAM**. We propose two novel approaches that jointly capture the tonal and the temporal aspects of melody. Our first approach uses melodic patterns, the most prominent cues

for *rāga* identification by humans. We utilize the discovered melodic patterns and employ topic modeling techniques, wherein we regard a *rāga* rendition similar to a textual description of a topic. In our second approach, we propose the *time delayed melodic surface*, a novel feature based on delay coordinates that captures the melodic outline of a *rāga*. With these approaches we demonstrate unprecedented accuracies in *rāga* recognition on the largest datasets ever used for this task. Although our approach is guided by the characteristics of melodies in **IAM** and the task at hand, we believe our methodology can be easily extended to other melody dominant music traditions.

Overall, we have built novel computational methods for analyzing several melodic aspects of recorded performances in **IAM**, with which we describe and interlink large amounts of music recordings. In this process we have developed several tools and compiled data that can be used for a number of computational studies in **IAM**, specifically in characterization of *rāgas*, compositions and artists. The technologies resulted from this research work are a part of several applications developed within the CompMusic project for a better description, enhanced listening experience, and pedagogy in **IAM**.

Resum

La descripció automàtica d'enregistraments musicals és crucial per interactuar amb grans volums de dades i per al desenvolupament de noves eines per a la pedagogia musical. La melodia és una faceta fonamental en la majoria de les tradicions musicals i, per tant, és un component indispensable per a la descripció automàtica d'enregistraments musicals. En aquesta tesi desenvolupem sistemes computacionals per analitzar aspectes melòdics d'alt nivell presents en la música clàssica de l'Índia (MCI), a partir dels quals descrivim i interconnectem grans quantitats d'enregistraments d'àudio. La descripció de melodies en la MCI, complexes i amb una base teòrica ben fonamentada, va més enllà de l'anàlisi estàndard de contorns de to (“pitch” en anglès), i, per tant, és un tema de recerca molt interessant i tot un repte. Analitzem les melodies dins del seu context tonal, identifiquem patrons melòdics, els comparem tant amb ells mateixos com amb altres enregistraments, i, finalment, caracteritzem el context melòdic específic de la música IAM: els *rāgas*. Tots els anàlisis s'han realitzat utilitzant metodologies basades en dades, amb un corpus musical de mida considerable.

Iniciem la tesi recopilant la col·lecció més gran de MCI obtinguda fins al moment. Aquesta col·lecció comprèn enregistraments de qualitat amb metades de música Hindustani i Carnatic, les dues grans tradicions de la MCI. A partir d'aquí analitzem el to predominant i normalitzem la peça pel context tonal. Un element important per a descriure melodies és la identificació d'unitats temporals rellevants, per la qual cosa detectem les ocurredades de nyās svaras en la MCI, que serveixen com a marques identificadoras dels patrons melòdics més destacats.

Utilitzant aquestes característiques melòdiques, extraiem els patrons melòdics recurrents més destacats. Aquests patrons són els blocs que construeixen les estructures melòdiques, tant en la improvisació i com en la composició. Per tant, són fonamentals per a la descripció de col·leccions de música MCI. Proposem partir d'un enfocament no supervisat que utilitza eines d'anàlisi basades en sèries temporals per descobrir patrons melòdics en grans col·leccions de música. En primer lloc, hem realitzat un anàlisi supervisat extensiu sobre la similitud melòdica, que és un component fonamental per al descobriment de patrons. A continuació, millorem els resultats (respecte al millor competidor segons l'estat de la qüestió) explotant les característiques peculiars dels patrons melòdics de la música MCI. Per identificar patrons musicalment rellevants, explotem les relacions entre els patrons descoberts mitjançant un anàlisi de xarxa. Extenses proves realitzades amb músics professionals revelen que els patrons melòdics descoberts són musicalment interessants i significatius.

Finalment, fem servir els nostres resultats per al reconeixement de *rāgas* en actuacions gravades d'IAM. Proposem dos enfocaments nous que capturen conjuntament el

to i els aspectes temporals de la melodia. El primer enfoc utilitza patrons melòdics, l'aspecte més important per als éssers humans a l'hora d'identificar rāgas. Utilitzem els patrons melòdics descoberts i fem servir tècniques de modelatge de temes (“topic modeling” en anglès), on considerem que la interpretació d'un raga és similar a la descripció textual d'un tema. En el nostre segon enfocament, proposem utilitzar el “time delayed melodic surface”, una característica innovadora basada en coordenades de retard que capture l'evolució melòdica del rāga. Amb aquests enfocaments demostrem una precisió sense precedents per al reconeixement de rāgas en el conjunt de dades més gran utilitzat mai per a aquesta tasca. Encara que el nostre enfocament està basat en les característiques de les melodies MCI i la tasca en qüestió, creiem que la nostra metodologia es pot estendre fàcilment a altres tradicions de la música on la melodia és rellevant.

En general, hem incorporat nous mètodes computacionals per a l'anàlisi de diversos aspectes melòdics per a interpretacions de MCI, a partir dels quals descrivim i inter-connectem gran quantitat d'enregistraments de música. En aquest procés hem recopilat dades i hem desenvolupat diverses eines que poden ser utilitzades per a diferents estudis computacionals per a MCI, específicament en la caracterització de rāgas, composicions i artistes. Les tecnologies resultants d'aquest treball d'investigació són part de diverses aplicacions desenvolupades dins el projecte CompMusic que pretén millorar la descripció, l'experiència auditiva, i la pedagogia de la MCI.

Resumen

La descripción automática del contenido de música grabada es crucial para la interacción con grandes colecciones de grabaciones de audio y para el desarrollo de nuevas herramientas que faciliten la pedagogía musical. La melodía es un aspecto fundamental para la mayoría de las tradiciones musicales, y es por tanto un componente indispensable para tal descripción. En esta tesis desarrollamos propuestas computacionales para el análisis de aspectos melódicos de alto nivel en interpretaciones musicales de Música Clásica de la India (MCI), con las que podemos describir e interrelacionar grandes cantidades de grabaciones de audio. Debido a su complejidad melódica y a su sólido marco teórico, la descripción de la melodía en MCI más allá de la línea melódica supone un interesante y desafiante objeto de investigación. Analizamos melodías en su contexto tonal, identificamos patrones melódicos, comparamos ambos tanto en piezas individuales como entre diferentes piezas, y finalmente caracterizamos el contexto melódico específico de MCI, los rāgas. Todos estos análisis se llevan a cabo mediante métodos dirigidos por datos en corpus de música de considerable tamaño y meticulosamente organizados.

La tesis comienza con la confección y estructuración de los mayores corpus musicales hasta la fecha de las dos tradiciones de MCI, indostaní y carnática. Dichos corpus están formados por grabaciones de audio de alta calidad y sus correspondientes metadatos. De estas extraemos la línea melódica predominante y la normalizamos según la tónica de su contexto. Un elemento importante para la descripción de melodías es la identificación de unidades temporales significativas, para lo que proponemos detectar en música indostaní las ocurrencias de nyās svaras, marcas que delimitan patrones melódicos musicalmente prominentes.

A partir de estas características melódicas, extraemos patrones melódicos recurrentes y musicalmente relevantes. Estos patrones son las unidades básicas con las que se construyen estructuras melódicas tanto en improvisaciones como composiciones, y por tanto son fundamentales para la descripción de colecciones de audio en MCI. Proponemos un método no supervisado basado en el análisis de las series temporales para el descubrimiento de patrones melódicos en colecciones musicales de tamaño considerable. En primer lugar llevamos a cabo un análisis supervisado en profundidad de similitud melódica, que es el componente crítico para el descubrimiento de patrones. A continuación mejoramos la propuesta más competitiva aprovechando las características melódicas propias de MCI. Para identificar patrones musicalmente significativos, aprovechamos las relaciones entre los patrones descubiertos mediante la implementación de análisis de redes. Exhaustivas evaluaciones auditivas por parte de músicos profesionales de los patrones melódicos descubiertos revelan que estos son musicalmente interesantes y significativos.

Finalmente, utilizamos nuestros resultados para el reconocimiento de rāgas en interpretaciones grabadas de MCI. Proponemos dos métodos nuevos que captan conjuntamente los aspectos tonales y temporales de la melodía. Nuestro primer método se sirve de patrones melódicos, los principales indicadores para la identificación de rāgas por parte de oyentes humanos. Utilizamos los patrones melódicos descubiertos y empleamos técnicas de modelado de temas, en las que equiparamos la interpretación de un rāga a la descripción textual de un tema. En nuestro segundo método, proponemos una superficie melódica de tiempo de retardo, una característica nueva basada en las coordinadas de retraso que captan el contorno melódico de un rāga. Con estos métodos alcanzamos precisiones sin precedentes en el reconocimiento de rāgas en los mayores conjuntos de datos nunca usados para esta tarea. Aunque nuestra propuesta se fundamenta en las características de las melodías en MCI y la tarea en cuestión, creemos que nuestra metodología puede ser fácilmente aplicable a otras tradiciones musicales predominantemente melódicas.

En general, hemos construido nuevos métodos computacionales para el análisis de varios aspectos melódicos de interpretaciones grabadas de MCI, con las que describimos e interrelacionamos grandes cantidades de grabaciones musicales. En este proceso hemos desarrollado varias herramientas y reunido datos que pueden ser empleados en numerosos estudios computacionales de MCI, específicamente para la caracterización de rāgas, composiciones y artistas. Las tecnologías resultantes de este trabajo de investigación son parte de varias aplicaciones desarrolladas en el proyecto CompMusic para la mejora de la descripción, experiencia de escucha, y enseñanza de MCI.

Contents

Abstract	XI
Resum	XIII
Resumen	XV
Contents	XVII
List of Symbols	XXI
List of Figures	XXIII
List of Tables	XXVII
1 Introduction	1
1.1 Motivation	1
1.2 Scientific Context	3
1.3 Opportunities and Challenges	4
1.4 Scope and Objectives	7
1.5 Thesis Outline	9
2 Background	13
2.1 Introduction	13
2.2 Terminology	13
2.3 Music Background	15
2.3.1 Indian Art Music	15
2.3.2 Melody in Indian Art Music	17
2.4 Related Work in Indian Art Music	23
2.4.1 Tonic Identification	23
2.4.2 Melodic Pattern Processing	31
2.4.3 Rāga Recognition	37
2.5 Related Work from Other Music Traditions	44
2.5.1 Key and Tonality Modeling	45
2.5.2 Pattern Processing in Music	47
2.6 Mathematical Background	55
2.6.1 Distance Measures	55
2.6.2 Dynamic Time Warping	56
2.6.3 Lower-bounds for DTW distance	57

2.7	Summary	58
3	Music Corpora and Datasets	61
3.1	Introduction	61
3.2	CompMusic Research Corpora	63
3.2.1	Criteria for Building CompMusic Corpora	63
3.2.2	Carnatic Music Corpus	65
3.2.3	Hindustani Music Corpus	70
3.2.4	Open-access Music Corpus	74
3.2.5	Storage and Access of Corpus	77
3.3	Test Datasets	78
3.3.1	Tonic Identification Datasets	78
3.3.2	Nyās Dataset	81
3.3.3	Melodic Similarity Dataset	81
3.3.4	Rāga Recognition Datasets	83
3.4	Summary	85
4	Melody Descriptors and Representations	87
4.1	Introduction	87
4.2	Tonic Identification: Approaches and Comparative Evaluation	88
4.2.1	Comparative Evaluation Setup	88
4.2.2	Results and Discussion	89
4.2.3	Summary of the Comparative Evaluation	99
4.2.4	Correcting Common Errors in Tonic Identification	99
4.3	Melody Processing	100
4.3.1	Predominant Pitch Estimation	100
4.3.2	Pitch Post-processing	102
4.3.3	Melody Representation	106
4.4	Tani Segmentation	107
4.5	Nyās Svara Segmentation	109
4.5.1	Method	111
4.5.2	Experimental Setup	115
4.5.3	Results and Discussion	117
4.5.4	Summary	119
4.6	Summary	119
5	Melodic Pattern Processing: Similarity, Discovery and Characterization	121
5.1	Introduction	121
5.2	Melodic Similarity: Approaches and Evaluation	124
5.2.1	Method	126
5.2.2	Evaluation Methodology	130
5.2.3	Results and Discussion	131
5.3	Improving Melodic Similarity	135

5.3.1	Method	137
5.3.2	Evaluation	142
5.3.3	Results and Discussion	143
5.4	Melodic Pattern Discovery	146
5.4.1	Method	147
5.4.2	Evaluation	157
5.4.3	Results and Discussion	158
5.5	Characterization of Melodic Patterns	161
5.5.1	Method	164
5.5.2	Evaluation	170
5.5.3	Results and Discussion	171
5.6	Summary and Conclusions	174
6	Automatic Rāga Recognition	177
6.1	Introduction	177
6.2	Pattern-Based Rāga Recognition	178
6.2.1	Vector Space Modeling of Melodic Patterns	179
6.2.2	Evaluation	183
6.2.3	Results and Discussion	185
6.3	Time Delayed Melodic Surface for Rāga Recognition	191
6.3.1	Time Delayed Melodic Surface	192
6.3.2	Evaluation	196
6.3.3	Results and Discussion	198
6.4	Effect of Dataset on Accuracy	202
6.5	Summary and Conclusions	204
7	Applications	207
7.1	Introduction	207
7.2	Dunya	207
7.3	Mobile Applications: Sarāga and Riyāz	210
7.4	Demos	212
7.5	Computational Musicology	216
7.6	Summary	219
8	Summary and Future Perspectives	221
8.1	Introduction	221
8.2	Summary of Contributions	222
8.3	Future Perspectives	225
A	Publications by the Author	229
B	Resources	233
C	Additional Figures and Tables	237

D Glossary	247
D.1 Acronyms	247
D.2 Music Terms	250
Bibliography	253

List of Symbols

The following is a list of different symbols used in the dissertation along with a short description of each symbol.

Symbol	Description
\mathcal{A}	Ordered list of node counts in a community of network of patterns across rāgas
\mathcal{B}	Ordered list of node counts in a community of network of patterns across recordings
B	Octave wrapping integer binning operator
c	Centroid of the distribution of nodes over recordings
C	Clustering coefficient of a network
C_i	Melodic pattern category in the Carnatic music dataset
\mathcal{C}	Community in a network
d	DTW local cost function
D	Distance measure for computing melodic similarity
\mathcal{G}	Undirected network of melodic patterns
\mathbf{G}	Goodness measure of a community in a network
H_i	Melodic pattern category in the Hindustani music dataset
I	Indicator function
i	Index
j	Index
k	Index
K	Number of nearest neighbors
\mathcal{L}	Likelihood of the representative rāga in a community of a network of melodic patterns
m	Index
\mathcal{N}	Nyās svara segment
\mathcal{O}	Big O notation
p	p-value in statistical hypothesis testing
p	Predominant pitch in Hz
\hat{p}	Predominant pitch in Cents
\wp	Melodic pattern
\wp	Array of melodic patterns

Symbol	Description
r	Audio recording
\mathcal{R}	Corpus of audio recordings
S	Svara frequency in Cents
\check{S}	time delayed melodic surface (TDMS)
\bar{S}	Power compressed TDMS
\hat{S}	Smoothened TDMS
\mathbf{S}	Normalized TDMS
\check{s}	One element of TDMS
\mathcal{T}	Tonic pitch of an audio recording in Hz
T	Time stamp
\mathcal{V}	Vocabulary of melodic patterns
w	Weight of an edge in a network of melodic patterns
W	Length of a melodic pattern in seconds
\hat{W}	Length of a melodic pattern in samples
Z	Normalization type used in melody representation
Δ	Distance between melodic patterns
$\tilde{\Delta}$	Melodic similarity threshold
ε	Allowed pitch deviation used in nyās segmentation
ψ	Temporal threshold used in nyās segmentation
α	Power compression factor in TDMS computation
v	Binary flatness measure used in nyās segmentation
Ω	Uniform time-scaling factor
Φ	Svara duration truncation threshold
ρ	Max pitch deviation used in nyās segmentation
σ_g	Standard deviation of the Gaussian kernel used in the TDMS computation
Θ	Heaviside step function
v	Flatness measure
\tilde{v}	Flatness threshold
Υ	Complexity weighting in melodic similarity computation
ω	Sampling rate of the pitch sequence in Hz
ς	Maximum error parameter in piece-wise linear segmentation method
ζ	Complexity estimate of a melodic pattern

List of Figures

1.1 Examples of the characteristic melodic phrases in Hindustani Music	7
1.2 Computational melodic analyses addressed in this thesis	9
2.1 Example of a concert setup in Carnatic music	17
2.2 An example of the kampitam gamaka in Carnatic music	20
2.3 An example of the mīnd alankār in Hindustani music	21
2.4 An example of chalan in Hindustani music	22
2.5 General block diagram of the tonic identification approaches	24
2.6 Spectrogram of an excerpt of Hindustani music	27
2.7 Pitch histograms constructed using two different methods	28
3.1 Details of the Carnatic music corpus	66
3.2 The number of artists versus the number of concerts	69
3.3 Overlap between the artists in the Carnatic music corpus and Kutcheris.com	69
3.4 Details of the Hindustani music corpus	72
3.5 Details of the open-access Carnatic music corpus	74
3.6 Details of the available annotations for the open-access Carnatic music corpus	75
3.7 Details of the open-access Hindustani music corpus	76
3.8 Details of the available annotations for the open-access Hindustani music corpus	76
3.9 Details of the rāga recognition dataset comprising Carnatic music recordings	85
3.10 Details of the rāga recognition dataset comprising Hindustani music recordings	86
4.1 Tonic identification accuracies of different approaches on four datasets . .	92
4.2 Tonic identification accuracies for Hindustani, Carnatic, male and female excerpts	93
4.3 Percentage of Pa, Ma and ‘Other’ type errors in tonic identification	96
4.4 Percentage of Pa, Ma and ‘Other’ type errors in tonic identification, using editorial metadata	97
4.5 Percentage of Pa, Ma and ‘Other’ type errors in tonic identification for different categories	98
4.6 Example of the predominant pitch representation of melody	100
4.7 Example of an octave error in predominant pitch contour.	103
4.8 Example of a post-processed predominant pitch segment	104

4.9	Pitch patterns corresponding to the mṛdaṅgam strokes	108
4.10	Illustration of the spectrogram a vocal and a tani section	108
4.11	A fragment of a pitch contour showing nyās segments	110
4.12	Block diagram of the proposed approach for nyās segmentation.	111
4.13	Illustration of the nyās segmentation process	112
4.14	Example of a normalized octave folded pitch histogram	114
5.1	Two types of approaches for pattern extraction in music recordings.	122
5.2	Examples of difference occurrences of a melodic pattern	125
5.3	Block diagram for computing melodic similarity	127
5.4	Matrix indicating the statistical significance of the performance difference between different method variants	133
5.5	Boxplot of average precision values for different types of melodic patterns	133
5.6	Examples of different occurrences of the rāga motifs	136
5.7	Examples of melodic patterns after duration truncation	137
5.8	Illustration of an erroneous case of melodic similarity in Carnatic music .	138
5.9	Block diagram for an improved melodic similarity computation	139
5.10	MAP scores for different duration truncation values	143
5.11	Boxplot of average precision for different types of melodic patterns in the Hindustani music dataset	145
5.12	Boxplot of average precision for different types of melodic patterns in the Carnatic music dataset	146
5.13	Block diagram for melodic pattern discovery	148
5.14	Block diagram of data processing modules for melodic pattern discovery .	149
5.15	Histograms of autocorrelation of the pitch subsequences for different lags	150
5.16	ROC curve for ‘flat’ and ‘non-flat’ region classification	152
5.17	Illustration of output of different distance measures	156
5.18	Distance distribution of seed melodic patterns	158
5.19	ROC curve for seed pairs and search patterns	159
5.20	Boxplot of average precision values for the different variants of the rank refinement method	161
5.21	Examples of the discovered melodic patterns.	162
5.22	Block diagram for characterizing melodic patterns	164
5.23	Evolution of the clustering coefficient of a network of melodic patterns .	167
5.24	Graphical representation of a network of melodic patterns	169
5.25	Mean musician ratings for the discovered melodic patterns	172
5.26	Histogram of mean musician ratings for all of the 100 melodic patterns .	173
6.1	Block diagram of the proposed phrase-based approach to rāga recognition.	180
6.2	Evolution of $\mathbf{C}(\mathcal{G})$, $\mathbf{C}(\mathcal{G}_r)$ and $\mathbf{C}(\mathcal{G}) - \mathbf{C}(\mathcal{G}_r)$ over values of $\tilde{\Delta}$	181
6.3	Accuracy of M _{VSM} and $\mathbf{C}(\mathcal{G}) - \mathbf{C}(\mathcal{G}_r)$ for different values of $\tilde{\Delta}$	187
6.4	Confusion matrix of the classification results by M _{VSM} on RRD _{CMD} . . .	188
6.5	Confusion matrix of the classification results by M _{VSM} on RRD _{HMD} . . .	189

6.6	Block diagram for TDMS computation.	193
6.7	TDMS before and after post-processing	195
6.8	Accuracy of $M_{\text{TDMS}}^{\text{KL}}$ as a function of different parameter values	200
6.9	Confusion matrix of the classification results by $M_{\text{TDMS}}^{\text{KL}}$ on RRD_{CMD}	201
6.10	Accuracy of M_{VsSM} as a function of number of rāgas in a subset of RRD_{CMD} and RRD_{HMD}	203
7.1	Screenshot of the recording page in Dunya	209
7.2	Screenshots of the mobile application, Sarāga	211
7.3	Screenshots of the mobile application, Riyāz.	213
7.4	A web demo for navigating through the discovered melodic patterns	214
7.5	A web demo of a network of the discovered melodic patterns	215
7.6	Screenshot of Rāgawise	217
7.7	Trajectories of the highest salience bin in long-time averaged pitch histograms computed across breath-phrases	218
7.8	Histogram of the ratio of the inter-onset-intervals of salient svaras across breath-phrases.	218
C.1	Examples of gamaka in Carnatic music	237
C.2	Confusion matrix of the classification results by M_{PC} on RRD_{CMD}	238
C.3	Confusion matrix of the classification results by M_{PC} on RRD_{HMD}	239
C.4	Confusion matrix of the classification results by M_{TDMS} on RRD_{HMD}	240

List of Tables

2.1	Summary of the existing tonic identification approaches.	25
2.2	Summary of the melodic pattern processing methods for IAM	32
2.3	Melodic characteristics utilized by the existing rāga recognition methods .	38
2.4	Summary of the existing rāga recognition methods	39
3.1	Coverage of the Carnatic music corpus	68
3.2	Completeness of the Carnatic music corpus	71
3.3	Coverage of the Hindustani music corpus	73
3.4	Completeness of the Hindustani music corpus	73
3.5	Summary of the tonic identification datasets	79
3.6	Details of the melodic similarity datasets	82
3.7	Details of the annotated characteristic melodic patterns in MSD dataset. .	83
3.8	Details of the annotated characteristic melodic patterns in MSD _{CM} dataset.	84
4.1	Tonic identification accuracies of seven methods on six different datasets using only audio data	91
4.2	Tonic identification accuracies of seven methods on six different datasets using both audio and editorial metadata	94
4.3	F-scores for the nyās boundary detection task	118
4.4	F-scores for the nyās and non-nyās label annotation task	118
5.1	MAP score and parameter details for the three best performing variants of the method for computing melodic similarity	131
5.2	MAP score and parameter details for the three best performing variants of the method for computing melodic similarity, without using ground-truth segmentation	134
5.3	MAP scores for MSD _{CM} ^{hmd} and MSD _{CM} ^{cmd} datasets obtained by M _B , M _{DT} , M _{CW1} and M _{CW2}	144
5.4	Percentage of exits after different lower bound computations	159
5.5	MAP scores for four variants of rank refinement method for each seed pattern category	160
5.6	Details of the dataset used for studying pattern characterization	171
5.7	Mean and standard deviation of μ_ϕ for each rāga	173
6.1	Accuracy of M _{VSM} , M _{PC} and M _{GK} on RRD _{CMD}	185
6.2	Accuracy of M _{VSM} and M _{PC} on RRD _{HMD}	186
6.3	Rāga recognition accuracy of the TDMS-based method variants	198

C.1	Svara names and notations used in Carnatic and Hindustani music	241
C.2	List of the rāgas in RRD _{HMD} along with their constituent set of svaras . .	242
C.3	Details of RRD _{HMD} dataset for each constituent rāga	243
C.4	List of the rāgas in RRD _{CMD} along with their constituent set of svaras . .	244
C.5	Details of RRD _{CMD} dataset for each constituent rāga	245

Chapter 1

Introduction

Information technology (IT) is constantly shaping the world that we live in. Its advancements have changed human behavior and influenced the way we connect with our environment. Music being a universal socio-cultural phenomena has been deeply influenced by these advancements. The way music is created, stored, disseminated, listened to, and even learned has changed drastically over the last few decades. A massive amount of audio music content is now available on demand. Thus, it becomes necessary to develop computational techniques that can process and automatically describe large volumes of digital music content to facilitate novel ways of interaction with it. There are different information sources such as editorial metadata, social data, and audio recordings that can be exploited to generate a description of music. Melody, along with harmony and rhythm, is a fundamental facet of music, and therefore, an essential component in its description. In this thesis, we focus on describing melodic aspects of music through an automated analysis of audio content.

1.1 Motivation

“It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming, and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text.”

(Selfridge-Field, 1998)

The importance of melody in our musical experiences makes its analysis and description a crucial component in music content processing. It becomes even more important for melody dominant music traditions such as Indian art music (IAM), where the concept of harmony (functional harmony as understood in common practice) does not exist, and the complex melodic structure takes the central role in music aesthetics.

Melodic analysis and description is not a recent phenomenon, it has been done by musicologists for hundreds of years. However, a computational approach to this task has opened up new directions and possibilities, taking it to a different scale altogether. Through computational approaches, melodic analysis and description can be performed at the level of an entire music repertoire, as opposed to a few music pieces typically considered in manually performed musicological studies. Needless to say that such computational analysis is work in progress with continuous attempts to improve the quality of the outcomes, so that they closely resemble to what can be done by human experts.

Most of the current computational approaches that analyze high-level melodic aspects of music such as melodic similarity and motifs work with its symbolic representations, thus covering only a particular view of music. There are significant challenges in extending such approaches to analyze recorded performances, mainly due to the difficulties involved in obtaining a meaningful symbolic representation from audio recordings. Therefore, for performance oriented music traditions such as **IAM**, where symbolic music representations are practically nonexistent and the aesthetics lie in the improvisatory aspects, such approaches are not directly applicable. Besides, melody, being a cultural phenomenon, should be studied within the cultural context of a music tradition. Thus, there is a need to develop culture-aware computational approaches that exploit the specificities of a music tradition to analyze and describe the melodic aspects of recorded music performances. **IAM** with its complex melodic framework, *rāga*, and well grounded-music theory, provides an ideal context to develop such approaches.

Melodic elements in **IAM** are hierarchically organized in accordance with the *rāga* grammar. At the lowest level there are *svaras*, which concatenate to form melodic phrases. These phrases group together to form passages, finally leading to a music piece. At each level these melodic elements adhere to the *rāga* grammar. In this thesis, we focus on computational approaches that analyze these melodic elements at different hierarchical levels to describe melodic aspects of **IAM** corpora.

Automated analysis and description of high-level melodic aspects of **IAM** has manifold applications. It can enable corpora level musicological studies such as characterization of music compositions, artists and *rāgas*. Since **IAM** follows an oral pedagogy, analysis of the recorded performances can shed light on the stylistic influences of teachers on their students, and to other artists. Establishing relationships between different melodic elements across recordings in a music collection opens up ways to define novel music similarity measures, and generate semantically meaningful description in terms of higher level melodic concepts, such as *rāgas*. This further enables several applications such as structuring and organizing large music archives, *rāga*-based music retrieval and culturally relevant music navigation and discovery. A rich description of different melodic elements can aid immensely in developing novel applications that address enhanced or augmented music listening experience. This aspect is specifically relevant in the case of **IAM**, wherein the music is largely access-

ible (in terms of the understanding) to musicians and music connoisseurs owing to the complexity of the melodic structures. Furthermore, being able to analyze and characterize different melodic elements directly from the audio recordings opens up novel and creative ways to approach music pedagogy. Specifically in the context of **IAM**, where the music nuances are learned implicitly through years of training, an objective description of melodic aspects can aid music students to learn from the recorded performances of maestros.

1.2 Scientific Context

Music information research (**MIR**) is a growing interdisciplinary research field that stands at the intersection of well established disciplines such as signal processing, pattern recognition, musicology, psychoacoustics, music perception and cognition, information science, and computer science. **MIR** primarily addresses topics involved in the understanding and modeling of music using information processing methodologies (Serra et al., 2013). In particular, it aims to advance our knowledge in representing, understanding, describing, retrieving, archiving and organizing music related data. This opens up a wealth of possibilities to develop novel ways to interact with music (Casey et al., 2008; Orio, 2006; Burgoyne et al., 2015).

The field of **MIR** has made significant progress in the last two decades. Its growth is fueled by the massive surge in the digital music content. A large number of **MIR** systems aim to describe and characterize music content in terms of different musical facets such as melody, rhythm, harmony, structure, and emotion. The definition, interpretation and relevance of these musical aspects are not universal and vary significantly across different music traditions and personal, cultural, and social contexts. A significant number of existing computational approaches in **MIR** fail to account for these factors, and thus, might be hitting the so-called "glass ceiling" (Pachet & Aucouturier, 2004; Casey et al., 2008). In addition, there also exists a semantic-gap between the automatically extracted music descriptors from audio signals and the high-level music concepts that humans relate to (Celma, 2006; Casey et al., 2008). Furthermore, since the research problems undertaken in **MIR** have mainly been shaped by the western commercial music of the past few decades, many of the technologies developed within **MIR** are not directly applicable to several other music traditions of the world (Serra, 2011). Thus, there is a need to bridge this gap and take a broader perspective to describe music by taking the cultural, social and the user context into account (Serra et al., 2013). It is in this context that the CompMusic project was envisioned.

CompMusic¹ (Computational Models for the Discovery of the World's Music) is a research project funded by the European Research Council (Serra, 2011). The project focuses on five music traditions of the world: Hindustani (North India), Carnatic

¹<http://compmusic.upf.edu/>

(South India), Turkish-makam (Turkey), Arab-Andalusian (Maghreb), and Beijing Opera (China). One of the main objectives of the CompMusic project is to promote and develop multicultural perspectives in MIR. In particular, the project aims to advance the research in computational description of music by identifying musically relevant problems coming from culture-specific contexts and developing domain specific approaches to solve them. Addressing the research problems in the context of diverse music cultures will not only help in advancing the knowledge in the specific cultures, but also expand the scope of the current research in MIR. It can also help bridge the semantic gap and push the glass ceiling.

CompMusic project follows a data-driven research methodology. The efficacy of the computational approaches following such methodologies are directly impacted by the quality of the data using which they are developed. Thus, one of the goals in the CompMusic project is to create quality data corpora that are representative of the performance practices of the music traditions. The corpora compiled and curated in the CompMusic project mainly comprise commercial quality audio recordings and the associated metadata.

The work presented in this thesis is carried out as a part of the CompMusic project and aligns with its goals. In this thesis, we focus on analysis and description of melodic aspects in IAM corpora. Our efforts are directed towards the bigger goal of developing culture-aware computational approaches that can utilize domain knowledge in order to produce semantically meaningful description of music. The cultural specificities of IAM have shaped our work at each step, whether it is the identification of relevant research problems, building the data corpora, or the methodology adopted by the computational approaches. The insights gained in the process of developing culture-aware and domain specific approaches will help expand the scope of the state of the art in MIR. Our work also paves way for cross-cultural studies, which can help us better understand the influence of the cultural training on perception and cognition of different musical aspects.

1.3 Opportunities and Challenges

IAM is a highly evolved music tradition with its origins dating back as early as 1500 BC, and is alive and thriving. It is a well studied music tradition with sophisticated and grounded music theory. The literature is replete with scholarly text written on musical concepts in IAM. However, this music tradition is not explored fully from a computational analysis point of view. The established music theories and the existing musicological work provides a strong base to formulate MIR tasks and develop computational models for automatic music description.

IAM is a performance centric music tradition, which has been transmitted orally across generations following a tradition of Guru-Shishya parampara (“lineage” system). The musical compositions in IAM merely act as skeletons in music perform-

ances, and the essence of the music lies in the improvisatory aspects. As a result of which, **IAM** mainly possesses recorded music repertoire.

Reliable extraction of even a low-level melody representation such as the predominant pitch from recorded performances is a challenging task. As a result of which, computational approaches for melodic description in audio recordings are still primarily focused on extracting such representations and have not been able to address higher level melodic analyses comprehensively. Specific heterophonic characteristics of **IAM** make it feasible to obtain a low-level melody representation from audio recordings using the current state of the art predominant pitch estimation methods. This is also indicated by the past MIREX (an international **MIR** evaluation campaign) results². Compare the accuracy obtained by different algorithms on INDIAN08³, MIREX05⁴ and MIREX09 0dB⁵ datasets from MIREX-2011. The feasibility of obtaining a reasonably accurate predominant pitch representation of melody from audio recordings enables us to focus on the description of higher level melodic aspects of music performances. Thus, **IAM** provides an opportunity to broaden the scope of the computational approaches for melodic analysis and description, much beyond describing the melodic aspects of music performances by merely the pitch contours.

Although the extraction of the predominant pitch in **IAM** recordings is relatively less complicated, its abstraction into a symbolic representation is a challenging task. One way of abstracting a continuous pitch contour is by performing melody transcription, which is still a challenging and an ill-defined task in the case of **IAM**, primarily due to its meandering melodic characteristics (Widdess, 1994; Rao et al., 1999). Moreover, the process of discretization of such melodic movements may result in a loss of information relevant to the characterization and description of melodies. Thus, difficulties in abstraction of a continuous melody representation in **IAM** poses challenges in its processing.

There is no standard frequency that is used as a reference for tuning instruments and voice in the performances of **IAM**. A lead artist can choose any convenient frequency as tonic, which acts as the reference using which all accompanying instruments are tuned. Tonic pitch varies across artists, and may also vary across the performances of an artist. These factors make it difficult to directly process melodies across different artists as well as across different performances of an artist.

“Music becomes intelligible to a great extent through self-reference, i.e., through the relations of new musical passages to previously heard material. Structural repetition and similarity are crucial devices in establishing such relations.” (Cambouropoulos, 2006)

²http://www.music-ir.org/mirex/wiki/MIREX_HOME

³http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/indian08/summary.html

⁴http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/mirex05/summary.html

⁵http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/mirex09_0dB/summary.html

“Only by repetition can a series of tones be characterized as something definite. Only repetition can demarcate a series of tones and its purpose. Repetition thus is the basis of music as an art.”

(Schenker et al., 1980)

Analysis of repeating melodic patterns has been instrumental in the description of melodies, and is therefore utilized by a large number of computational approaches in MIR. Repeating melodic patterns are integral to the melodic framework in IAM, the *rāga*. They act as building blocks to construct melodies within the *rāga* grammar. There are different types of melodic patterns in IAM with their well defined functional roles. For example, a certain type of patterns correspond to melodic ornaments, some others form the opening line of compositions and, musically the most significant patterns are those that characterize *rāgas* (Section 2.3). Thus, IAM provides an interesting opportunity to develop computational approaches for discovery and characterization of melodic patterns from audio collections. However, the improvisatory nature of this music tradition makes this task challenging. The characteristic melodic phrases of *rāgas* act as the basis for artists to improvise, providing them with a medium to express creativity during *rāga* rendition. Artists bring in novelty through creatively transforming these melodic phrases as much as possible within the periphery defined by the *rāga* grammar. Therefore, the surface representation of these melodic phrases vary a lot across their occurrences. This high degree of variation in terms of the overall duration of a phrase, non-linear time warpings and the added melodic ornaments together pose a big challenge for melodic similarity computation and pattern extraction in IAM. In Figure 1.1, we illustrate this variability by showing the pitch contours of the different occurrences of three characteristic melodic phrases of *rāga Alahaiyā bilāval*. We can clearly see that the duration of a phrase across its occurrences varies a lot, and the steady melodic regions are highly varied in terms of the duration and the presence of melodic ornaments. These characteristics of the melodic patterns in IAM provide an opportunity to gain deeper insights into the perception of melodic similarity as well as how it is influenced by the cultural aspects.

Discovery of melodic patterns is a computationally complex task, specifically when performed using music parallelism (Cambouropoulos, 2006). It becomes even more challenging in the case of IAM given the long duration of the audio recordings, some of which may even last for an hour.

“The rāga is more fixed than a mode, and less fixed than the melody, beyond the mode and short of melody, and richer both than a given mode or a given melody.”

(Martinez, 2001, p. 96)

“A rāga is not a tune, nor is it a ‘modal’ scale, but rather a continuum with scale and tune as its extremes.”

(Powers, 1959)

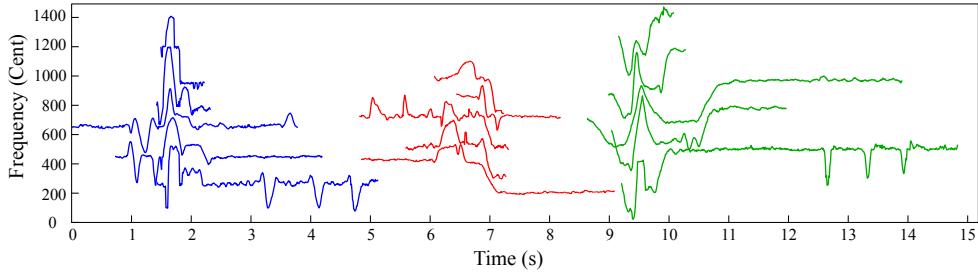


Figure 1.1: Pitch contours of occurrences of three different characteristic melodic phrases in Hindustani music. Contours are transposed in frequency, and shifted in time for a better visualization.

Rāga thus is a fascinating topic of research, specifically in the context of MIR. Being a complex melodic framework it involves an intricate interplay between different melodic elements both in terms of the tonality and their temporal relations. Therefore, its computational characterization and recognition poses unique challenges as well as provides new opportunities. It is worth mentioning that recognizing rāga in a musical performance requires domain expertise, which further emphasizes the complexity of the task.

IAM thus provides a context that is highly conducive to developing novel computational approaches to describe higher level melodic aspects of music, specifically in collections of recorded performances.

1.4 Scope and Objectives

Analysis and description of melodic aspects of music is a broad research topic that can be approached from a number of perspectives and different academic disciplines. In this thesis, we take a data-driven engineering approach and focus solely on the computational aspects of melodic analysis, making use of the established music theories. Our applied research methodology stands at the intersection of signal processing, machine learning and time-series analysis. We focus on content-based processing, wherein the input data to our approaches comprise mainly audio recordings, and in a few cases their associated editorial metadata. The approaches proposed in our work are developed and evaluated using music collections of IAM that includes both Hindustani and Carnatic music. We now outline our broad objectives in this thesis.

- To curate and structure representative music corpora of IAM that comprise audio recordings and the associated metadata, and use that to compile sizable and well annotated tests datasets for melodic analyses.

- To develop data-driven computational approaches for discovery and characterization of musically relevant melodic patterns in sizable audio collections of **IAM**
- To devise computational approaches for automatically recognizing *rāgas* in recorded performances of **IAM**.

In order to achieve these broad objectives, there are a number of computational tasks pertaining to melodic analysis of **IAM** that are addressed in this thesis. A visual summary of these tasks is provided in Figure 1.2. We now briefly enumerate these tasks.

Melodies in **IAM** need to be analyzed within their tonal context set by the tonic pitch of the lead performer. Therefore, automatic tonic identification becomes the first step in melodic analysis of this music. Though a number of methods are proposed for this task, including our own, there is no consensus on the best performing approach as they are evaluated on different datasets and experimental conditions (Gulati et al., 2014a). In this thesis, we aim to perform an exhaustive comparative evaluation of different tonic identification approaches on a variety of music material to select the most robust and accurate approach to be used in our work.

An important step in analyzing melodies, specifically in a pattern-based analysis, is the identification of meaningful melodic segments. We aim to devise a segmentation approach that can facilitate melodic pattern processing tasks in **IAM**.

Computation of melodic similarity is critical in pattern processing of melodic sequences. Since characteristics of melodies in **IAM** differ significantly from those in several other music traditions, it becomes important to investigate thoroughly the influence of different melody representations, melody normalization strategies and similarity measures on the computation of melodic similarity for this tradition. Such an analysis will not only reveal the challenges involved in the task, but will also help in identifying ways in which specific peculiarities of this music tradition can be exploited for improving melodic similarity.

Discovery of short-duration melodic patterns in audio recordings is a challenging task. It becomes even more challenging when the discovery is performed at the level of an entire corpus that comprises hundreds of hours of audio content. We aim to develop a methodology to mine meaningful repeating melodic patterns in large audio collections of **IAM**.

The characteristics and the functional roles of repeating melodic patterns vary a lot across music traditions. For some music traditions frequently occurring patterns might be the most important ones. Whereas, in some others, highly repetitive patterns might be musically trivial. Characterization of the melodic patterns should thus be studied within the context of a specific music tradition. We aim to develop an approach that can exploit domain-specific knowledge to effectively characterize melodic patterns in **IAM**.

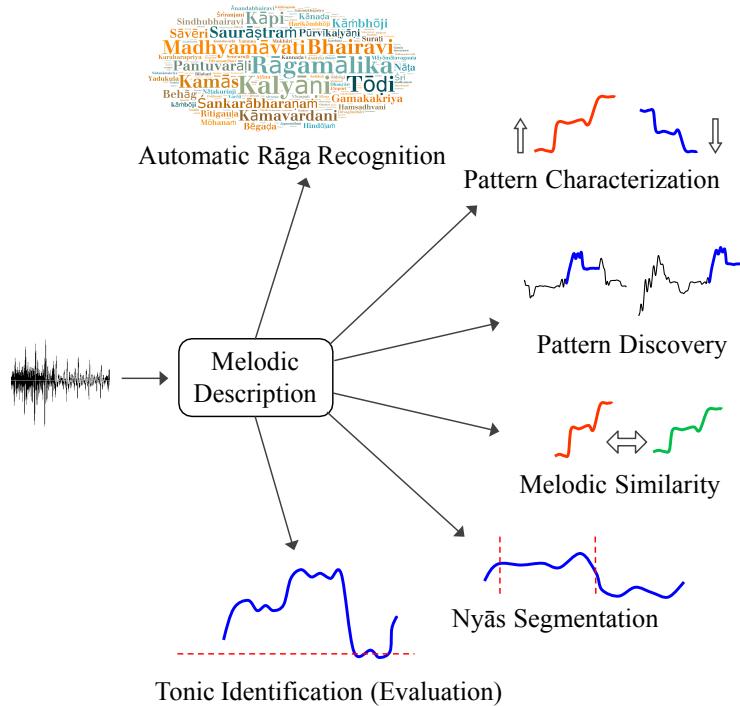


Figure 1.2: Computational tasks within melodic analysis of IAM that are addressed in this thesis.

Finally, we aim to investigate one of the most studied and relevant topics in the computational analysis of IAM, automatic rāga recognition. Our goal is to devise an approach that can successfully utilize both the tonal and the temporal characteristics of melody to perform the task.

Our work aligns with the philosophy of open-access and reproducible research. The data and the code pertaining to this thesis is made publicly available online (Appendix B).

1.5 Thesis Outline

There are eight chapters in this thesis, wherein the primary contributions are contained in Chapter 3 to 6. Each of these chapters contains an introduction, the main body and a summary of the key results and conclusions. A significant amount of the content in these chapters is derived from our publications Gulati et al. (2014a,b,c, 2015b,c, 2016c,b,a). Most of the work in these papers is done in collaboration with other researchers and musicians, which is duly indicated wherever required.

In Chapter 2, we provide an overview of the music and scientific background, with

emphasis on the existing literature relevant to the work presented in this thesis. We start with a brief introduction to **IAM** and its music concepts pertaining to melody (Section 2.3). We present our review of the current computational approaches for tonic identification, *rāga* recognition and melodic pattern processing in the context of **IAM** (Section 2.4). We critically analyze and compare these existing approaches in terms of the algorithmic design and evaluation methodology, in which we highlight their shortcomings and identify potential avenues of scientific contributions. In addition, we also present a brief review of the existing literature on tonality modeling and pattern processing in **MIR**, covering topics such as structural segmentation, motivic analysis and query-by-humming (QBH) (Section 2.5).

In Chapter 3, we describe the **IAM** corpora and different test datasets that are curated as a part of our work within the CompMusic project. We enumerate the set of design criteria followed to compile the music corpora and present a short evaluation of the goodness of the corpora with respect to these criteria (Section 3.2.1). Subsequently, a detailed description of a number of test datasets used for evaluations in this thesis is provided (Section 3.3).

In Chapter 4, we describe the processes followed to extract relevant melody descriptors and melody representations that are used by the methods described in the subsequent chapters. We start with a comparative evaluation of different tonic identification approaches to select the best approach to work with in this thesis (Section 4.2). Subsequently, we describe the procedures followed for extracting and post-processing the predominant pitch from audio recordings (Section 4.3). We then describe the processing steps applied to segment solo percussion sections, *tanis*, in the performances of Carnatic music (Section 4.4). Such sections need to be discarded in the pre-processing stage of all our melodic analysis approaches. With the low-level melody representation and the tonal context provided by the tonic we begin to perform higher level melodic analyses. We present an approach to segment melodies based on the concept of *nyās svaras*, which serve as landmarks that demarcate melodic patterns in Hindustani music (Section 4.5).

In Chapter 5, we present our main contributions and describe approaches for different computational tasks within melodic pattern processing. There are three related tasks addressed in this chapter, melodic similarity computation, pattern discovery and pattern characterization. We first investigate different choices of melody representation, distance measure and normalization strategy for computing melodic similarity in the context of *rāga* motifs in **IAM** (Section 5.2). It includes an exhaustive evaluation of different procedures and their parameter settings commonly used for this task. We subsequently describe ways to improve melodic similarity by exploiting peculiar characteristics of melodies in **IAM** (Section 5.3). Having learned the optimal set of procedures and system parameters in a supervised setup, we then utilize this knowledge for discovering melodic patterns using an unsupervised methodology (Section 5.4). Finally, we describe our approach to characterize the discovered melodic patterns in order to identify *rāga* motifs (Section 5.5).

In Chapter 6, we present the other significant part of our scientific contributions in the thesis. This chapter addresses one of the most studied topics in computational analyses of **IAM**, automatic **rāga** recognition, for which we propose two novel approaches. Our first approach utilizes the discovered melodic patterns and employs vector space modeling techniques to perform this task (Section 6.2). Our second approach uses a novel melodic representation, the **time delayed melodic surface (TDMS)**, which encodes both the tonal and the temporal aspects of melodies that are relevant to characterize **rāgas** (Section 6.3). We evaluate these methods and compare their performance with the state of the art methods using the largest datasets ever used for this task, wherein they outperform state of the art by large margins. Results prove the feasibility and effectiveness of using melodic patterns and **TDMS** representations of melody for **rāga** recognition on sizable datasets.

In Chapter 7, we present demos and a few concrete examples of applications that utilize the outcomes of our research work presented in this thesis. In particular, we introduce **Dunya**, a system that consolidates and provides access to the data, tools and technology developed in the CompMusic project (Section 7.2). In order to demonstrate the outcome of our melodic pattern discovery and **rāga** recognition approaches more directly, we present web-based demos (Section 7.4). To emphasize the utility of our work in a commercial context, we present two mobile applications: **Sarāga**, which provides an enhanced music listening experience, and **Riyāz**, which facilitates self-paced learning of both Hindustani and Carnatic music (Section 7.3).

Finally, in Chapter 8 we present an overall summary of the thesis, list our main contributions, and discuss possible future perspectives for melodic description in **IAM**.

This thesis also contains four appendix sections. In Appendix A, we list the relevant publications by the author. In Appendix B, we provide links to the relevant resources pertaining to our work such as music corpora, datasets, code, and other relevant tools. All the additional figures and tables that help in a better interpretation of our evaluation results are contained in Appendix C. Appendix D presents the glossary of the abbreviations and other terms used in this thesis.

Chapter 2

Background

2.1 Introduction

In this chapter, we present our review of the existing literature related with the work presented in this thesis. In addition, we also present a brief overview of the relevant music and mathematical background. We start with a brief discussion on the terminology used in this thesis (Section 2.2). Subsequently, we provide an overview of the selected music concepts to better understand the computational tasks addressed in our work (Section 2.3). We then present a review of the relevant literature, which we divide into two parts. We first present a review of the work done in computational analysis of **IAM** (Section 2.4). This includes approaches for tonic identification, melodic pattern processing and automatic *rāga* recognition. In the second part, we present relevant work done in **MIR** in general, including topics related to pattern processing (detection and discovery), and key estimation (Section 2.5). Finally, we provide a brief overview of selected scientific concepts (Section 2.6).

2.2 Terminology

In this section, we provide our working definition of selected terms that we have used throughout the thesis. To begin with, it is important to understand the meaning of melody in the context of this thesis. As we see from the literature, defining melody in itself has been a challenge, with no consensus on a single definition of melody (Gómez et al., 2003; Salamon, 2013). We do not aim here to formally define melody in the universal sense, but present its working definition within the scope of this thesis. Before we proceed, it is important to understand the setup of a concert of **IAM**. Every performance of **IAM** has a lead artist, who plays the central role (also literally positioned in the center of the stage), and all other instruments are considered as accompaniments. There is a main melody line by the lead performer, and typically, also a melodic accompaniment that follows the main melody. A more detailed

description of the concert setup in **IAM** is provided in Section 2.3. Since **IAM** is predominantly a performance-based music tradition, even the studio recordings and the released commercial music follow the same setup.

In the context described above, merging relevant parts of the definitions given by Paiva et al. (2006) and Levitin (2002) covers to a large extent the scope of melody in **IAM** from a computational point of view. These definitions are:

“the dominant individual pitched line in a musical ensemble”

(Paiva et al. (2006))

“an auditory object that emerges from a series of transformations along the six dimensions: pitch, tempo, timbre, loudness, spatial location, and reverberant environment”

(Levitin (2002))

The first definition falls short of considering several other dimensions of sound such as timbre and loudness that are important in the perception and production of melody, which are taken into account in the second definition. The concept of audio as a mixture of sounds from multiple instruments (*‘ensemble’*) is missing in the second definition. The idea of continuity and smoothness of melody is expressed by *‘line’* in the former and by *‘series of transformations’* in the latter. Combining these two definitions we obtain our working definition as:

“an auditory object that emerges from a continuous series of transformations along six dimensions: pitch, tempo, timbre, loudness, spatial location, and reverberant environment, by the dominant individual melodic source in a music ensemble.”

Though our definition of melody considers several dimensions of sound, for computational purposes we use a low-level representation of melody that describes its pitch and temporal dimension. We represent melody by a continuous pitch time-series corresponding to the lead artist in an audio recording, also referred to as predominant pitch in the subsequent chapters. In **MIR** it is also commonly referred to as the predominant melody.

We now proceed to define the terms such as melodic patterns, phrases and motifs. It is important to disambiguate them with other seemingly synonymous terms such as melodic fragment and segment. Melodic fragment and segment in this thesis refer to a continuous time region in melody (essentially a pitch subsequence). Both these terms do not entail any musical connotation and are used synonymously in this document. On the other hand, by a melodic pattern we refer to a recurring melodic fragment, wherein the scope and the meaning of repetition is contingent on the perceived melodic similarity by expert listeners in **IAM**. Terms such as melodic patterns, fragments and segments thus refer to temporal units in our low-level representation of

melodies. The term melodic phrase on the other hand is used in the context of music, as being a unit of melody that encapsulates an idea or a musical thought by an artist. However, even this term does not necessarily imply the phrase being characteristic of a *rāga*. To denote the characteristic phrase of a *rāga* we specifically use the term ‘*characteristic*’ along with melodic phrase, or at times simply denote by the term *rāga* motif. The term polyphonic audio in the context of music recordings in our work basically signifies that the recordings comprise a mixture of multiple instruments played simultaneously. It does not mean that the music or melody is polyphonic (functional polyphony) as understood in the context of western classical music.

2.3 Music Background

In this section, we provide a brief introduction to **IAM**. We first briefly describe the general aspects of this music tradition and, subsequently, provide a short introduction to the musical concepts related to the melodic aspects. Note that the description provided here is not comprehensive, and it is essentially to facilitate the understanding of the subsequent chapters. For a deeper understanding of these concepts we provide additional references wherever needed.

2.3.1 Indian Art Music

In our work, **Indian art music (IAM)** refers to two art music traditions of the Indian subcontinent: Hindustani music, also known as North Indian music (Bor et al., 2010; Danielou, 2010), prominent in the northern and central regions of India, Pakistan, Nepal, Afghanistan and Bangladesh, and Carnatic music, widespread in the southern regions of the Indian subcontinent (South India and Srilanka) (Viswanathan & Allen, 2004; Singh, 1995). **IAM** is also commonly referred to as **Indian classical music (ICM)**. Throughout the thesis we use the term “art” instead of “classical” to refer to these music traditions. Raja (2012)[p. 1] presents an interesting argument emphasizing the appropriateness of such a terminology. To give a brief historical perspective, the roots of **IAM** can be traced back to *sāmved*, which is one of the four *vedas* that describes music at length (Trivedi, 2008; Singh, 1995). The *sāmved* dates back to around 1000 BC, and consists of a collection of religious hymns (taken from *ṛgved*), to be sung using specifically indicated melodies called as *sāmagān* (Griffith, 2004). However, the current form of **IAM** is a confluence resulting from the cultural interactions between the Persian, Greek, Arabic, Iranian and Indian cultures (Kaul, 2007; Saraf, 2011; Singh, 1995).

With its long existing history, **IAM** continues to thrive in diverse sociocultural contexts both inside and outside of India. There is an active audience base, and several music festivals such as Sawai Gandharva Bhimsen Mahotsav⁶ and Madras Music

⁶<http://sawaigandharvabhimsenmahotsav.com/>

Season⁷ that exclusively feature these art music traditions. Notably, Madras Music Season is also one of the largest music festivals in the world that organizes more than 1500 individual artist concerts in a span of six weeks. **IAM** is a well studied music tradition with sophisticated and grounded music theory. It has a substantial musico-logical literature and scholarly text written on different musical concepts.

Over the centuries, **IAM** has been orally transmitted across generations, following a hierarchical model of music training such as *gharānā* (or school of music) in Hindustani music (Saraf, 2011; Mehta, 2008). Though the fundamental musical concepts used across *gharānās* are the same, each *gharānā* has its own ideology and characteristic style of music performance (Deshpande, 1989).

Both Hindustani and Carnatic music are performance oriented music traditions, and are mainly improvisatory in nature. In Carnatic music, a concert, also referred to as *kachēri*, is the natural unit of music performance. It is the unit typically considered for organization and digital distribution of Carnatic music content. A concert of Carnatic music typically comprises around 10 music pieces. Though Carnatic music is improvisatory in nature, the performances are based on compositions. Most of the compositions are to be sung, as a result of which, vocal music is dominant in Carnatic music. Even in instrumental music, artists aim to mimic vocal singing (Viswanathan & Allen, 2004). In Hindustani music, individual music pieces tend to be long in duration (one single piece can last up to an hour). A concert typically contains a handful of such pieces. Vocal music is dominant in Hindustani music as well. However, as compared to Carnatic music instrumental performances in Hindustani music are much more prevalent.

We now describe the performance (or concert) setup in **IAM**, which also gives us an idea about the characteristics of the recorded acoustic signal. **IAM** is essentially heterophonic in nature, with a main melody being sung or played by the lead artist (Bagchee, 1998). Quintessentially, an instrument provides melodic accompaniment and follows the melody of the lead performer (Viswanathan & Allen, 2004). A typical arrangement in a performance of **IAM** consists of a lead performer (in rare cases a duo), a rhythm accompaniment generally provided by *tablā* in Hindustani music and *mṛdāngam* in Carnatic music, a constantly sounding drone in the background, and frequently, a melodic accompaniment using harmonium or *sāraṅgi* in Hindustani music and violin in Carnatic music. The drone sound which is mainly produced by a *tānpura* is the only component that adds a harmonic element to the performance (Bagchee, 1998). In Figure 2.1, we show a typical concert setup for performances in Carnatic music. Instruments are indicated by different numbers. We notice that in addition to the main percussion instrument in Carnatic music, *mṛdāngam*, there are two other percussion instruments, *kanjira* and *ghatam*.

Due to a wider geographical spread over the Indian subcontinent, Hindustani music is much more diverse and heterogeneous compared to Carnatic music. There are several

⁷<http://www.musicacademymadras.in/>



Figure 2.1: Example of a concert setup in Carnatic music. The numbers indicate different instruments: *mṛdaṅgam* (1), *kanjira* (2), *tānpura* (3), *ghatam* (4), violin (5), electronic *tānpura* (6), lead singer (Vignesh Ishwar) (7).

forms and styles within this music tradition such as *dhrupad*, *khyāl*, and *thumrī*. These forms differ considerably from each other and are characterized based on their singing styles and instrumentation (Bor et al., 2010). In this thesis, we primarily focus on the *khyāl* form, which is currently one of the most frequently performed forms in Hindustani music.

Despite a number of significant differences between Hindustani and Carnatic music, they share similar musical concepts. In both these music traditions, *rāga* (often pronounced as *rāg* in Hindustani music) is the melodic framework, and *tāla* (often pronounced as *tāl* in Hindustani music) is the rhythmic framework. For the rest of this document we will use the terms *rāga* and *tāla* to denote these concepts for both the music traditions. Since this thesis focuses on the melodic aspects of **IAM**, we provide a brief overview of the *rāga* concept in the subsequent section. We also provide a short explanation of the melody related terminology commonly used in **IAM**.

2.3.2 Melody in Indian Art Music

Melodies in **IAM** are based on the framework of *rāga*. It is one of the core musical concepts used in the composition, performance, music organization, and pedagogy of this music tradition (Bagchee, 1998; Danielou, 2010). Numerous compositions in Indian folk and film music are also based on *rāga* (Ganti, 2013).

Musicological literature on **IAM** is replete with the explanations and definitions of *rāga*. Some relevant ones are as follows.

“The rāga is more fixed than a mode, and less fixed than the melody, beyond the mode and short of melody, and richer both than a given mode or a given melody.” (Martinez, 2001, p. 96)

Which closely relates to

“A rāga is not a tune, nor is it a ‘modal’ scale, but rather a continuum with scale and tune as its extremes.” (Powers, 1959)

A more concrete definition, specifically from an analytical point of view, is given by:

“A rāga is most easily explained as a collection of melodic gestures, along with techniques for developing them. The gestures are sequences of notes that are often inflected with various micro-pitch alterations and articulated with an expressive sense of timing. Longer phrases are built by joining these melodic atoms together.”

(Chordia & Şentürk, 2013)

Overall, we see that rāga being more than a scale, mode and a discrete note sequence is highlighted in all the definitions. For a computational modeling of melodic aspects in IAM, the definition of rāga given by Chordia & Şentürk (2013) appears appropriate. Another way to comprehend the concept of rāga is to understand the melodic aspects that characterize rāgas, such as the set of svaras, ārohana-avrōhana, chalan and its characteristic melodic phrases. A brief description of these melodic aspects is provided in the subsequent paragraphs. For a more comprehensive description of these concepts we refer to Danielou (2010); Bagchee (1998); Viswanathan & Allen (2004).

Svaras: The seven solfège symbols (Sa, Re, Ga, Ma, Pa, Dha and Nī, in short-form) used in IAM are termed as svaras (Danielou, 2010; Bagchee, 1998). Except Sa and Pa (the fifth scale degree with respect to the base svara Sa), every other svara has two or three variations. In Table C.1 we provide a comprehensive list of all the svaras, their different variants, and the notations followed in Hindustani and Carnatic music to denote them. Every rāga comprises a set of typically five to seven svaras. Each of these svaras has a well defined functional role in the context of a given rāga (Viswanathan & Allen, 2004). Svaras in a rāga are structured hierarchically, where the vādi and the samvādi svaras are the first and the second most prominent svaras in a melody. Svaras also have some other functional roles such as serving as a nyās, which is discussed in the subsequent paragraphs.

Tonic pitch: The concept of tonic is fundamental to the melodic structures in IAM (Viswanathan & Allen, 2004; Danielou, 2010). It is the base pitch of a performer, carefully chosen in order to explore the full pitch range effectively in a given rāga rendition. In a performance of IAM, tonic pitch is constantly reinforced by the drone sound in the background that is typically generated by the tānpura. It acts

as a reference and the foundation for melodic integration throughout the performance (Deva, 1980). All the accompanying instruments such as the *sāraṅgi*, violin, *tablā* and *mr̥daṅgaṁ* are tuned using the tonic of the lead performer. It should be noted that the tonic pitch in IAM refers to a particular pitch value and not to a pitch-class. The frequency range of the tonic pitch for male and female singers spans more than one octave, roughly 100-260 Hz (Sengupta et al., 2005). In any performance of IAM, the tonic pitch is the *Sa* (short-form of *ṣadja*) svara around which the *rāga* is built upon (Danielou, 2010; Bagchee, 1998). Other set of *svaras* used in the performance derive their meaning and purpose in relation to this reference, and to the specific tonal context established by the given *rāga* (Deva, 1980).

Nyās: Dey (2008) presents various interpretations and perspectives on the concept of *nyās* in Hindustani music according to the ancient, medieval and modern authors. In the context of the current form of Hindustani music, the author describes *nyās* as that process in a performance of a *rāga* where an artist pauses on a particular svara, in order to build and subsequently sustain the format of a *rāga* (Dey, 2008, p. 70). The set of *svaras* on which the pause is permitted in a *rāga* grammar are referred to as *nyās svaras*. Every *rāga* comprises a set of *svaras* at which an artist can momentarily pause to release the musical tension built in the melody. Dey further elaborates the concept of *nyās* in terms of the action, subject, medium, purpose and effect associated with it. Typically, occurrence of *nyās svaras* mark the ending of a melodic phrase. A *nyās* is mostly manifested in a melody as a long held *svara*. However, there are several exceptions to it, which mainly depend on the local melodic context and the global tonal context defined by the *rāga*. *Rāga* grammar primarily acts as a guideline, and it does not explicitly define rules for the occurrences of *nyās* in a melody.

Ārōhana-Avrōhana: The ascending and descending progression of *svaras* in a melody of IAM are referred to as *ārōhana* and *avrōhana*, respectively. Every *rāga* has a defined constraint on how a melody can progress through its constituent *svaras*. Thus, *ārōhana*-*avrōhana* is a characterizing aspect of *rāgas*.

Gamaka And Alankār: One of the characterizing features of melodies in IAM is the presence of continuous gliding melodic gestures. Particular categories of these melodic gestures around *svaras* are commonly termed as *gamakas* (Krishna & Ishwar, 2012). A frequently occurring type of *gamaka* in Carnatic music is *kampitam*⁸, which involves an oscillatory pitch movement around a *svara* (Figure 2.2). Some other examples of *gamakas* include *odukkal*⁹ and *sphuritam*¹⁰, which are illustrated in Figure C.1b and Figure C.1a, respectively. Krishna & Ishwar (2012) present an insightful discussion on different *gamakas* in Carnatic music. There are various ways in

⁸ Audio: <https://www.freesound.org/people/sankalp/sounds/360771/>, Pitch: Figure 2.2

⁹ Audio: <https://www.freesound.org/people/sankalp/sounds/360770/>, Pitch: Figure C.1b

¹⁰ Audio: <https://www.freesound.org/people/sankalp/sounds/360769/>, Pitch: Figure C.1a

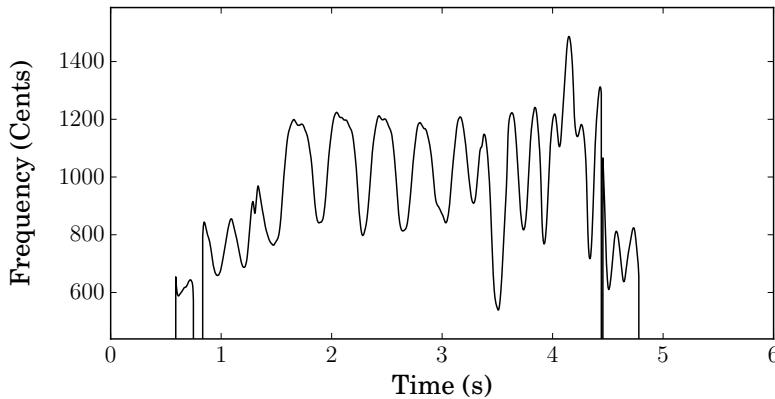


Figure 2.2: An example of the kampitam gamaka in Carnatic music.

which gamakas can be classified. A commonly found classification strategy describes 15 different types of gamakas in Carnatic music (Ramanathan, 1999; Janakiraman, 2008; Narayanaswami & Jayaraman, 2011). For a more detailed description of gamakas we refer to Narayanaswami & Jayaraman (2011). Note that gamakas are integral part of svaras in Carnatic music, and are not merely added ornamentations. Gamakas are also used in Hindustani music, though their usage and role is very different compared to that in Carnatic music.

There is another category of melodic gestures in Hindustani music called alankārs (literally meaning ornaments), which are primarily regarded as melodic ornamentations. There are different types of alankārs such as murkī, khatkā, kan-svara and mīnd. An example of the mīnd alankār¹¹, which involves a slow gliding pitch movement, is provided in Figure 2.3. A detailed description of these melodic gestures can be found in Bagchee (1998).

Characteristic Melodic Phrases (or rāga motifs): Every rāga has a set of characteristic melodic phrases (also referred to as pakads in Hindustani music) that capture the essence of the rāga (Bagchee, 1998; Rao et al., 1999; Viswanathan & Allen, 2004). These melodic phrases act as a building block to construct melodies. They provide a base for artists to express their creativity through improvisation within the rāga grammar. Characteristic melodic phrases are the most prominent cues used by human listeners for identifying rāgas (Krishna & Ishwar, 2012; Rao & Rao, 2014). In this thesis we also refer to these melodic phrases as rāga motifs.

Chalan: In addition to the characteristic melodic phrases discussed above, another important feature of a rāga is its chalan (literally meaning gait or movement) (Rao et al., 1999; Bagchee, 1998; Rao & Rao, 2014). It can be thought of as an abstraction

¹¹ Audio (13.5-18 s) <https://www.freesound.org/people/sankalp/sounds/360625/>, Pitch: Figure 2.3

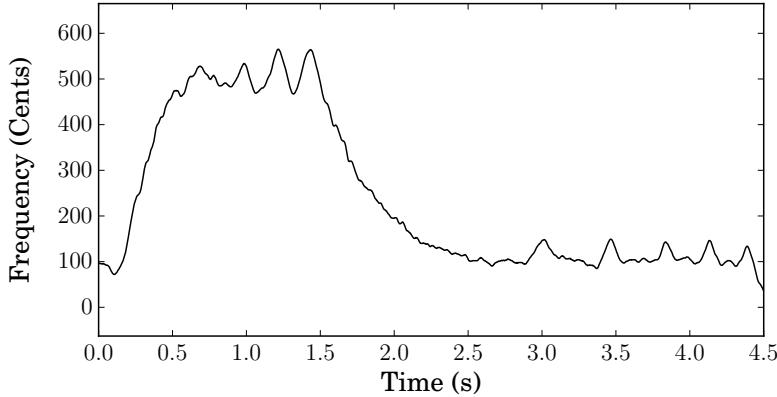


Figure 2.3: An example of the mīnd alankār in Hindustani music.

of the melodic phrases. The *chalan* defines the melodic outline of a *rāga*, that is, how a melodic transition is made from one *svara* to another, the precise intonation to be followed during the transition, and the proportion of time spent on each *svara*. In Figure 2.4, we illustrate this concept through an example. We show a segment of melody in *rāga bhīmapalāśī*¹² (Figure 2.4a) and in *rāga Bāgēśī*¹³ (Figure 2.4b). In both the cases, a melodic transition is made through the same set of *svaras* S, m, and g, in the same order. We notice that the characteristics of the transition regions in the melody and the duration of the *svaras* are different. These aspects of melodies are defined by the *chalan* of a *rāga*.

2.3.2.1 Allied Rāgas

Rāgas that share a common set of *svaras* and have a similar melodic phraseology are referred to as allied *rāgas* (Krishna & Ishwar, 2012). Typically, these *rāgas* are differentiated based on subtle melodic nuances and a set of melodic phrases that are specific to them (Meer, 1980, p. 74-76). Meer provides a detailed account of similarities across different sets of *rāgas* at various levels of their melodic organization. He also describes in depth the phenomenon of temporary alliance of *rāgas*. Some relevant parts of the text from his book are as follows:

“...Rāgas can be mixed to produce a new one, a common process. Some rāgas still retain the traits of the parent rāga although the older mixtures stabilize into a definite character...The ideal of mixing rāgas is to combine two rāgas with a similar mood, but with no more than one or two common points, so that the contrast between the rāgas can be maintained and their union is only temporary...It appears from the foregoing

¹²Audio: <http://www.freesound.org/people/sankalp/sounds/360427/>

¹³Audio: <http://www.freesound.org/people/sankalp/sounds/360428/>

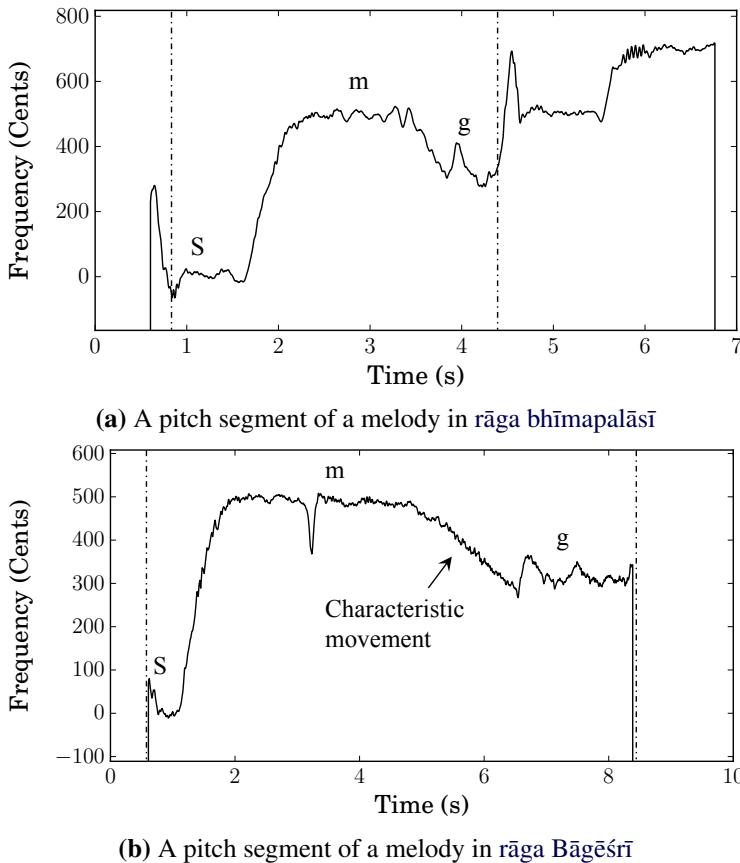


Figure 2.4: An example of a melodic movement through svaras S, m, and g, in two rāgas, bhīmapalāśī and Bāgēśī. This example illustrates the concept of chalan in Hindustani music, wherein the characteristics of the melodic movements are specific to the rāgas.

that some rāgas are closely related whereas others are more independent. Of the latter again there are varying degrees: some rāgas belong mainly to one rāga but have some flavour of another, other rāgas are quite separate, again others have their own strong identity with another rāga mixed in. A classification should indicate these degrees of interrelation..."

2.3.2.2 Recurring Melodic Units in Indian Art Music

As seen above, there are different kinds of melodic gestures and patterns that are used in the melodies of IAM. Factors such as the functional roles of these melodic patterns, the local melodic context, rāga grammar, and the artists' creativity determine the extent of their repetition and transformation in a rāga performance. Besides the gamakas, alankārs, and the rāga motifs, which are used across different forms and styles within these music traditions, there are also composition specific melodic pat-

terns. Mukhda of a song, which is the opening melodic line of a composition, is an example of such a pattern. A composition in IAM mainly provides a skeleton in a rāga performance, wherein melodic phrases such as the mukhdas of the composition are repeatedly used as anchor points to maintain a continual reference to the composition. In the scope of this thesis, we group different types of the melodic patterns discussed above into three main categories: gamaka type patterns, rāga motifs and composition specific patterns. Within the gamaka type patterns, we consider all the melodic patterns which are not specific to a given rāga or a composition. These patterns are used transversally. However, certain gamakas might bear a loose implicit relation with particular rāgas and their svaras. Second category of melodic patterns, the rāga motifs, include the characteristic melodic phrases of rāga. This is musically a well defined and clearly delimited melodic pattern category. Finally, as the third category, we consider patterns that are specific to a composition and are not used across different compositions (for example, mukhda of a song). Note that, several times music compositions use characteristic phrases of rāgas, but since these phrases are used across multiple compositions, we consider them within the rāga motifs category (Meer, 1980; Bagchee, 1998).

2.4 Related Work in Indian Art Music

We now proceed to present our review of the existing approaches for a number of computational tasks that are relevant to this thesis. In particular, we review literature on the three most relevant topics pertaining to the melodic analysis of IAM: tonic identification (Section 2.4.1), melodic pattern processing (Section 2.4.2), and rāga recognition (Section 2.4.3).

2.4.1 Tonic Identification

Identification of the tonic pitch of the lead artist in an audio recording is a crucial first step in tonal analysis of IAM (Section 2.3.2). A meaningful comparison of melodies across different artists and their recordings can be done by taking into account the tonal context established by the tonic. In this section, we present our review of the existing methods for tonic identification in audio recordings of IAM. Since one of our objectives in this thesis is to perform an extensive comparative evaluation of a number of these methods (Section 4.2), we review them in detail. In order to better interpret the results of the comparative evaluation, and to relate them with the processing steps and the parameter choices, we also provide the necessary implementation details, wherever required. The content of this section is taken mainly from our published study (Gulati et al., 2014a).

There have been various efforts to automatically identify the tonic pitch of the lead artist from an audio recording of IAM (Salamon et al., 2012; Gulati et al., 2012; Bellur et al., 2012; Ranjani et al., 2011; Sengupta et al., 2005; Chordia & Şentürk,

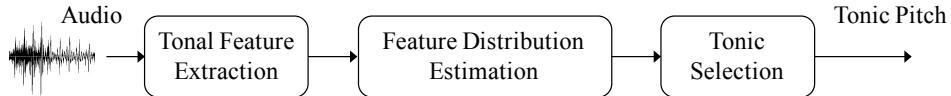


Figure 2.5: General block diagram of the processing steps used by the tonic identification approaches.

2013). These approaches mainly differ in terms of the musical cues that they utilize to identify the tonic, and the type of music material they are devised for (Hindustani or Carnatic music, vocal or instrumental music). Despite the differences, all these approaches can be divided into three main processing blocks, as shown in Figure 2.5. The only exception to this schema is the approach proposed by Sengupta et al. (2005).

In all the aforementioned approaches, the three main processing blocks are the following: feature extraction, feature distribution estimation and tonic selection. Since the task of tonic identification involves an analysis of the tonal content of the audio signal, the features extracted in the first block are always pitch related. In the second block, an estimate of the distribution of these features is obtained using either Parzen window based density estimation or by constructing a histogram. The feature distribution is then used in the third block to identify the tonic. The peaks of the distribution correspond to the most salient pitch values used in the performance (usually the *svaras* of the *rāga*), one of which corresponds to the tonic pitch. As the most salient peak in the distribution is not guaranteed to be the tonic, various techniques are applied to select the peak that corresponds to the tonic.

In Table 2.1, we provide a summary of the existing methods for tonic identification. The common processing blocks and the main differences between them become evident from this table. We now provide a brief description of each of these methods organized in terms of these processing blocks. A detailed review of these methods is also done in Gulati et al. (2014a). For a more detailed description of these methods we refer to their respective publications listed in Table 2.1.

2.4.1.1 Tonal Feature Extraction

In the tonal feature extraction block (Figure 2.5), the methods extract pitch-related features from the audio signal for further processing. With the exception of Salamon et al. (2012) and Gulati et al. (2012), all approaches use a single feature, the pitch in the audio. Salamon et al. (2012) uses a multipitch salience feature in order to exploit the tonal information provided by the drone instrument. Finally, Gulati et al. (2012) use both the multipitch salience feature and the predominant melody. Note that whilst pitch and fundamental frequency (f_0) are not the same (the former being a perceptual phenomenon and the latter a physical quantity), we use them interchangeably here.

We now provide an overview of the algorithms used by the different approaches mentioned above for extracting f_0 and the multipitch salience from audio recordings. Ran-

Method	Features	Feature Distribution	Tonic Selection
M _{RS} (Sengupta et al., 2005)	Pitch (Datta, 1996)	NA	Error minimization
M _{RH1} /M _{RH2} (Ranjani et al., 2011)	Pitch (Boersma & Weenink, 2001)	Parzen-window-based PDE	GMM fitting
M _{JS} (Salamon et al., 2012)	Multi-pitch salience (Salamon et al., 2011)	Multi-pitch histogram	Decision tree
M _{SG} (Gulati et al., 2012)	Multi-pitch salience (Salamon et al., 2011)	Multi-pitch histogram	Decision tree
M _{AB1} (Bellur et al., 2012)	Predominant melody (Salamon & Gómez, 2012)	Pitch histogram	Decision tree
M _{AB2} (Bellur et al., 2012)	Pitch (De Cheveigné & Kawahara, 2002)	GD histogram	Highest peak
M _{AB3} (Bellur et al., 2012)	Pitch (De Cheveigné & Kawahara, 2002)	GD histogram	Highest peak

ABBREVIATIONS: NA=Not applicable; GD=Group Delay, PDE=Probability Density Estimate

Table 2.1: Summary of the existing tonic identification approaches.

jani et al. (2011) use the Praat software¹⁴ to obtain the pitch contours (Boersma & Weenink, 2001). The software implements the algorithm by Boersma (1993), which is primarily proposed for speech signals and has also been used for monophonic music recordings in the past. Bellur et al. (2012) uses YIN, an average magnitude difference function (AMDF) based pitch estimation algorithm proposed by De Cheveigné & Kawahara (2002). YIN is mainly developed for speech signals. However, it has been used in a number of studies in MIR for pitch estimation from polyphonic music signals. Sengupta et al. (2005) use a method based on phase space analysis (PSA) proposed by Datta (1996) for extracting the f_0 from monophonic audio recordings.

One of the possible caveats of the aforementioned pitch (strictly speaking, f_0) estimation methods is that they are all primarily designed for monophonic signals containing a single sound source. This means that the number of estimation errors could increase as we add more instruments into the mixture. However, due to the heterophonic nature of IAM, and the prominent lead voice, monophonic pitch trackers often manage to detect the f_0 of the lead artist to an extent even in the presence of accompaniment instruments. One way of overcoming this problem is by using a predominant pitch estimation algorithm. Gulati et al. (2012) use the method proposed by Salamon & Gómez (2012) for estimating the pitch sequence of the predominant melody from the audio signal. Gulati et al. (2012) exploit the pitch information of the melody in the second stage of their approach to identify the specific octave of the tonic pitch (the tonic pitch-class is identified during the first stage of the algorithm).

As noted earlier, some proposed methods for tonic identification (Salamon et al., 2012; Gulati et al., 2012) use a multipitch approach. Instead of extracting the predominant melodic component from the audio signal, the methods compute a multipitch time-frequency representation of pitch salience over time (Salamon et al., 2011). The motivation for using multipitch analysis is twofold: first, as noted earlier, the music material under investigation is non-monophonic (includes many instruments playing simultaneously). Second, the tonic is continuously reinforced by the drone instrument, and this important cue can not be exploited by only extracting a single pitch value for each frame of the audio recording. To illustrate this point, in Figure 2.6 we display the spectrogram of a short audio excerpt of Hindustani music. Two types of harmonic series are clearly visible in the plot: the first consists of nearly straight lines and corresponds to the drone instrument (playing Sa and Pa). The second harmonic series (which start approximately at time 1 s) corresponds to the voice of the lead performer. Since the drone instrument is constantly present in the signal, a histogram of the peaks of the salience function will have prominent peaks at the pitches of the drone instrument, and this is exploited by Salamon et al. (2012) and Gulati et al. (2012) for identifying the tonic. The main difference between the two approaches is that whilst Salamon et al. (2012) directly identify the tonic pitch from the histogram, Gulati et al. (2012) divide the task into two stages: in first, the tonic pitch-class is

¹⁴Version 5.3.

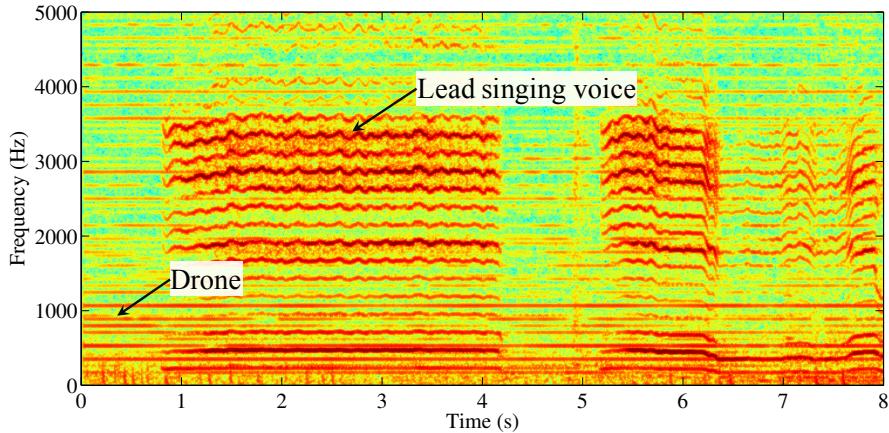


Figure 2.6: Spectrogram of an excerpt of Hindustani music with two clearly visible types of harmonic series, one belonging to the drone and the other to the lead voice.

identified using an extension of Salamon et al. (2012), and subsequently, the correct tonic octave is identified using the predominant melody information.

2.4.1.2 Feature Distribution Estimation

The tonal features extracted by the different tonic identification approaches are subsequently analyzed in a cumulative manner (cf. block two in Figure 2.5). The pitch values from all analysis frames (whether a single value is computed per frame or multiple values) are aggregated into a pitch distribution function, which reflects the (possibly weighted) rate of occurrence of different pitch values in the entire audio excerpt. The peaks of the pitch distribution function represent the most frequent (or salient if weighting is used) pitches in the recording, one of which is the tonic. The only exception is the approach proposed by Sengupta et al. (2005), which instead of analyzing the distribution of the features, computes an aggregate error function in order to select the tonic. The methods used by the different tonic identification approaches for estimating the pitch distribution function are described below.

In Salamon et al. (2012) and Gulati et al. (2012), the pitch values of the peaks of the salience function in every frame are aggregated into a histogram. The top 10 peaks in every frame are used, ensuring that in addition to the lead instrument/voice, the pitch content of other accompanying instruments is also captured, most importantly the notes played by the drone instrument. The frequency range considered for selecting the peaks of the salience function for constructing the histogram is restricted to 100-370 Hz (note that the typical frequency range for the tonic 100-260 Hz). The reason for computing the histogram beyond 260 Hz is that in some cases the aforementioned methods can exploit the presence of a peak corresponding to the fifth/-fourth (Pa/Ma) above the tonic in order identify the tonic pitch. Since in many cases

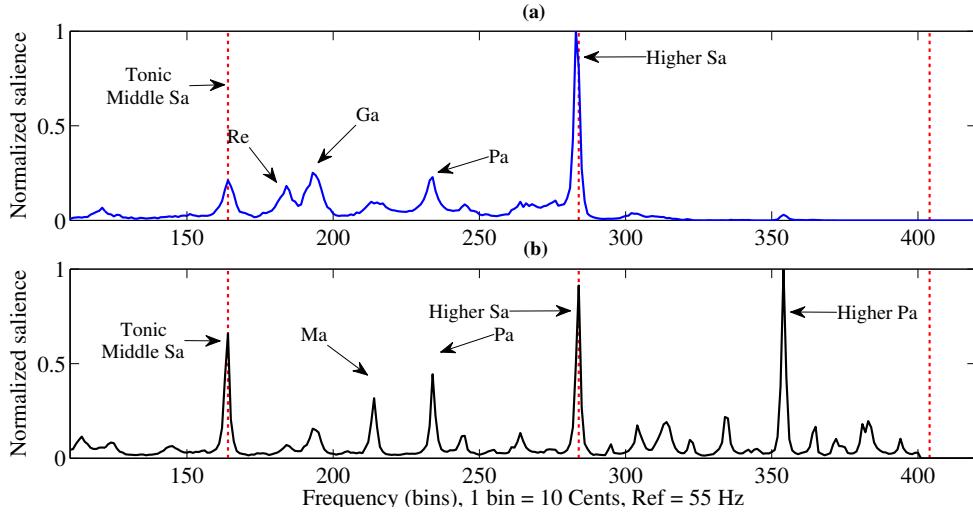


Figure 2.7: Pitch histograms for the same audio excerpt constructed using (a) the predominant melody (in blue) and (b) the peaks of a multipitch salience function (in black). The tonic pitch-class locations are indicated with red dotted lines.

the lead voice/instrument is considerably louder than the drone sound, the weights of the salience peaks are ignored when computing the histogram, meaning only the rate of occurrence is taken into account. As noted earlier, the result is that the pitches produced by the drone instrument (the tonic and Pa, Ma or Ni) manifest in the form of high peaks in the histogram, since the drone sounds continually in the recording. The resulting pitch distribution thus depends heavily on the notes of the drone instrument. This would not be the case if they considered the predominant melody for computing the histogram, in which case the pitch distribution would depend on the chosen rāga, thus increasing the complexity of identifying the tonic.

In Figure 2.7, we display two pitch histograms, computed using (a) the pitch of the predominant melody and (b) the peaks of a multipitch salience function. Both histograms are computed from the same three-minute audio excerpt. We see that in the histogram computed using the predominant melody (a), the prominent peaks correspond to svaras Sa, Ga and Re (the prominent svaras of rāga Sindh Bhairavī), whereas in the multipitch histogram (b), the top three peaks correspond to Sa (in two octaves) and Pa, which are the prominent svaras produced by the drone instrument.

In Bellur et al. (2012), a histogram is constructed using a frequency range of 40–800 Hz with a 1 Hz resolution and later post processed using a group delay (GD) function. The authors show that by assuming that the constructed pitch histogram is the squared magnitude of resonators in parallel, group delay functions can be applied to obtain a better resolution for the peaks in the resulting histogram. It is also shown that a group delay function accentuates peaks with lesser bandwidths. Given that

the Sa and Pa svaras in all the octaves are relatively less inflected, this characteristic of the **GD** function is shown to be beneficial for improving the accuracy of tonic identification. The processed histograms are referred to as **GD** histograms.

Bellur et al. (2012) also propose the concept of segmented histograms. In order to exploit the omnipresence of the Sa *svara*, the authors propose to segment the pitch contour into smaller units, and construct a **GD** histogram for each of these units. Given that Sa is likely to be present in all the units, the corresponding peak will be enhanced in the **GD** histograms. The individual histograms are then multiplied bin-wise. The authors report that this helps in reducing the salience of the other svaras which might not be present in all the segments. Tonic selection is then performed on the resulting histogram, referred to as the segmented **GD** histogram.

Instead of using a histogram, Ranjani et al. (2011) use a Parzen window estimator to compute a pitch density function (Bishop, 2006; Duda et al., 2000). Parzen window estimators (or kernel density estimators) are non-parametric density estimators. The choice of kernel function can control the smoothness of the estimated density. They are widely used as an alternative to histograms to alleviate the artificial discontinuities at the boundaries of the bins of the histogram, and thus, aid in peak picking process. In addition, they do not require partitioning of data into distinct bins. The authors use Parzen window estimators with Gaussian kernels for estimating the density of the extracted pitch frequencies.

2.4.1.3 Tonic selection

In this section, we describe the last processing block shown in Figure 2.5, where the pitch distribution function is used to identify the tonic pitch. The peaks of the pitch distribution function correspond to the most frequent (or salient) pitches present in the audio signal. Depending on how the pitch distribution is computed, the peaks either coincide with the *svaras* of the *rāga* used in the rendition or with the *svaras* produced by the drone instrument. The problem of tonic identification is thus reduced to selecting the peak of the distribution that corresponds to the tonic of the lead artist. As noted earlier, the peak corresponding to the tonic pitch is not always the highest peak in the distribution. For this reason, various strategies have been proposed for analyzing the pitch distribution and selecting the peak that corresponds to the tonic. The complexity of the approaches varies from simply selecting the highest peak of the histogram to the application of machine learning algorithms in order to automatically learn the best set of rules for selecting the tonic peak. The different tonic selection strategies used in the literature are presented below.

Ranjani et al. (2011) model the pitch distribution using semi-continuous Gaussian mixtures (Huang et al., 2001), motivated by the following two musical cues in IAM: first, the relative positions of the *svaras* with respect to the tonic hover around a mean ratio (Krishnaswamy, 2003b) and second, the Sa and Pa are the prakrthi *svaras*, which means they are sung/played without any inflections (Manikandan, 2004; Krish-

naswamy, 2003a). Peaks of the pitch density function within a suitable pitch range are selected as possible tonic candidates. The variance of the pitch distribution around the peaks is estimated by modeling each tonic candidate (i.e. a peak) with a Gaussian distribution. As noted above, one of the key characteristics of the Sa and Pa *svaras* is that they do not contain pitch inflections. Thus, the parameters of the Gaussian model are used to infer the correct tonic candidate. The pitch range used for selecting tonic candidates is 100-250 Hz. When the editorial metadata of the audio recording is known, the pitch range is further constrained depending on the gender of the lead artist. The range for male singers is set to 100-195 Hz and for female singers is set to 135-250 Hz.

Salamon et al. (2012) use a classification based approach to identify the peak of the multipitch histogram that corresponds to the tonic pitch (or tonic pitch-class in the latter). Since all the pitches in a performance are in relation to the tonic, the relationships between the peaks of the histogram (height and distance) are used to compute a set of features, which are then used to train a classifier for identifying which peak corresponds to the tonic. In this way, rather than having to manually define a template for selecting the tonic, an optimal set of rules are learned automatically using machine learning. The authors extract height and distance related features for the top 10 peaks in the multipitch histogram. The authors show that for the tonic identification task the C4.5 decision tree classifier (Quinlan, 1993) yields the highest classification accuracy.

Gulati et al. (2012) use a similar classification-based approach to first identify the peak of the histogram that corresponds to the tonic pitch-class. The correct tonic octave is then determined in the second stage of processing, which is also classification-based. For every candidate tonic pitch (candidates have the same pitch class but are in different octaves) a set of 25 features is computed. The features are the values of the melody histogram at 25 equally spaced locations spanning two octaves centered around the tonic pitch candidate. The classification task is a two-class problem, whether or not the tonic candidate is in the correct octave. As done in Salomon et al. (2012), a C4.5 decision tree is trained using the Weka data-mining software for the classification. For a detailed description of the method we refer to Gulati (2012).

Sengupta et al. (2005) use an error minimization technique to identify the tonic. They employ a brute force approach in which a large number of pitch values within a pre-defined frequency range are considered as candidates for the tonic pitch. A cumulative deviation is computed between the steady state regions of the pitch contour and the pitch values of the closest *svaras* to these regions, which are obtained using three different tuning schemas given a tonic candidate. The tonic candidate which results in the minimum deviation is selected as the tonic of the musical excerpt.

Bellur et al. (2012) propose a simple approach, picking the highest peak of the pitch distribution as the tonic. In two out of the three proposed variants of their method, the bin value of the highest peak of the segmented GD pitch histogram is selected as the tonic pitch. The frequency range of the histogram is restricted to 100-250 Hz. When

the gender information of the lead artist for an audio recording is available, this range is further restricted. In addition to the simple highest peak approach, Bellur et al. (2012) also propose a template matching process to identify the tonic. This procedure is comparable to the semi-continuous GMM fitting proposed by Ranjani et al. (2011), which exploits the smaller degree of pitch variation around the Sa and Pa svaras. The template that the authors use to identify the tonic candidate uses three octaves and consider the pitch distribution values at tonic and its fifth in different octaves. For more information we refer to Gulati et al. (2014a).

As we see from this review on the task of tonic identification, a variety of methods are proposed in the literature. These methods vary considerably within each processing block of the task. Many of these studies show promising results (above 90%). However, since they are evaluated on different datasets with different sets of measures and evaluation setup, they cannot be directly compared. We note that in none of these studies an attempt is made to compare the performance with other studies. In order to deeply understand the strengths and shortcomings of these approaches on different types of music material it is essential that they are compared under the same experimental setup and using the same music collection.

2.4.2 Melodic Pattern Processing

In this section, we review the existing approaches for melodic pattern processing in audio collections of IAM. By melodic pattern processing we refer to a number of tasks that involve the computational analysis of melodic patterns such as pattern similarity, pattern detection, and pattern discovery. Analysis of melodic patterns is a well studied research task in MIR and computational musicology (Section 2.5.2). However, for IAM, despite the importance of melodic patterns in the *rāga* framework, this task has gained attention only recently, mainly during the course of this dissertation.

In Table 2.2, we summarize the existing approaches for pattern processing in IAM and provide relevant details to better compare these approaches. We see that there are three closely related but different pattern processing tasks that these approaches address: 1) Pattern detection, where given a query melodic pattern the objective is to retrieve its other occurrences in the test audio recordings (Ross & Rao, 2012; Ross et al., 2012; Ishwar et al., 2013; Dutta & Murthy, 2014b; Ganguli et al., 2015), 2) Pattern distinction, where given a query pattern the objective is to retrieve its other instances from a pool of annotated melodic patterns (Ishwar et al., 2012; Rao et al., 2013, 2014), 3) Pattern discovery, where given a collection of music recordings the objective is to discover melodic patterns in the absence of any ground truth annotations of the melodic patterns (Dutta & Murthy, 2014a). The task of pattern distinction can be considered as a subtask of pattern detection, where the differences being the absence of the irrelevant patterns in the search space, and the use of pre-segmented melodic patterns. We differentiate between these two tasks in this review because the complexity involved in these tasks is considerably different. Also, the approaches

Method	Task	Melody Representation	Segmentation	Similarity Measure	Speed-up	#Rāgas	#Rec	#Patt	#Occ
Ishwar et al. (2012)	Distinction	Continuous	GT annotations	HMM	-	5 ^c	NA	6	431
Ross & Rao (2012)	Detection	Continuous	Pa nyās	DTW	Segmentation	1 ^h	2	2	107
Ross et al. (2012)	Detection	Continuous, SAX-12,1200	Sama location	Euc., DTW	Segmentation	3 ^h	4	3	107
Ishwar et al. (2013)	Detection	Stationary point, Continuous	Brute-force	RLCS	Two stage method	2 ^c	47	4	173
Rao et al. (2013)	Distinction	Continuous	GT annotations	DTW	-	1 ^h	8	3	268
Dutta & Murthy (2014a)	Discovery	Stationary point, Continuous	Brute-force	RLCS	-	5 ^c	59	-	-
Dutta & Murthy (2014b)	Detection	Stationary point, Continuous	NA	Modified- RLCS	Two stage method	1 ^c	16	NA	59
Rao et al. (2014)	Distinction	Continuous	GT annotations	DTW	-	1 ^h	8	3	268
Ganguli et al. (2015)	Detection	BSS, Transcription	-	Smith- Waterman	Discretization	34 ^h	50	NA	1075

^b Hindustani music collection^c Carnatic music collection

ABBREVIATIONS: #Rec=Number of recordings; #Patt=Number of unique patterns; #Occ=Total number of annotated occurrences of the patterns; GT=Ground truth; NA=Not available; “_”=Not applicable; Euc.=Euclidean distance.

Table 2.2: Summary of the methods proposed in the literature for melodic pattern processing in IAM. Note that all of these studies were published during the course of our work.

proposed for pattern distinction do not address the issues related with the computational complexity, a challenge often encountered in the task of pattern detection and discovery. Note that, there are a few studies done on detection and modeling of specific melodic ornaments in Hindustani and Carnatic music (Subramanian et al., 2012; Datta et al., 2007; Narayan & Singh, 2014; Pratyush, 2010). We do not consider these studies in the current review since they focus on a particular type of short-duration melodic ornament (or gesture). Moreover, the methodology used in some of these approaches is already covered in our review of the other methods.

As seen from Table 2.2, a majority of the existing methods follow a supervised approach and focus on the task of pattern detection or pattern distinction. This can be attributed to the challenges involved in the task of pattern discovery, specifically in terms of the computational complexity. In addition, since this research topic has recently gained attention in MIR for IAM, the primary focus of the approaches in the beginning is to devise meaningful melodic similarity model. Based on the description of these approaches there are three main processing units involved in this task: melody representation, melody segmentation and similarity (or dissimilarity) computation. There is often an interplay between the choices made within these three units, specifically between the melody representation and the similarity measure. As it is also highlighted in Gómez et al. (2003), pitch and timing variations across occurrences of the melodic patterns can be either handled in the melody representation or during the computation of the melodic similarity. In the former case, a generic or a musically agnostic distance measure might be sufficient for the computation of melodic similarity, whereas, in the latter, a music specific distance measure that can incorporate domain knowledge and can handle timing and pitch variations is required.

We first review the existing approaches in terms of the three processing units mentioned above. From Table 2.2, we notice that with only a couple of exceptions (Ross et al., 2012; Ganguli et al., 2015) all other approaches work with a fine grained continuous melody representation. This to a large extent is attributed to the characteristics of the melodies in IAM, due to which the extraction of a reliable symbolic or discrete melody representation becomes a challenging task (Widdess, 1994). Moreover, the transitory melodic regions between the *svaras* in a melody are found to be important in the computation of melodic similarity (Datta et al., 2007; Gupta & Rao, 2012), which are lost in a simple transcription of melodies. Ross et al. (2012) and Ganguli et al. (2015) examine the affect of abstracting the melody representation by using techniques such as symbolic aggregate approximation (SAX) (Lin et al., 2003) and behavioral symbol sequence (BSS) (Tanaka et al., 2005). Ganguli et al. (2015) also propose a heuristic-based pseudo melody transcription approach to obtain a discrete melody representation. It was found that these abstracted representations reduce the computational cost by a significant factor. However, their accuracy remains inferior compared to a continuous melody representation (considering the best performing distance measure for both the types of melody representations) (Ross et al., 2012; Ganguli et al., 2015). Moreover, these discrete representations are evaluated on a small

dataset comprising a specific style of singing within Hindustani music. Therefore, the applicability of such abstracted melody representations to all types of melodic styles in both Hindustani and Carnatic music is questionable. Ishwar et al. (2013), Dutta & Murthy (2014a) and Dutta & Murthy (2014b) use an abstracted melodic representation that exploits specific melodic characteristics of Carnatic music. To represent a melody the authors consider only the stationary points (where the slope becomes zero) of a continuous melody representation. However, such a representation is too coarse to compute a reliable melodic similarity, and therefore, it is primarily used to prune the search space in order to reduce the computational cost. The final computation is done by using a continuous melody representation. Overall, we see that devising a melody representation that can encapsulate and abstract melodic characteristics to aid the computation of melodic similarity is a challenging task in IAM. We also notice that a continuous melody representation, which places minimal assumptions on the melodic style, appears to be the most versatile representation.

As noted before, an important aspect in pattern detection is melody segmentation. While there are well studied models for melody segmentation in symbolic representation of music (Cambouropoulos, 2006; Rodríguez L. et al., 2014; Bozkurt et al., 2014), to the best of our knowledge, segmentation models for IAM in audio recordings are nonexistent. As a result of which, approaches for pattern detection in IAM either tend to use a brute force segmentation strategy or use a local alignment-based distance measures that do not require an explicit melody segmentation (Table 2.2). Ross & Rao (2012) and Ross et al. (2012) detect specific rhythmic and melodic landmarks (*sama* locations and *nyās svara* onsets) in the audio recordings to determine the location of the potential melodic pattern candidates. However, these approaches are very specific to a certain type of melodic patterns, musical style, and to only slow tempo (*vilambit lay*) music compositions. For example, *sama* location can indicate roughly the onset of a *mukhda* phrase in a recording of Hindustani music. But, it has no definite relationship with the location of the characteristic melodic phrases of *rāgas*. Similarly, the *Pa nyās* segmentation strategy followed by Ross & Rao (2012) can work only with the melodic phrases ending in the *Pa svara*, and mainly for slow tempo compositions where the concept of *nyās svara* is prominently present. Moreover, detecting these landmarks in itself is a challenging task (Srinivasamurthy & Serra, 2014; Gulati et al., 2014b). Thus, these approaches might not generalize and scale to other types of melodic patterns and to large music collections. Overall, we see that there is a lack of phrase-level segmentation models for melodies in IAM.

Melodic similarity (or dissimilarity) measure is another crucial block in the melodic pattern processing tasks. From Table 2.2, we see that a majority of the approaches use a dynamic programming-based similarity measure. Ross & Rao (2012); Ross et al. (2012); Rao et al. (2013, 2014) use a similarity measure based on different variants dynamic time warping (DTW), Ishwar et al. (2013); Dutta & Murthy (2014a,b) use a rough longest common subsequence (RLCS)-based similarity measure, and Ganguli et al. (2015) employ Smith-Waterman algorithm to compute melodic similar-

ity (Smith & Waterman, 1981). The dominance of dynamic programming-based similarity measures can be attributed to the fact that the melodic patterns in IAM undergo a large degree of non-linear timing variations, which can further be attributed to the improvisatory nature of this music tradition. Computing sequence similarity without any temporal alignment, such as in the Euclidean distance, falls short of measuring a meaningful melodic similarity in IAM (Ross et al., 2012). Although, a thorough comparison of the Euclidean distance with the dynamic programming-based similarity measures for the same melody representation is lacking in the literature. Some of the existing studies also propose enhancements to the well-known distance measures. Dutta & Murthy (2014b) propose to modify the intermediate steps involved in the computation of the RLCS distance to make it more suitable to melodic sequences. These modifications are reported to result in an improvement in the precision of the system, while maintaining the same recall. However, the study is conducted using only 59 pattern instances in 16 excerpts belonging to only one *rāga*. Rao et al. (2014) propose to learn an optimal shape of the global path constrained applied in the DTW-based distance measure. However, as reported by the authors, the learned global constraint degraded the performance of the method. Moreover, since the constraint learning is performed for a particular pattern category, such a technique is not applicable to an unseen data, which is the case in the task of pattern discovery. In contrast to these time-series matching-based approaches, there are a few approaches that use statistical pattern matching paradigms. Ishwar et al. (2012); Rao et al. (2014) consider this task similar to that of a keyword spotting in speech, and use hidden markov models (HMMs) to essentially perform pattern classification. The evaluations of the HMM-based system show promising results. However, the authors address a relatively easier task of pattern distinction, where the search space did not contain any irrelevant pattern candidates. Moreover, since there was no baseline system considered in the evaluations, a comparison of the HMM-based approach to the other methods is left to a comparative evaluation.

We now compare the existing approaches in terms of the other relevant aspects involved in the task of melodic pattern processing. An important consideration in pattern detection and discovery task for sizable datasets is the computational complexity. Since the similarity measures that perform well, as noted above, are mainly based on dynamic programming, computational complexity becomes even a bigger concern. As seen from Table 2.2, not every approach addresses this issue. This can be due to the small size of the datasets used to evaluate the approaches, for which these systems are not computationally intractable. Ganguli et al. (2015) improve the computational efficiency of their method by using an abstracted low-dimensional discrete melody representation. However, as mentioned before, the performance of such a system is inferior to that using a continuous melody representation, and the scalability of such an approach to diverse music material is questionable. Furthermore, the accuracy of such an approach is always limited by the performance of the melody transcription system. Another type of optimization is to perform the task of pattern detection in

two stages as proposed by Dutta & Murthy (2014b); Ishwar et al. (2013). In the first stage a coarse melody representation can be used to identify the regions in the audio recordings that are more likely to contain the relevant patterns. Such a coarse melody representation drastically reduces the computational cost involved in the operation. Once the search space is pruned, in the second stage, a fine grained continuous melody representation is used to reliably detect the pattern occurrences. The coarse melody representation proposed by Ishwar et al. (2013) is mainly applicable to Carnatic music (and not to Hindustani music) as it utilizes the presence of *gamakas*. Besides, this method does not compute a theoretical lower-bound, and the pruning is based on an empirically determined threshold. This means that it is not very suitable for tasks such as pattern discovery, where determining a musically relevant similarity threshold in itself is a challenge. Furthermore, the first stage of this method does not result in a perfect (100%) recall, and therefore, it can potentially become a bottleneck of the system. Some other proposals for reducing the computational complexity involve top-down segmentation methods that exploit the specific type of melodic and rhythmic landmarks such as *sama* and *nyās* onsets (Ross & Rao, 2012; Ross et al., 2012). Occurrences of a certain types of melodic phrases are marked by these events. However, as explained earlier, such an approach makes the system tuned to a particular type of melodic patterns and musical forms within **IAM**. Overall, we notice that the existing approaches do not utilize any generalizable algorithms to reduce the computational complexity of the task.

Based on our review, we see that there are several avenues for improvement and scientific contributions on this research topic, one of which is the evaluation setup. Apart from the fact that the evaluation setup is considerably different across studies, even within a study it needs to improve to generate reliable results. Most of the approaches are evaluated with only a few recordings belonging to a handful of *rāgas*. Thus, the dataset size and diversity is one of the major limitations of the experimental setup used in the existing studies. The evaluation measures used by a number of these approaches are ad-hoc. For example, Ganguli et al. (2015) uses only the first five phrases taken from the starting of the recordings to evaluate the system, though, the other annotated pattern instances when retrieved are regarded as the true hits. Thus, not considering all the annotated pattern instances as a query creates an evaluation bias. Another example is in Ishwar et al. (2013), where the authors measure precision, recall and F-measure using only the top ten retrieved results, wherein the number of relevant patterns in the system are of the order of 100. Also, the procedure followed to calculate these measures in such a specific scenario is not described in the article. Thus, there is a need to promote and use well established evaluation metrics typically used in information retrieval tasks to make the results more comparable.

An observation that can be made from Table 2.2 is that none of the existing approaches are evaluated on both Hindustani and Carnatic music collections. The melodic characteristics between these two music traditions vary considerably. Thus, evaluating the same system with both the music traditions can provide new insights in to the short-

comings and strengths of the different melodic representations and similarity measures. Another observation is that there is only one method that addresses the task of pattern discovery (Dutta & Murthy, 2014a). This method is shown to work with short duration (nearly 15 s) audio excerpts comprising the first line of the compositions, which are specifically recorded and segmented for the study. Scalability of such an approach to hundreds of hours of music collections is questionable. Understandably, pattern discovery is a complex and a computationally expensive task compared to pattern detection. Nonetheless, there should be more research efforts directed towards it, especially since the size of the datasets used in the supervised approaches tends to be small owing to the challenges involved in the annotation process.

Based on our discussion above, we see that there are a variety of approaches proposed for melodic pattern processing in **IAM**. However, a direct comparison across these approaches cannot be made, since these approaches are evaluated on different experimental setup. A systematic improvement in these approaches calls for a comprehensive comparative evaluation to understand their limitation and strengths for different type of music content within **IAM**. Apart from comparing the existing approaches directly, there are several other aspects that require a systematic evaluation. For example, an optimal sampling rate is an important consideration in a continuous melody representation. But, despite the fact that most of the approaches use such a representation, they do not study this aspect. Similarly, none of the approaches except Rao et al. (2014) address the issue of pitch octave transposition, which is frequently encountered across repeated instances of melodic patterns. Other issues regarding the redundancy filtering and the characterization of the discovered melodic patterns are also to be addressed. Thus, we see that there is a large scope of improvement in the computational methodologies addressing pattern processing tasks in **IAM**.

2.4.3 Rāga Recognition

Recognizing automatically the *rāga* of a given audio recording is one of the most important tasks in melodic description of **IAM** (Section 2.3.2). In this section, we present a review of the existing computational approaches for this task. Our main objective in this review is to consolidate existing work on this topic, highlight the strengths and shortcomings of different methodologies, and finally, identify potential venues of improvement.

The accuracy of a *rāga* recognition system largely depends on the size of the test dataset in terms of the number of *rāgas* and the chosen set of *rāgas*. The task becomes more challenging when the chosen set of *rāgas* in the dataset share a common set of *svaras* and are allied *rāgas* (Section 2.3.2.1). Since the approaches reviewed in this section are evaluated on different datasets, we refrain from comparing the absolute *rāga* recognition accuracies across studies.

We now proceed to review the existing *rāga* recognition approaches. In Table 2.3 and Table 2.4, we summarize the prominent existing approaches for *rāga* recognition.

Methods	Svara set	Svara salience	Svara intonation	ārōhana-avrōhana	Melodic phrases	Svara discretization	Temporal aspects
Pandey et al. (2003)					•	•	Yes
Chordia & Rae (2007)	•	•		•	•	•	•
Belle et al. (2009)	•	•	•	•	No		
Shetty & Acharya (2009)	•	•	•	•	Yes		
Sridhar & Geetha (2009)	•	•	•	•	Yes	•	•
Koduri et al. (2011)	•	•	•	•	Yes		
Ranjani et al. (2011)	•	•	•	•	No		
Chakraborty & De (2012)	•	•	•	•	Yes		
Koduri et al. (2012)	•	•	•	•	Both		
Chordia & Şentürk (2013)	•	•	•	•	No		
Dighe et al. (2013a)	•	•	•	•	Yes	•	
Dighe et al. (2013b)	•	•	•	•	Yes		
Koduri et al. (2014)	•	•	•	•	No		
Kumar et al. (2014)	•	•	•	•	Yes	•	
Dutta et al. (2015) ^a	•	•	•	•	No	•	

^aThis method performs rāga verification and not recognition

Table 2.3: Rāga recognition methods proposed in the literature along with the melodic characteristics they utilize to perform the task. We also indicate if a method uses a discrete svara representation of melody. The methods are arranged in chronological order.

Method	Tonal Feature	Tonic Identification	Feature	Recognition Method	#Rāgas	Dataset (Dur./Num.) ^a	Audio Type
Pandey et al. (2003)	Pitch (Boersma & Weenink, 2001)	NA	Svara sequence	HMM and n -Gram	2	- / 31	MP
Chordia & Rae (2007)	Pitch (Sun, 2000)	Manual	PCD, PCDD	SVM classifier	31	20 / 127	MP
Belle et al. (2009)	Pitch (Rao & Rao, 2009)	Manual	PCD (parameterized)	k -NN classifier	4	0.6 / 10	PP
Shetty & Achary (2009)	Pitch (Sridhar & Geetha, 2006)	NA	#Svaras, Vakra svaras	Neural Network classifier	20	- / 90	MP
Sridhar & Geetha (2009)	Pitch (Lee, 2006)	Singer identification	Svara set, its sequence	String matching	3	- / 30	PP
Koduri et al. (2011)	Pitch (Rao & Rao, 2010)	Brute force	PCD	k -NN classifier	10	2.82 / 170	PP
Ranjanji et al. (2011)	Pitch (Boersma & Weenink, 2001)	GMM fitting	PDE	SC-GMM and Set matching	7	- / 48	PP
Chakraborty & De (2012)	Pitch (Sengupta, 1990)	Error minimization	Svara set	Set matching	-	- / -	-
Koduri et al. (2012)	Predominant pitch (Salamon & Gómez, 2012)	Multipitch-based	PCD variants	k -NN classifier	43	- / 215	PP
Chordia & Sentürk (2013)	Pitch (Camacho, 2007)	Brute force	PCD variants	k -NN and statistical classifiers	31	20 / 127	MP
Dighe et al. (2013a)	Chroma (Lartillot et al., 2008)	Brute force (vādi-based)	Chroma, Timbre features	HMM	4	9.33 / 56	PP
Dighe et al. (2013b)	Chroma (Lartillot et al., 2008)	Brute force (vādi-based)	PCD variant	RF classifier	8	16.8 / 117	PP
Koduri et al. (2014)	Predominant pitch (Salamon & Gómez, 2012)	Multipitch-based	PCD (parameterized)	Different classifiers	45*	93 / 424	PP
Kumar et al. (2014)	Predominant pitch (Salamon & Gómez, 2012)	Brute force	PCD + n -Gram distribution	SVM classifier	10	2.82 / 170	PP
Dutta et al. (2015)*	Predominant pitch (Salamon & Gómez, 2012)	Cepstrum-based	Pitch contours	LCS with k -NN	30†	3 / 254	PP

^a In the case of multiple datasets we list the larger one * This method performs rāga verification and not recognition

* Authors do not use all 45 rāgas at once in a single experiment, but consider groups of 3 rāgas per experiment † Authors finally use only 17 rāgas in their experiment
ABBREVIATIONS: Dur.: Duration of the dataset, Num.: Number of recordings, NA: Not available, -: Not available, SC-GMM: semi-continuous GMM, MP: Monophonic, PP: Polyphonic

Table 2.4: Summary of the Rāga recognition methods proposed in the literature. The methods are arranged in chronological order.

Both the tables comprise exactly the same set of approaches, but differ in terms of the type of information provided for each approach. In Table 2.3, we indicate the different characteristic features of a *rāga* that these approaches exploit to perform the task. The melodic attributes that we have considered in the summary are: *svara* set, *svara* salience, *svara* intonation, *ārōhana-avrohāna* and melodic phrases. In addition, to get a better understanding of these approaches, we also mark if an approach uses a discretized melodic representation, and if it considers the temporal aspects of the melody to perform the task. In Table 2.4, we provide the relevant details for each approach in terms of the common processing blocks such as the feature extraction, the tonic identification, the learning method used to recognize *rāgas*, and the other relevant dataset details. With both these tables we get a bird's-eye view of the existing work done in *rāga* recognition.

From Table 2.3, we see that the most frequently used melodic attribute is the set of *svaras* in a *rāga*, which is also computationally one of the most basic features to extract. *Svara* set is considered as a feature for *rāga* recognition in a both explicit and implicit manner by different approaches. Chakraborty & De (2012); Ranjani et al. (2011) explicitly extract the comprising set of *svaras* in an audio recording. The *rāga* of the recording is then identified by matching the estimated *svara* set with the stored set for each *rāga*. The exact procedure followed to map the estimated *svara* set with a unique *rāga* label is missing in the articles. As can be imagined, it is a rather naive approach since there are several *rāgas* that share the same set of *svaras* and are differentiated based on more elaborate melodic and temporal characteristics.

Along with the set of *svaras*, *rāga* grammar also defines the functional roles of these *svaras* (Section 2.3.2). In particular, there is a vocabulary to describe the salience of the *svaras* in a melody such as *vādi* and *saṁvādi* (the two most salient *svaras* in a melody). Thus, one of the ways to differentiate between the *rāgas* that share a common set of *svaras* is to also consider the salience of the comprising *svaras* in the analysis. Computing *svara* salience for all the possible *svaras* frequencies implicitly incorporates the *svara* set feature as well. This is the reason why this feature appears for nearly all the approaches mentioned in Table 2.3. Chordia & Rae (2007) propose a feature that incorporates the salience of different *svaras* in a melody for recognizing *rāgas*. The authors represent *svara* saliences using a 12 bin pitch-class distribution (PCD) computed as a histogram of the pitch sequence. This global feature is robust to pitch octave errors and is shown to perform well on a sizable dataset. Note that the approach proposed by Chordia & Rae (2007) for computing PCD implicitly considers the duration of the *svaras* in a melody for estimating their salience. However, the exact meaning of the salience of a *svara* in a melody is not explicitly defined in the music theory, and therefore, it can be interpreted and computed in multiple ways. Koduri et al. (2011) explore two different approaches for computing PCD, which differ in the way *svara* salience is interpreted. One of their proposed approaches weighs the salience by the duration of the *svaras* as also done in Chordia & Rae (2007). The other approach considers the frequency of occurrence of the *svaras* as their salience,

irrespective of their duration. The former approach was reported to obtain a better accuracy.

A simple extension to the 12 bin PCD feature mentioned above is computing the pitch distribution using a fine grained bin boundary. A high resolution PCD in addition to the svara saliences also captures the intonation aspects of the svaras. Such a fine grained PCD is used in Chordia & Şentürk (2013); Koduri et al. (2012); Belle et al. (2009); Kumar et al. (2014). These studies report a superior performance by using the high resolution PCD as compared to a 12 bin PCD. Note that in Chordia & Şentürk (2013) the authors refer to the high resolution PCD by fine-grained pitch distribution (FPD). Apart from the high resolution PCDs, there are other variants of the PCD feature, wherein the technique used for computing the pitch distribution is based on the concept of kernel density estimation (KDE). Such a variant of the PCD feature is used in Chordia & Şentürk (2013); Ranjani et al. (2011) and is reported to further improve the rāga recognition accuracy. These variants are referred to as kernel-density pitch distribution (KPD) in Chordia & Şentürk (2013) and probability density estimate (PDE) in Ranjani et al. (2011).

The high resolution PCD feature mentioned above implicitly captures some of the intonation aspects of the svaras in a melody. However, controlling the importance of the specific intonation aspects in the PCD feature space for rāga recognition is a challenging task. To address this issue, Belle et al. (2009); Koduri et al. (2014) propose to use a parametrized version of the PCD, wherein the parametrization is performed for each svara in the melody. Belle et al. (2009) extract four different features for each svara: the peak position, mean position, variance and overall probability of the svara. In a similar manner, Koduri et al. (2014) extract six features for each svara: the peak position, peak amplitude, mean, variance, skewness and kurtosis of the svara distribution. Melodies in Carnatic music contain gamakas (Section 2.3.2), during which the pitch deviation even in the rendition of a single svara can reach up to 200 cents. To capture the intonation aspects in such scenarios it is essential to identify which svara is being rendered at a given point in time in a melody. For that, Koduri et al. (2014) also propose an alternate approach to compute a context-based svara distribution by categorizing pitch contours based on the melodic context. Subsequently, the parametrization to extract six features is performed on these context-based svara distribution. Koduri et al. (2014) report that features extracted using context-based svara distribution perform better in the rāga recognition task.

One of the shortcomings of the approaches discussed so far is that they do not consider the temporal aspects of melody, which are fundamental in characterization of rāgas. There exist a number of methods that statistically capture the temporal aspects of melody by modeling essentially the ārōhana-avrōhana progression of the rāgas (Pandey et al., 2003; Chordia & Rae, 2007; Shetty & Achary, 2009; Sridhar & Geetha, 2009; Dighe et al., 2013a; Kumar et al., 2014). A number of these methods compute svara sequence and employ techniques such as HMM and *n*-Gram to model the temporal aspects (Pandey et al., 2003; Dighe et al., 2013a; Kumar et al., 2014).

Some approaches compute a svara transition representation that captures the temporal aspects, such as the pitch-class dyad distribution (PCDD) in Chordia & Rae (2007), and the svara combination feature in Shetty & Achary (2009). These features are fed to a classifier to learn the discriminatory model for different rāgas. Few of these approaches also utilize characteristic melodic phrases of rāgas (Pandey et al., 2003; Sridhar & Geetha, 2009). They store a dictionary of pre-defined melodic patterns for each rāga, and subsequently detect their occurrences in the svara sequences obtained from the test recordings to recognize rāgas. The scalability of these approaches is however questionable, since they have been evaluated on a dataset comprising only two to three rāgas.

The approaches mentioned above invariably use a discrete representation of melody by either performing a simple svara-level quantization of the estimated pitch contours or by using a slightly more sophisticated melodic transcription technique (Pandey et al., 2003). Since automatic melodic transcription in IAM still remains a challenging and a rather ill-defined task (Widdess, 1994), this step may introduce errors that further propagate and influence the final accuracy of the methods. Although, a formal quantitative evaluation of the influence of the errors introduced in the melody transcription stage on the final accuracy is yet to be performed. Apart from the challenges involved in melody transcription, another limitation of the approaches mentioned above is that they fall short of capturing the characteristics of the continuous melodic transitions across the svaras, which is a relevant information for rāga recognition. Chalan of a rāga outlines the way melody progresses from one svara to another (the continuous melodic transitions), and therefore, is a characteristic feature (Section 2.3.2). Such fine grained temporal aspects of melody are considered in the approaches that use a continuous melodic representation and utilize melodic patterns for recognizing rāgas. However, due to the challenges involved in discovering melodic patterns in continuous melody representation, there are not many approaches that follow this methodology. Dutta et al. (2015) perform rāga verification using automatically discovered melodic patterns from specific sections (pallavi lines) of audio recordings in Carnatic music. A rāga verification system as opposed to a recognition system assumes that a specific rāga is claimed and the system checks whether the claimed rāga is correct or not. Thus, rāga verification can be regarded as a subset of rāga recognition with a reduced complexity of the task.

With the exception of the methods proposed by Dighe et al. (2013a,b) all other methods use pitch as the tonal feature for performing rāga recognition (Table 2.4). Existing methods for rāga recognition employ a variety of pitch estimation algorithms. While some of these algorithms are specifically designed to work with polyphonic audio music content (Salamon & Gómez, 2012), others are primarily suitable for monophonic speech signals (Boersma & Weenink, 2001). Thus, incorrect estimation of the pitch from the audio signals can be a source of error in the system. However, since some of these methods are evaluated using monophonic audio music content, and the others with polyphonic content, it is hard to make that conclusion from the results reported in

the studies. Verification of this hypothesis remains up to a comprehensive comparative evaluation on a common dataset. As noted above, the two exceptions to using the pitch feature for *rāga* recognition are the methods proposed by Dighe et al. (2013a,b), which use a 12 bin chroma feature to perform the task. Chroma feature has been widely used for key and mode identification task (Section 2.5.1). In the computation of the chroma features all the tonal components in the audio music signal are considered. For the case of IAM that would imply considering the *tablā* and the *tānpura* sound, which is often in the background and reinforce the base *svara* Sa. Thus, by considering the tonal components that are not at all related to the underlying *rāga* can degrade the performance of the method. However, since none of the studies (Dighe et al., 2013a,b) compare the performance with a pitch feature based *rāga* recognition system, no conclusion can be drawn without a comparative evaluation.

A crucial step in *rāga* recognition is making the method invariant to the tonic pitch of the lead artist used in an audio recording. As seen in Table 2.4, there are multiple ways in which this task is addressed by the existing approaches. A number of these studies either perform a tonic normalization by manually detecting its value for each recording, or they only consider performances in a fixed pre-defined tonic pitch (Pandey et al., 2003; Chordia & Rae, 2007; Belle et al., 2009; Shetty & Achary, 2009). In either of the cases, these methods are not scalable to real-world collections. This is because the tonic pitch varies across the artists and their recordings, and manually extracting this information is a cumbersome task. To alleviate this limitation, several methods either employ an external automatic tonic identification module (Koduri et al., 2012, 2014) or they explicitly identify tonic pitch prior to *rāga* recognition (Ranjani et al., 2011; Chakraborty & De, 2012). Another approach is to jointly estimate the tonic pitch and the *rāga* of a recording (Chordia & Şentürk, 2013; Koduri et al., 2011; Kumar et al., 2014). Joint recognition typically involves following a brute force methodology in which different feature candidates corresponding to all possible tonic values (usually quantized to the salient *svara* pitch values in the melody) are considered. The candidate that results in the best match is used to infer both the tonic and the *rāga* label. However, as shown in Chordia & Şentürk (2013) knowing a reliable tonic pitch in advance results in a significantly better performance compared to a brute force joint estimation. This indicates that an external module that can reliably identify tonic pitch can significantly boost the performance of *rāga* recognition. Using an external module for tonic identification might be advantageous because the relevant acoustic features for estimating tonic pitch and *rāga* might be different. For example, the background drone sound of the *tānpura* does not directly facilitate *rāga* recognition methods, however, that information can be exploited to reliably identify the tonic pitch in the recording (Gulati et al., 2014a).

An important component in any data-driven research is the presence of a diverse and a sizable music corpora, which is largely missing from the existing work on *rāga* recognition. There are different methods proposed for *rāga* recognition that use a variety of tonal features and learning methodologies. However, we know a little about their

comparative performances. This can be largely attributed to the lack of standard datasets (or corpus) for *rāga* recognition. As seen from Table 2.4, the datasets used for evaluation by the existing methods vary immensely in terms of the number of *rāgas*, the chosen set of *rāgas*, the duration and the number of the audio recordings per *rāga*, and the type of audio content (monophonic or polyphonic). With such diverse datasets, it is difficult to draw any concrete conclusions on the performance of the methods across different studies. Even the survey studies such as those in Koduri et al. (2011) have not performed any exhaustive comparative evaluations on the same dataset and under the same experimental setup. Thus, creating diverse, sizable and sharable datasets that are representative of the music tradition is a potential avenue for contribution in this task. In addition to the datasets, poor description of the implementation details is another common factor amongst several existing studies. This situation becomes even more difficult as none of the approaches make their code publicly available for ensuring reproducibility of the research results.

In addition to the issues mentioned above, we also identify other core avenues for improving the state-of-the-art in *rāga* recognition. From Table 2.3, it becomes evident that despite characteristic melodic phrases being the most salient cues for *rāga* recognition, there is a lack of approaches that effectively utilize them for this task. Thus, a system that can reliably extract melodic patterns in an audio music collection and can use those patterns for *rāga* recognition can be very valuable. From Table 2.3, we also notice that the set of approaches that capture the temporal aspects of melody use a discretized representation of melody. These approaches fail to capture the melodic transitions across *svaras*. Thus, there is also a need for approaches that can statistically capture both the tonal and the fine grained temporal aspects of melody by utilizing a continuous melody representation. Both these voids in the existing research on *rāga* recognition are addressed in this thesis in Chapter 6.

2.5 Related Work from Other Music Traditions

In the previous section, we reviewed the relevant work done in MIR for IAM. However, there are several tasks addressed for other music traditions of the world that are also pertinent to our work presented in this thesis. In this section, we review available literature on key and tonality modeling (Section 2.5.1), structural segmentation and pattern processing (Section 2.5.2), the topics in MIR that are most related to our work. Since in this thesis we work with recorded performances, during our review also we primarily focus on the approaches that operate on audio music collections. Note that the literature review presented in this section is not intended to comprehensively summarize and analyze the state-of-the-art, but aims at providing a broad-level perspective on the relevant research topics.

2.5.1 Key and Tonality Modeling

One of the research topics that is most related to the task of *rāga* recognition is that of modeling the tonality of an audio recording. In the last decade, tasks such as key estimation and chord recognition have been amongst the most popular topics in the MIR community. These research problems are studied from a computational, musicological and music cognition point of view. In this section, we review some of the prominent work done in key and tonality estimation.

In the literature there are numerous attempts to model and understand the human perception and cognition of tonality (Longuet-Higgins & Steedman, 1971; Krumhansl & Shepard, 1979; Chew, 2000; Krumhansl, 2000; Cohen, 1991). Different ideas have been proposed to explain the way human listeners perceive a key. Two very influential and complementary view points of key perception are distributional view and structural view (Brown, 1988; Temperley & Marvin, 2008). The distributional view to tonality considers that the perception of musical key depends on the the aggregate distribution of the pitch-classes in a music piece. Listeners possess some sort of cognitive template that relates with the distribution of the pitch-classes in all the major and the minor keys. However, there are other factors that affect the perceptual salience of a pitch-class such as its repetition, usage in melodic patterns and metrical position. By ignoring such important musical factors, distributional view receives several criticisms as well. An alternate explanation is the structural view, which emphasizes the importance of pitch ordering and the intervallic and scale-degree patterns that pitches in a music piece create.

A majority of the existing methods for key estimation tend to use the distributional viewpoint. One of the seminal works in automatic key estimation is the approach proposed by Krumhansl & Kessler (1982); Krumhansl (2001). The approach is based on key-profiles that represent the compatibility (or fittingness) of different pitch-classes for different keys. The key-profiles are experimentally derived using the probe-tone method described in Krumhansl & Shepard (1979). Krumhansl & Shepard (1979) showed that the judgments of fitness of a tone in the context of a short excerpt (comprising a cadence or a scale) that clearly defined a key is hierarchical in nature, with tones more commonly used in that key being found to be better fitting. Krumhansl and Kessler averaged these hierarchical responses across different contexts and keys to create a single major key-profile and a minor key-profile. The approach for key estimation works by correlating these reference key-profiles with the pitch-class profiles (PCPs) derived from the test musical material. Later, Temperley (1999) proposed modifications to this method. One of the major changes was the alteration in the key-profiles such as increasing the weight of the seventh scale degree in major and the raised seventh in minor mode to make them musically more relevant.

The methods mentioned above for key estimation work with a symbolic representation of music. However, a large number of the approaches for key estimation in audio recordings are essentially motivated by the same methodology (Gómez, 2006;

Pauws, 2004b; Peeters, 2006a). Although, the task of key estimation from audio recordings becomes much more challenging as the extraction of a reliable melody representation such as a music score from polyphonic music recordings is still a challenge (Gómez, 2006). A frequently used methodology for key estimation from audio signals comprise mainly three blocks: low-level feature extraction from audio signals, PCP computation, and finally, key estimation using the computed PCP features (Peeters, 2006a). Extracting individual pitches corresponding to different constituent instruments in a polyphonic audio mixture is a challenging task due to the presence of harmonics of these sounds sources. One of the ways to handle this issue is to suppress the harmonics of the individual sound sources (Cremer, 2004; Peeters, 2006a). The Second approach is based on considering the harmonics in the computation of the feature (Gómez, 2006; Izmirli, 2005). The chroma features proposed in Gómez (2006), harmonic pitch-class profiles (HPCP), have been proven to be very robust in working with polyphonic music recordings. These features are used in a variety of tasks in MIR such as cover-song identification (Serrà, 2011), audio thumbnailing (Bartsch & Wakefield, 2001) and structural analysis of music (Paulus & Klapuri, 2006). Computation of a PCP vector typically involves computing a global aggregate of the features computed for the individual audio frames (Izmirli, 2005; Pauws, 2004b).

Given a PCP (or a HPCP) summary vector, a typical approach to estimate the key is to use the maximum key-profile correlation method proposed in Krumhansl (2001). What this method essentially does is to perform a correlation of the estimated tonal profiles with all possible theoretical key-profiles derived in Krumhansl & Kessler (1982). The key-profile that results in the maximum correlation is marked as the key of the music piece. In addition to this cognition-inspired model of tonality, there have been some proposals for key estimation that use a machine learning approach to model tonality (Gómez & Herrera, 2004). The authors explore a number of classifiers such as support vector machines (SVM) and *k*-nearest neighbors (*k*-NN), and show that a machine learning-based approach performs better than the cognition-inspired model of tonality. The models discussed so far for key estimation do not consider the temporal aspect into account. HMMs can incorporate temporal dependencies, and have been used for key and chord estimation tasks (Noland & Sandler, 2006; Peeters, 2006b; Papadopoulos & Peeters, 2007). Although the performance of an HMM-based system for key estimation in certain music traditions is shown to be inferior to that of cognition-based models (Peeters, 2006b). For a comprehensive overview of the studies pertaining to tonality and key estimation in western music we refer to Gómez (2006).

The studies mentioned above mainly focus on the analysis of western popular and western classical music traditions. However, these methodologies have also influenced the approaches proposed for modeling tonality in other music traditions of the world, such as in Turkish makam music (TMM) (Gedik & Bozkurt, 2010, 2009; Bozkurt, 2008). One of the major differences between these approaches and the ones

mentioned above is the type of tonal feature used in the analysis. In the majority of the studies for TMM, the predominant pitch is used as the tonal feature as opposed to the chroma feature frequently used for western popular music. Another significant difference is in the pitch resolution of the PCP feature. Due to the presence of the micro-tonalities and the continuous melodic movements between the notes, a fine grained pitch distribution is considered as a feature for modeling tonality in TMM. Bozkurt (2008); Gedik & Bozkurt (2010) use a pitch distribution with bin-width as 1/3 Holdrian comma (Akkoç, 2002), which is approximately 7.5 cents. This results in a 159 dimensional PCD vector compared to a 12, 24 or 36 dimensional PCP vector often used for tonal analysis in western popular and western classical music traditions (Gómez, 2006). Such a fine grained representation of the tonal content of an audio recording has also been used for African music (Moelants et al., 2009) and Indian art music (IAM) (Chordia & Şentürk, 2013).

2.5.2 Pattern Processing in Music

Pattern detection and pattern discovery are active research topics in several domains dealing with multi-media content such as music (Klapuri, 2010), audio (Herley, 2006) and speech (Park & Glass, 2008). We refer to both these tasks jointly as pattern processing. Before delving in to the literature, it is helpful to distinguish the two terms, pattern detection (or pattern matching) and pattern discovery (or pattern extraction). In pattern detection, given a query, the objective is to retrieve all its occurrences (exact or inexact) from a target set. In the context of music, a query typically comprise a short music excerpt and the target set is either a long music piece or their collection (Ghias et al., 1995). In pattern discovery on the other hand, there is no query excerpt, the objective is to discover repeating segments (exact or inexact) from the data itself following an unsupervised methodology (Dannenberg & Hu, 2003). There are two types of pattern discovery tasks in MIR, intra-opus discovery and inter-opus discovery (Conklin & Anagnostopoulou, 2001). In the former, the unit of the data from where the patterns are discovered is a single music piece, and in the latter, the discovery is performed at the level of a music corpus (containing multiple music pieces).

Music pattern discovery is an important task in several areas within music research (Collins et al., 2011; Conklin & Anagnostopoulou, 2011; Serrà et al., 2014; Nieto et al., 2012). Different types of musical patterns are discovered at different time-scales: long-duration repetitions such as different sections of a music piece (Serrà et al., 2012; Goto, 2006), relatively small-duration repetitions being themes, riffs (Hsu et al., 2001), and melodic motifs (Collins, 2011). Our review of the pattern processing approaches is divided into three parts: motif discovery in symbolic music domain (Section 2.5.2.1), structural segmentation of audio recordings (Section 2.5.2.2), and query-by-humming (QBH) (Section 2.5.2.3). We reiterate that the literature review presented in this section mainly aims to provide a broad perspective on the research topics, and is not comprehensive.

2.5.2.1 Motif Discovery in Symbolic Music

Literature is replete with approaches for melodic pattern discovery in symbolic music representations of western classical music (Cambouropoulos, 1997; Meredith, 2006; Conklin & Anagnostopoulou, 2001; Lartillot, 2005a). Even before the advent of computational machinery melodic pattern analysis has been widely used in musicological studies. Computational approaches for motif discovery typically involve processing blocks such as melody representation (Meredith, 2006), melody segmentation (Cambouropoulos, 2006), melodic similarity computation (Cambouropoulos, 2001b; Marsden, 2012b), pattern discovery methodology (Collins et al., 2013; Meredith et al., 2002) and redundancy filtering or pattern ranking (essentially identifying musically relevant patterns) (Lartillot, 2005b; Conklin, 2010).

For symbolic music data there are two main types of melody representations often used in the literature, viewpoint representation and geometric representation. Viewpoint representation encodes multiple aspects of melody (such as the pitch, pitch interval and duration between notes) as strings of symbols (Conklin & Anagnostopoulou, 2001; Conklin & Witten, 1995). These symbols can be further grouped into two categories (Meredith et al., 2002): event string, where the symbols represent a musical event, and interval string, where the symbols represent the transformation of the current event relative to the previous one (pitch interval for example). Geometric representation on the other hand converts each note in to a pitch-time space (or even higher-dimensions) (Meredith et al., 2002). In this representation a melody is essentially considered as a shape in an n-dimensional space, wherein the patterns are identified as approximately identical shapes (Meredith et al., 2002). Strengths and weaknesses of both these representations are thoroughly discussed in the literature (Cambouropoulos, 2009; Meredith et al., 2002). In general, viewpoint representations are argued to be more appropriate for monophonic music, and geometric representations is more suited to handle polyphonic music.

An essential step in analyzing and describing melodies is to segment them into meaningful temporal units, which typically correspond to melodic phrases. Music parallelism is an important factor in music segmentation, and therefore, has also been utilized for segmenting melodic sequences (Cambouropoulos, 2006). The task of melody segmentation and pattern discovery are interdependent. Pattern discovery can facilitate melody segmentation (Cambouropoulos, 2006), and a meaningful pre-segmentation of melodies can significantly improve the performance of the pattern induction techniques (Hiraga, 1997). Cambouropoulos et al. (2001) summarize different melodic pattern processing approaches and highlight the ones that require pre-segmented melodic sequences. There are several other models for melody segmentation apart from using music parallelism. One of the ways to group musical events, which is another way to look at segmentation process, is through Gestalt principles (Lerdahl & Jackendoff, 1983; Tenney & Polansky, 1980). Grouping in melodic sequences is often done through the identification of local discontinuities between melodic events

in terms of their temporal proximity, pitch and duration. Events that are contiguous in time or are similar in terms of other parameter form groups. In Cambouropoulos (1996, 2001a), the authors outline several problems in the application of the low-level Gestalt rules in existing theories. They propose a *Local Boundary Detection Model* that constructs a boundary salience based on the strength of the local discontinuities in melodic sequences, wherein the peaks of the salience function are considered as potential candidates for local boundaries. The authors describe change-rule and proximity-rule, which are shown to be more effective approaches for low-level segmentation compared to other models. For a detailed review of the existing approaches for melody segmentation we refer to Cambouropoulos et al. (2001).

One of the most crucial processing blocks in pattern discovery is the computation of melodic similarity. There are different ways in which this task is approached. Broadly, the existing approaches can be divided into two categories, the ones performing exact matching, and the others, that use an approximate matching technique (Cambouropoulos et al., 2001). Typically, the choice of the method to compute melodic similarity depends on the melody representation. If the melodic variations are abstracted in the representation, an exact matching technique might be used to detect patterns. Otherwise, an approximate pattern matching technique, which often involves usage of dynamic programming based algorithms is applied (Rolland, 1999). A number of approximate pattern matching algorithms are discussed in Section 2.5.2.3.

Pattern discovery methodologies that are based on string matching algorithms can be broadly grouped into two categories depending on whether or not they employ indexing structures (Janssen et al., 2013). An example of an approach that does not employ any indexing structure is Nieto & Farbood (2012), which uses a brute-force strategy to match all possible pattern candidates with different lengths. Such approaches tend to be computationally complex and often intractable for large volumes of data. Approaches which do employ indexing structures typically convert melodic sequences to a tree-like structure for efficient retrieval of patterns (Knopke & Jürgensen, 2009; Conklin & Anagnostopoulou, 2011). For a comparative description of these different approaches we refer to Janssen et al. (2013); Meredith et al. (2002).

Computational approaches for pattern discovery typically result in large volumes of patterns, many of which are musically uninteresting (Marsden, 2012a; Conklin, 2010). As a result, these approaches apply a filtering step to select only the musically significant patterns. There are different strategies to perform this selection. A frequently used approach is to prefer long duration patterns (Cambouropoulos, 2006; Karydis et al., 2006). Some approaches prefer patterns that occur more frequently than others (Cambouropoulos, 2006; Meredith et al., 2002). The methods proposed by Conklin (2010); Conklin & Anagnostopoulou (2011) are also based on patterns' frequency of occurrence, but inversely weighted by its frequency of occurrence in an anti-corpus. Collins et al. (2011) evaluate several strategies for assigning importance to discovered melodic patterns.

Overall, we see that a considerable amount of work is done in melodic motif discovery using symbolic music representations. However, these approaches cover only a particular view of music, and are not directly applicable to analysis of melodic aspects of recorded performances. This is largely due to the difficulties in obtaining a reliable symbolic representation of melody from an audio recording. As suggested by Collins et al. (2014), the audio-symbolic gap can be bridged by a reliable melody transcription system. However, for certain music traditions such as IAM, melody transcription is a challenging task (Widdess, 1994). Due to these factors it is desirable to develop computational approaches for melodic pattern discovery that can directly work on continuous melody representations extracted from audio recordings.

2.5.2.2 Structural Segmentation

A considerable amount of research work on pattern discovery in audio recordings is focused on structural segmentation of music, wherein different sections of a song can be regarded as long-duration music patterns (Paulus et al., 2010). While some of the existing approaches focus on merely segmenting the audio recordings into meaningful sections, others also cluster and label these sections to produce a better description of the audio recordings.

Approaches for structural analysis in audio typically start with extracting low-level audio features such as Mel-frequency cepstral coefficients (MFCCs) and chroma features (for example, HPCP (Gómez, 2006)) that represent the timbral and the tonal (encompassing harmonic) characteristics of a music recording. The frame-based features are often summarized or averaged within the beat durations since the relevant chroma information often remains the same in that time-scale. This step also reduces the dimensionality of the audio representations. Using the reduced audio representation, a typical approach for structural segmentation involves computing a self-similarity matrix (SSM) (Foote, 2000). SSM captures similarities between all beat duration segments with the rest of such segments. There are other ways of representing self-similarity of a recording such as through recurrence plots, which are based on the concept of delay-coordinates (Serrà et al., 2014). Several approaches use time-lag matrices, which are typically derived from SSM or recurrence plots (Goto, 2006).

Paulus et al. (2010) describe three basic segmentation principles for inferring music structure: novelty, homogeneity, and repetition. Novelty-based approaches work by detecting transitions between acoustically contrasting segments of the song. Foote (2000) uses a novelty-based method to detect the point of transition in a recording. The method involves correlation of a short-time checkerboard kernel along with diagonals of the SSM. The one dimensional novelty curve that results from this correlation is then used to detect the point of change. Homogeneity-based approaches aim to detect segments in time that are consistent in terms of musical property such as timbre and harmony. Levy & Sandler (2008) exploit homogeneity principle and proposes an approach for structural segmentation that uses the hidden markov model (HMM).

Repetition-based approaches aim to identify repeating long-time patterns that relate to the structure of the piece (Goto, 2006; Dannenberg & Hu, 2003; Müller & Kurth, 2006). Serrà et al. (2014) propose an approach based on the combination of structure features and segment similarity that combines all the segmentation principles and show improved results. Repetition-based structural segmentation methodologies are also utilized for a number of other pattern discovery tasks in MIR, such as audio thumbnailing or summary generation (Chai & Vercoe, 2003; Aucouturier & Sandler, 2002; Muller et al., 2013; Nieto et al., 2012).

2.5.2.3 Query-by-humming

The approaches mentioned above typically focus on long-duration music repetitions and use music representations that consider multiple aspects of music such as timbre, rhythm and harmony. Since in this thesis we focus on analysis of short-duration patterns in melody representations, the existing literature on the task of QBH is relatively more relevant. This task has been studied in MIR for more than two decades (Ghias et al., 1995; McNab et al., 1996). However, as opposed to pattern discovery, QBH is a pattern matching task. In QBH, given a sung or hummed melodic query from a user, the objective is to retrieve music from a database based on matching the query fragment.

Initial work on QBH was focused on the retrieval from MIDI or score databases. One of the earlier approaches was proposed by Ghias et al. (1995). The authors extract the pitch from the incoming audio query and convert it to a string sequence comprising three different symbols (U, D, S). These symbols represent the relative pitch changes across successive notes as higher or upper (U), lower or down (D), and same (S). A similar representation is also extracted from a database comprising MIDI files. Usage of such a melody representation makes the system robust to absolute pitch, which is one of the desired properties of a QBH system. Another desired feature is to handle insertion and deletion type of errors in the query string representation. These errors essentially arise due to the timing variations in the query phrase compared to the original music piece. Taking these objectives into account, the database search is performed using an approximate string matching algorithm (Baeza-Yates & Perleberg, 1992).

There are different kinds of melody representations used in the task of QBH. A set of methods represent melody by using the absolute pitch values (McNab et al., 1996; Uitdenbogerd & Zobel, 1998). Some approaches consider only the pitch intervals between the adjacent notes (Pauws, 2004a), and the others use only a relative pitch change across notes (up or down) (Ghias et al., 1995). There are very few approaches that consider a continuous pitch representation extracted from the query excerpt in the matching (Mazzoni & Dannenberg, 2001). Since melodies inherently embed rhythm information, the duration of the notes becomes an important factor in recognizing melodies. It has been shown that embedding the duration information of the notes in

a melody representation improves the retrieval performance in QBH (Pardo & Birmingham, 2002). Again, the duration information can be included in a number of ways. Pardo & Birmingham (2002) propose to use quantized versions of the **inter onset interval** (IOI) and **inter onset interval ratio** (IOIr) computed in both linear and log domain. Such a representation also makes the system more invariant to linear tempo scalings. Dannenberg et al. (2007) perform a comparative evaluation of different representations of melody and show that best results are obtained with the representation that uses relative pitch intervals and log IOI ratios.

There are different approaches to match the melody representation obtained from the query with the target database. One of the most commonly used approaches is to use string matching based algorithms that are often based on dynamic programming. Mazzoni & Dannenberg (2001) follow a brute-force approach and match the continuous pitch representation of the query with every possible segment in the target song for every possible pitch transformation. The authors use a DTW-based distance measure to compute the melodic similarity. Such an approach was found to be better than the approaches that perform pitch quantization on query pitch representation. However, a brute-force segmentation and the usage of a DTW-based distance measure makes it practically intractable for large datasets. One of the ways to avoid brute-force segmentation is to use a subsequence variant of the DTW distance. Such a DTW variant enables the query string to match the reference song from any position in the song. Jang & Gao (2000) use a subsequence variant of DTW with local constraints to avoid pathological warping. The authors embed transposition invariance in the cost function computation within the DTW-distance by using a heuristic-based recursive approach. Another variant of DTW is constrained dynamic time warping (cDTW), which is suitable in both whole and subsequence matching scenarios and is known to be computationally more efficient than DTW (Lijffijt et al., 2010). Zhu & Shasha (2003a) utilize some well developed indexing techniques from the time-series analysis domain to make the DTW-based QBH systems scalable to sizable datasets. The authors combine the idea of uniform time warping (UTW) with DTW and propose a variant referred to as local dynamic time warping (LDTW). The method is essentially a two-step process which starts by first globally time-stretching the two subsequences (the query and target) to the same length. Subsequently, in the second step a band-constrained DTW is used for computing the melodic similarity. LDTW variant of DTW is reported to result in tighter lower-bounds, and is therefore computationally more efficient. In addition, the authors use the GEMINI framework to reduce the dimensionality of the time-series before indexing (Keogh et al., 2001). The pitch transposition invariance in this study is incorporated by mean normalization of the query and the target sequences. Through an extensive set of experiments the authors have shown that such a system performs better both in terms of the precision and the computational cost as compared to the previous approaches (Zhu & Shasha, 2003a).

In addition to the DTW-based sequence matching approaches, different variants of edit distance are also used in QBH. Uitdenbogerd & Zobel (1999) use longest com-

mon subsequence (LCS) to match query sequence to a MIDI database. LCS measures the largest common number of elements between two sequences. Since this method allows for gaps during the alignment it is more robust to addition noise in the query subsequence. User queries often contain such additions (for example, elongated notes), which makes this method suitable for the task of QBH. However, if the allowed gaps are not controlled, this method can lead to a large number of false positives. Algorithm proposed by Iliopoulos & Kurokawa (2002) alleviate some of these issues by taking into account a bounded number of gaps in the target sequence. Lin et al. (2011) propose RLCS, an improved variant of LCS that avoids common problems that occur in the global alignment matching. A rough equality for two notes is defined for constructing the RLCS. This method also considers both the width-across-query (WAQ) and the width-across-reference (WAR) and combines them with the weighted length of the corresponding RLCS to obtain a measurement score for the RLCS. Kotsifakos et al. (2011) also improve LCS by introducing the gaps-range-tolerances framework. The authors refer to their method as subsequence matching with bounded gaps and tolerances (SMBGT). SMBGT allows for controlled amount of gaps in both the query and target sequences, variable levels of matching tolerance, and also constrains the maximum match length. Apart from these methods, dynamic programming-based local alignment methods such as the one proposed in Smith & Waterman (1981) are also explored in the context of QBH (Uitdenbogerd & Zobel, 1999). Such methods can compare variable lengths of two sequences and can optimize the similarity measure by a local alignment of the sequences. Apart from the whole and subsequence string matching methods mentioned above there are also model-based approaches (HMM and *n*-Gram) used for QBH (Durey & Clements, 2001; Jang et al., 2005; Uitdenbogerd & Zobel, 1999; Dannenberg et al., 2007).

Until recently, a significant number of approaches discussed above could only match queries against symbolic databases (mainly MIDI files) (Kotsifakos et al., 2012). This is primarily due to the challenges involved in obtaining a reliable melody representation from polyphonic audio recordings (Salamon et al., 2013). One of the solutions is to match queries against other queries recorded by the users (for example, methodology used in SoundHound¹⁵), which are eventually mapped to audio recordings. However, such an approach suffers from the “cold start” problem. Given this situation, an automated method to generate melody database is very valuable (Salamon et al., 2013).

The methodologies used in the QBH task are also extended to detect short-duration melodic patterns in several music traditions (Pikrakis et al., 2003, 2012, 2016; Ross et al., 2012; Dutta & Murthy, 2014b). In these studies, instead of a user humming a melody, the query fragment is obtained from the melodic phrase annotations done by domain experts. Given some instances of melodic phrases in audio recordings, the objective in these studies is to automatically annotate their other repeated occurrences.

¹⁵<http://www.soundhound.com/>

This task is also sometimes referred to as **query-by-example** (QBE).

Pikrakis et al. (2003) illustrate some of the challenges involved in the detection of melodic patterns in audio recordings of Greek tradition music. A number of these challenges arise due to the errors introduced by an inaccurate pitch estimation step. In order to make the system invariant to these errors, the authors propose a variant of DTW, referred to as **context dependent dynamic time warping** (CDDTW). The method operates on a melody representation comprising frequency jumps across successive notes derived from a simple pitch quantization of continuous pitch contours. Closely related variants of this method are also used for detection melodic patterns in audio recordings of two fandango styles in Flamenco music (Pikrakis et al., 2012; Gómez et al., 2012). In a recent study, Pikrakis et al. (2016) perform a melodic pattern detection task, but on a phrase labeled MIDI corpus of Flamenco music. The authors propose an extension to the algorithm proposed by Needleman & Wunsch (1970). The extension proposed by the authors enables this algorithm to be used as a subsequence matching algorithm (the original algorithm is for global alignment). Transposition invariance is ensured by using a melody representation that considers relative pitch interval across successive MIDI symbols (notes). Note that in this method the ground truth annotations are also done on MIDI sequences, thus further reducing the amount of errors that might arise in the melody extraction stage from the reference annotations. A number of approaches for pattern detection following similar methodologies are also proposed for IAM (Ross et al., 2012; Dutta & Murthy, 2014b; Ganguli et al., 2015), which we have reviewed in Section 2.4.2.

As mentioned earlier, one of the main tasks addressed in this thesis is that of pattern discovery in audio collections. There are very few studies that address this task. Kroher et al. (2015) aim to discover melodic patterns in audio recordings of Flamenco music. The authors start by segmenting the audio recordings based on the singing voice activity. An audio fragment between any two unvoiced regions is considered as a valid segment to be used for discovery. The method operates on a chroma representation extracted from the audio recordings (Bartsch & Wakefield, 2005). The authors use a 24 bin resolution chroma feature due to the micro-tonality present in the music style. Subsequently, a pairwise similarity is computed for all the extracted segments using a local sequence alignment method. The top 15 patterns are then selected and clustered together using a simple heuristic-based approach in order to identify all the repetitions of a unique melodic phrase. This study reports promising results in evaluation done with 11 audio recordings of Flamenco music containing three to seven phrases per recording. However, this approach is computationally expensive due to the usage of a dynamic programming-based distance measure. The scalability of such an approach to hundreds of hours of audio music collection is questionable, which is one of the main desired features of the pattern discovery approach that we aim for in this thesis.

2.6 Mathematical Background

In this section, we present a brief description of the main mathematical concepts that are used in the subsequent chapters. This description primarily intends to serve as a quick reference, and by no means provides a comprehensive information on the concept. For a more detailed description of these concepts external references are provided. Note that the scope of the mathematical symbols used in this section is only within the description of the individual concepts.

2.6.1 Distance Measures

In this section we provide formulations for the distance measures used in this thesis.

2.6.1.1 Euclidean Distance

The Euclidean distance between two n -dimensional points X and Y is the length of the line segment connecting them. If $X = x_1, x_2 \dots x_n$, and $Y = y_1, y_2 \dots y_n$ are two points in Euclidean n -space, then the Euclidean distance (\mathcal{D}_E) between them is given by:

$$\mathcal{D}_E(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

2.6.1.2 Bhattacharyya Distance

The Bhattacharyya distance (\mathcal{D}_B) for two probability distributions X and Y over the domain R , is defined as (Bhattacharyya, 1946):

$$\mathcal{D}_B(X, Y) = -\ln(BC(X, Y)) \quad (2.2)$$

where:

$$BC(X, Y) = \sum_{r \in R} \sqrt{X(r)Y(r)} \quad (2.3)$$

2.6.1.3 Kullback–Leibler divergence

The Kullback–Leibler divergence (\mathcal{D}_{KL}) from Y to X , where X to Y are two discrete probability distributions, is defined as (Kullback & Leibler, 1951):

$$\mathcal{D}_{KL}(X||Y) = \sum_i X(i) \log \frac{X(i)}{Y(i)} \quad (2.4)$$

Symmetric Kullback–Leibler divergence is given by $\mathcal{D}_{KL}(X||Y) + \mathcal{D}_{KL}(Y||X)$.

2.6.2 Dynamic Time Warping

Dynamic time warping (DTW) is a well-known dynamic programming-based time series alignment algorithm. This algorithm was originally proposed for speech recognition (Sakoe & Chiba, 1978). DTW is widely used for measuring similarity (or dissimilarity) between two temporal sequences that often require an optimal non-linear time alignment. A comprehensive information about DTW in the context of music sequences can be obtained from Müller (2007).

We now briefly describe the computation of the DTW distance (taken from Keogh & Ratanamahatana (2004); Müller (2007)). Given two time series, a sequence X of length N , and a sequence Y of length M , where:

$$X = x_1, x_2 \cdots x_i \cdots x_N \quad (2.5)$$

$$Y = y_1, y_2 \cdots y_j \cdots y_M \quad (2.6)$$

A local cost matrix $C \in \mathbb{R}^{N \times M}$ of the two real valued sequences X and Y can be defined by:

$$C(i, j) = d(x_i, y_j) \quad (2.7)$$

where, $d(x_i, y_j)$ is the local distance measure between x_i and y_j .

A warping path is a sequence $p = (p_1, p_2 \cdots p_i \cdots p_L)$ that defines a mapping between X and Y with $p_l = (i_l, j_l) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$ satisfying three conditions:

- **Boundary conditions:** $p_1 = (1, 1)$ and $p_L = (N, M)$. The warping path is required to start and finish in the diagonally opposite corners of the matrix C .
- **Step size:** $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1 : L - 1]$. This restricts warping path to adjacent cells in the matrix.
- **Monotonicity:** Given $p_l = (i_l, j_l)$ and $p_{l-1} = (i_{l-1}, j_{l-1})$, then $i_l - i_{l-1} \geq 0$ and $j_l - j_{l-1} \geq 0$. This requires p to be monotonically increasing.

In the computation of the DTW we are interested in the warping path that minimizes the warping cost:

$$\text{DTW}(X, Y) = \min \left\{ \sum_{l=1}^{l=L} C(p_l) \right\} \quad (2.8)$$

This optimal alignment path can be computed in $\mathcal{O}(NM)$ by using dynamic programming. A accumulated $N \times M$ cost matrix D can be defined such that $D(N, M) = \text{DTW}(X, Y)$. The accumulated cost matrix D can be computed as:

$$D(i, j) = \min\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\} + d(x_i, y_j) \quad (2.9)$$

The optimal path can be computed by tracking back the accumulated cost matrix D . Starting with $p_L = (N, M)$ and supposing $p_l = (i, j)$, p_{l-1} can be computed as:

$$p_{l-1} : \begin{cases} (1, j - 1), & \text{if } i = 1 \\ (i - 1, 1), & \text{if } j = 1 \\ \operatorname{argmin}\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\}, & \text{otherwise,} \end{cases} \quad (2.10)$$

There are several variants of the classical DTW described above. One of the variant that we will use changes the step condition in the DTW calculation. In several situations classical DTW suffers from pathological warpings, wherein a single element from one sequence is mapped to a large number of elements in the other sequence. In order to avoid such situations the step size condition can be modified to constrain the slope of the warping paths. The step size condition described above can be modified to $p_{l+1} - p_l \in \{(2, 1), (1, 2), (1, 1)\}$ for $l \in [1 : L - 1]$. As a result of this modification the warping paths have a local slope bound of $\frac{1}{2}$ and 2. With this step condition the accumulated cost matrix D can be recursively computed as:

$$D(i, j) = \min\{D(i - 2, j - 1), D(i - 1, j - 2), D(i - 1, j - 1)\} + d(x_i, y_j) \quad (2.11)$$

Note that we have not described the initialization procedure of the cost matrix D to start recursion, for which we refer the readers to Müller (2007).

In addition to the step size condition, another variants of the DTW impose different kinds of constraints on the warping path. Such constraints are not only advantageous in terms of the computational complexity, but they also prevent pathological alignment problems. Two very well known global path constraints are the Sakoe-Chiba band (Sakoe & Chiba, 1978) and the Itakura parallelogram (Itakura, 1975). In the subsequent chapters we use Sakoe-Chiba band constraint. More information about different variants of DTW can be obtained from (Rabiner & Juang, 1993; Müller, 2007).

2.6.3 Lower-bounds for DTW distance

Time and space complexity of the classical DTW variant is $\mathcal{O}(NM)$ for the two sequences X and Y considered above. Thus, a similarity search under DTW is very demanding in terms of the CPU time. As described in Keogh & Ratanamahatana (2004) one of the ways to address this problem is to use efficient lower-bounds on the DTW distance. Lower bounding functions facilitate in pruning the sequences candidates that could not possibly be the best matches. To understand it further we present

a pseudo-code of a sequential scan search for query Q in Algorithm 1. Different possible subsequence candidates that can be possible matches are represented by C_i .

Algorithm 1 Sequential scan with lower bounding technique

```

bsf = ∞                                ▷ best distance so far
for all subsequence candidates in database do
    dist_LB = lower_bound( $C_i, Q$ )
    if dist_LB < bsf then
        dist_True = DTW( $C_i, Q$ )
        if dist_True < bsf then
            bsf = dist_True
            index = i                         ▷ index of the best match

```

The desired properties of a good lower-bounding function are: significantly low computational complexity compared to DTW, and the lower bound must be tight (Keogh & Ratanamahatana, 2004). There are a small number of lower-bounding techniques applicable to DTW as compared to other string edit or tree edit distance measures. In this thesis, we use the lower bounding technique proposed by Keogh & Ratanamahatana (2004) (hereafter, referred to as LB_Keogh).

A global or a local path constraint on DTW can be used to define a constraint on the indices of the warping path $p_l = (i_l, j_l)$ such that $j - r \leq i \leq j + r$, where r is the allowed range of warping. For the case of (Sakoe & Chiba, 1978), a global band r is independent of i . Keogh & Ratanamahatana (2004) defines two new sequences (U and L) using r :

$$U_i = \max(x_{i-r} : x_{i+r}) \quad (2.12)$$

$$L_i = \min(x_{i-r} : x_{i+r}) \quad (2.13)$$

With U, L and $d(x_i, y_j)$ (as Euclidean distance), LB_Keogh lower bound can be defined as:

$$\text{LB}_\text{Keogh}(X, Y) = \sqrt{\sum_{i=1}^N \begin{cases} (y_i - U_i)^2, & \text{ify}_i > U_i \\ (y_i - L_i)^2, & \text{ify}_i < L_i \\ 0, & \text{otherwise} \end{cases}} \quad (2.14)$$

The proof that LB_Keogh produces a lower-bound on DTW, i.e. $\text{LB}_\text{Keogh}(X, Y) \leq \text{DTW}(X, Y)$ can be obtained from Keogh & Ratanamahatana (2004).

2.7 Summary

This chapter provided the scientific and music background relevant to the work presented in this thesis. We briefly introduced a number musical concepts and terminologies

related with the melodic facets in **IAM**. Within the scientific background, we focused on the existing approaches for relevant computational tasks in **IAM**. We presented our review of the current methods for tonic identification, melodic pattern processing and *rāga* recognition. We critically analyzed and compared these methods in terms of algorithmic design, evaluation methodology and their scalability on sizable datasets. In general, we found that a majority of the current methods are not evaluated on representative datasets of **IAM**, and these studies rarely compare their approach with other existing approaches. A number of avenues for scientific contributions were identified in this context. To provide a broader perspective, we also presented an overview of the relevant literature in **MIR** for music traditions other than **IAM**. We highlighted the prominent work done in tonality modeling, motivic analysis using symbolic music representations, structural segmentation and **QBH**. At the end of the chapter, we provided a brief description of different distance measures used in our work. We focused mainly on the **DTW** distance and its variants, and the lower-bounding techniques used in this thesis.

Chapter 3

Music Corpora and Datasets

3.1 Introduction

A research corpus is a collection of data compiled to study a research problem. A well designed research corpus is representative of the domain under study. It is practically infeasible to work with the entire universe of data. Therefore, to ensure scalability of information retrieval technologies to real-world scenarios, it is important to develop and test computational approaches using a representative data corpus. Moreover, an easily accessible data corpus provides a common ground for researchers to evaluate their methods, and thus, accelerates knowledge sharing and advancement.

Not every computational task requires the entire research corpus for development and evaluation of approaches. Typically a subset of the corpus is used in a specific research task. We call this subset a test corpus or test dataset. The models built over a test dataset can later be extended to the entire research corpus. Test dataset is a static collection of data specific to an experiment, as opposed to a research corpus, which can evolve over time. Therefore, different versions of the test dataset used in a specific experiment should be retained for ensuring reproducibility of the research results. Note that a test dataset should not be confused with the training and testing split of a dataset, which are the terms used in the context of a cross validation experimental setup.

In MIR, a considerable number of the computational approaches follow a data-driven methodology, and hence, a well curated research corpus becomes a key factor in determining the success of these approaches. Due to the importance of a good data corpus in research, building a corpus in itself is a fundamental research task (MacMullen, 2003). MIR can be regarded as a relatively new research area within information retrieval, which has primarily gained popularity in last two decades. Even today, a significant number of the studies in MIR use ad-hoc procedures to build a collection of data to be used in the experiments. Quite often the audio recordings used in the experiments are taken from a researcher's personal music collection. Availability of a good representative research corpus has been a challenge in MIR (Serra, 2014). This

can be attributed to an extent to the large variety of research problems studied within **MIR**, lack of standardized methodologies for data collection and annotation, and most of all, due to the constraints posed by the copyrighted content.

In recent years, there have been various efforts to compile large collections of music related data to build a research corpus that can be used to study a number of computational tasks in **MIR**. One such example is the **million song dataset (MSD)** (Bertin-Mahieux et al., 2011), which is already used in several studies for a variety of computational tasks (Serrà et al., 2012; Sturm, 2012). However, owing to the copyright issues, the audio recordings in **MSD** are not available. Building a good representative research corpus, which in **MIR** would typically mean compiling a large collection of music recordings and their related metadata is a substantial effort. A successful and sustainable strategy could be to make it a community effort. An endeavor in this direction is AcousticBrainz (Porter et al., 2015)¹⁶, which aims to crowd source acoustic information for music recordings and make them available under public domain. Data in AcousticBrainz is indexed by unique **MusicBrainz identifier (MBID)**. MusicBrainz¹⁷ is an open music encyclopedia that collects music metadata and makes it publicly available. Such open repositories can also serve as corpus for a variety of research problems in **MIR**. In addition to these, there have also been efforts to compile music corpus such as COFLA corpus¹⁸ in order to perform computational studies of specific music traditions (Kroher et al., 2016).

While there is an increasing effort towards building and using a representative research corpus in **MIR**, there are not many studies that address formally the task of building a good research corpus. There is a lack of studies that discuss the criteria for determining the goodness of a corpus for a particular task and systematic ways to compile and curate research corpus. Some recent efforts towards this direction include the work by Peeters & Fort (2012) in which the authors present a unified way to describe annotated **MIR** datasets. Humphrey et al. (2014) define a specification to store annotations in a more unified way to promote reproducibility and easy access of the corpora. As a part of the CompMusic project, Serra (2014) presents a set of design criteria for building research corpora that is representative of a given domain of study. These criteria are based around considerations such as purpose, coverage, completeness, quality and reusability.

In this chapter, we describe the CompMusic research corpora built for studying a number of computational tasks in **MIR** of **IAM**. Before describing the corpora, we briefly discuss the methodology and the design criteria used to compile and curate the corpora. Note that the sources used for compiling the corpora are not comprehensive. Our primary aim is to present the approach that we used to build the corpora rather than to justify a specific data source. In addition to the corpora, we also describe the test datasets that we built for studying different melodic aspects in audio collections

¹⁶<https://acousticbrainz.org/>

¹⁷<https://musicbrainz.org/>

¹⁸http://www.cofla-project.com/?page_id=170

of IAM (Section 3.3). These test datasets are used for the development and evaluation of a number of approaches described in the subsequent chapters.

The compilation and curation of this research corpora has been a collective effort by the CompMusic team. This task mainly involved defining the criteria for selecting music material, procuring audio recordings, adding the associated editorial metadata to the MusicBrainz, organizing and maintaining music collections on the servers, and correcting erroneous metadata with the help of the domain experts. Contributions by the author have been made in all these steps with more focus on the Hindustani music corpus. The description of the corpora and the design criteria provided in the subsequent sections is primarily taken from the work presented by Srinivasamurthy et al. (2014); Serra (2014).

3.2 CompMusic Research Corpora

As mentioned in Section 1.2, the CompMusic project focuses on data-driven computational approaches to describe music recordings and emphasize the use of domain knowledge of a particular music tradition. The project considers five different music traditions: Arab-Andalusian (Maghreb), Beijing Opera (China), Turkish makam music (Turkey), Hindustani (North-India) and Carnatic (South-India) music. One of the key ideas in the project is that there are some universal music concepts such as melody and rhythm, which are common across different music traditions. But, many important aspects of a music piece can be better understood and appreciated by focusing on the specificities of the music tradition. Therefore, a significant effort in the CompMusic project has been to compile a representative research corpora that captures the specificities of different music traditions considered in the project. Furthermore, an effort is made to define the criteria that can be used to build and assess a good research corpora. In the subsequent section, we briefly describe these design principles.

3.2.1 Criteria for Building CompMusic Corpora

Serra (2014) enumerates a set of design criteria for building a representative research corpus. These are the principles used in the CompMusic project to build music corpora with which to computationally analyze different music traditions. For this thesis to be self-contained, we here provide a brief description of these criteria based on the explanation given by Serra (2014).

Purpose: The purpose for building a data corpus should be clearly specified at the onset. This includes defining the research problems that need to be addressed and the research methodologies that will be used. In the CompMusic project, one of the main objectives is to develop methodologies to extract musically meaningful melody and rhythm related features from audio music recordings. The approaches

are mostly based on signal processing and machine learning techniques. A research corpus should take these factors into account.

Coverage: As mentioned, a good research corpus should be representative of the domain under study. Coverage is a measure of representativeness of a corpus with respect to the concepts to be studied. Given the focus on the quantitative approaches in the CompMusic project, we need sufficient instances of each concept for the data to be representative. For melodic analysis, we need audio recordings and accompanying metadata that represent the diversities present in the melodic aspects of Hindustani and Carnatic music such as different forms, sections, variety of *rāgas*, artists from different schools of music, and recordings in different śrutis.

Completeness: To successfully use data in meaningful analyses it should be complemented by appropriate metadata. Completeness indicates the completeness of the associated information or metadata for each audio recording. For the CompMusic corpus it mainly refers to the completeness of the editorial metadata and of the descriptive information that accompany each audio recording.

Quality: The quality of the data in a research corpus should be good. In our case it means that the audio should be well recorded, and the accompanying metadata should be accurate. In the CompMusic project, we use good quality commercially produced audio recordings, and the accompanying information is obtained from reliable sources. However, at times, there are errors in this information even after using reliable sources such as editorial metadata on CD covers. In such cases, the metadata is validated and corrected with the help of domain experts.

Reusability: Reusability of a corpus is fundamental for reproducibility of the research results. An important aspect that impacts reusability is the ease of access to the corpus. Ideally, a corpus should be easily accessible by the research community and it should be well structured for an easy integration into the work flow. In CompMusic, we address the issues regarding reusability and sharing by emphasizing the use of open repositories that are either already suitable or can be easily adapted to our needs. Following this philosophy, we use MusicBrainz for organizing the editorial metadata. In addition, we make the corpus accessible through a RESTful Dunya web application programming interface (API).

The CompMusic corpora for IAM comprise two corpus: Carnatic music corpus and Hindustani music corpus. *Rāga* is the melodic framework and *tāla* is the rhythmic framework in both Carnatic and Hindustani music (Section 2.3.1). They are the two key musical concepts in IAM around which music is composed, performed, organized and taught. As a result, both corpora are compiled based primarily on these concepts. We now proceed to describe the specificities of both these corpora. The description

below includes the type of data that constitute a corpus, the unit of a data sample, selected references for measuring completeness and coverage of the data, and resources for obtaining complementary information about musical concepts.

3.2.2 Carnatic Music Corpus

The Carnatic Music corpus primarily comprises audio recordings and its associated editorial metadata. This is the data mainly used by the signal processing and machine learning approaches. In addition, there is a small collection of lyrics, scores, contextual information on music concepts, and community (social) information from online music forums used mainly for semantic analysis.

While building a representative music corpus there are several considerations specific to the music tradition that are taken into account. For Carnatic music, a concert, also referred as a (*kachēri*), is the natural unit of music performance. It is the unit typically considered for organization and digital distribution of Carnatic music content. Though Carnatic music is improvisatory in nature, it is predominantly based on compositions. Most of the compositions are to be sung, as a result of which, vocal music is dominant in Carnatic music. Even in instrumental music, the lead artist aims to mimic vocal singing (Viswanathan & Allen, 2004). Based on these considerations, we consulted expert musicologists and musicians such as Shri T. M. Krishna¹⁹ to arrive at a representative audio collection of Carnatic music.

The main institutional reference for building Carnatic music collection has been the **Madras Music Academy (MMA)**. The **MMA** was conceived to be the institution that would set the standards for Carnatic music. Since 1929, the **MMA** hosts annual conferences on music, which has eventually led to the December music festival of Madras, one of the largest cultural events of the world. The **MMA** has been driving the scholarly research in Carnatic music since a long time and has thus influenced the evolution of the musical concepts being used. The **MMA** has an expert panel that sets the standards and fixes procedures for selecting artists for the music festival. Since a long time the **MMA** has been recording Carnatic music performances and its archive is considered as one of the main references for Carnatic music. However, the archive is not openly available online. We therefore procured the audio recordings through other commercial sources while following the criteria used by the **MMA**. We started by collecting the releases of the artists who have performed in the **MMA** in the last five years. Subsequently, we expanded the collection to include their teachers and other musicians popular in their era.

One of the main record labels that specializes in Carnatic music recordings is Charsur²⁰, which has been publishing high quality commercial CDs for over 15 years now. The core of the Carnatic music audio collection is from their catalog of music concerts. The details of the corpus in terms of the unique number of recordings, releases,

¹⁹https://en.wikipedia.org/wiki/T._M._Krishna

²⁰<http://charsur.com/home.php>

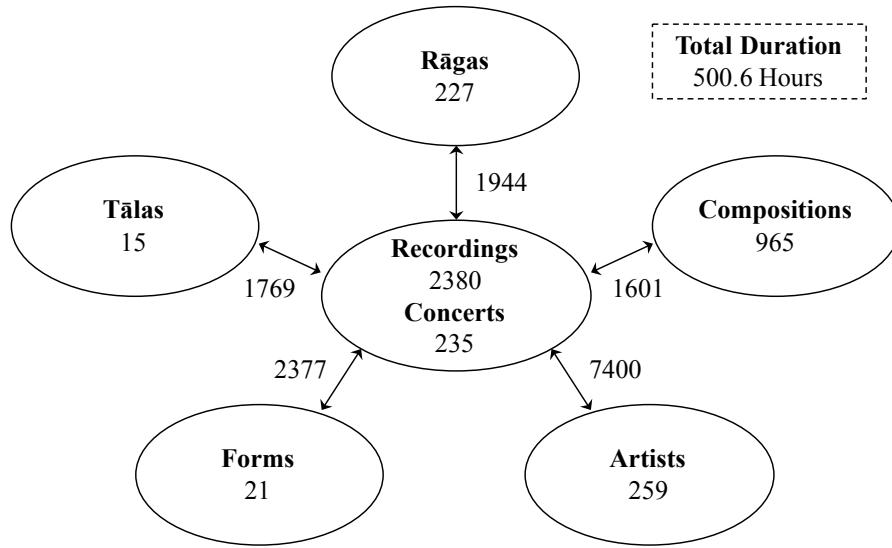


Figure 3.1: Details of the Carnatic music corpus in terms of the number of different musical entities and relationships between them.

artists, *rāgas*, *tālas* and compositions is shown in Figure 3.1. In total, the corpus currently consists of 235 concerts comprising 2380 audio recordings spanning over 500 hours of audio data.

As mentioned, vocal music constitutes a significant part of Carnatic music and it is largely based around compositions. Therefore, lyrics play an important role in this music tradition. Scores, on the other hand, have a limited usability as the music is primarily improvisatory in nature. Nonetheless, for some computational analyses lyrics and scores might be useful, and hence, we make an effort to compile a small collection of them. For most of the currently performed compositions there exists several published compilations of lyrics and scores, for example, the ones by the three most recognized composers: Tyagaraja (Rao, 1995b), Syama Sastri (Rao, 1997) and Dikshitar (Rao, 1995a). However, this data is not available in machine readable format and hence is not directly accessible for computational analysis. There are some open repositories of lyrics such as sahityam.net²¹ that provide lyrics in machine readable format. Sahityam.net is considered as the Wikipedia of lyrics of Carnatic music and is our primary source for lyrics. It currently hosts lyrics for about 1820 compositions of Carnatic music. Sources that provide music scores of Carnatic music in a machine readable format are scarce. A compilation of the scores done by Dr. Shivkumar Kalyanaraman²² is the main source of scores for Carnatic music in the CompMusic project.

²¹http://sahityam.net/wiki/Main_Page

²²<http://www.shivkumar.org/>

In addition to the signal processing and machine learning based approaches, semantic analysis of IAM has been another topic of research in the CompMusic project. The music community and music concepts related information collected from various sources over the Internet comprise the input data to semantic analysis and is a part of the Carnatic music corpus. Kutcheris.com²³ and Wikipedia²⁴ are two sources utilized for obtaining such an information. Kutcheris.com is a good source of artist biographies and up-to-date information about music venues, concerts and other related events. The category of Carnatic music on Wikipedia is a good source of contextual information including music concepts. There has also been an effort to contribute to Wikipedia by adding information with the help of domain experts. In addition to these two sources, we refer to rasikas.org²⁵, an active music forum for gathering views of the Carnatic music community on various facets of the music tradition. In the case of Carnatic music, the data from rasikas.org can be considered as ideal for studying tasks such as community profiling.

3.2.2.1 Coverage of Carnatic Music Corpus

Coverage analysis of a corpus aims to measure the representativeness and comprehensiveness of a corpus with respect to the reference sources that represent the music tradition. For the case of the Carnatic music corpus, coverage analysis is performed for rāgas, tālas, performing artists and composers. Kutcheris.com is our primary source for measuring artist coverage since it is up-to-date with current artists and their performances. We use the last five years of their concert listing from 2009-2014. Release catalog from Charsur, our main reference as a record label also provides information about rāgas, tālas, artists and composers. Raaga.com²⁶, an Indian music streaming service with a channel dedicated to Carnatic music is another source we considered for this analysis. It should be noted that Raaga.com has several light music forms included in their Carnatic music channel, which we have purposefully not included in our corpus. Hence, the numbers derived from an analysis done on the data from Raaga.com will have an adverse influence because of these additional music forms. The procedure followed for obtaining information from these sources and the pre-processing done before the analysis is explained in the article by Srinivasamurthy et al. (2014).

For each music entity we define a coverage measure, the *overlap* (O) as:

$$O_e^r = \frac{|X_e^c \cap X_e^r|}{|X_e^r|} \quad (3.1)$$

where O_e^r is the *overlap* measure of the musical entity e with respect to the reference source r , X_e^c is the set of entities in the corpus and X_e^r is the set of entities in the

²³<http://www.kutcheris.com/>

²⁴https://en.wikipedia.org/wiki/Category:Carnatic_music

²⁵<http://www.rasikas.org>

²⁶<http://play.raaga.com/carnatic>

	Corpus	Raaga.com	Kutcheris	Charsur
Rāgas	246	489 (42%)	NA	301 (68%)
Tālas	18	16 (100%)	NA	21 (85%)
Composers	131	598 (17%)	NA	256 (42%)
Artists	233	501	2978	264 (48%)

Table 3.1: Summary of the coverage of the Carnatic music corpus. The numbers in the parenthesis indicate the computed *overlap* measure. NA denotes not available.

reference source. $|X|$ denotes the cardinality of a set X . In Table 3.1 we summarize the coverage of the musical entities along with the overlap measure with respect to the reference sources mentioned above. We see that the coverage of the *rāgas* in the corpus is satisfactory and of the *tālas* is good. For composers and artists the numbers are low when compared to Raaga.com, which can be attributed to the presence of light Carnatic music forms in their database.

Not all the performing artists in the corpus (Table 3.1) are lead artists. Among these 233 artists 74 are lead artists (lead vocal or lead instrumental), 28 are accompanying violinists and 48 are percussionists. Since Carnatic music corpus predominantly comprises vocal music, coverage of lead or vocal artists becomes more important. Also not every lead artist is equally popular, that should also be a consideration in measuring the representativeness of the corpus. The concerts listed by Kutcheris.com span the whole year and all through the day. However, the evening concerts are more recognized, and we took that to be a measure of the popularity of an artist. For a better coverage analysis, we thus consider three categories of artists: Artists-Set-1 (all the artists), Artists-Set-2 (artists who have performed in the evening concerts, through the year) and Artists-Set-3 (artists who have performed in evening concerts between November and January). Of the 2978 total artists present in Set-1 on Kutcheris.com concert listings, there are 1814 artists in Set-2 and 1472 artists in Set-3.

In addition to the timing of the concerts, popularity of an artist can also be measured based on the number of concerts. Though there are a large number of artists listed in Kutcheris.com, we notice that the distribution of the number of concerts they have performed decreases exponentially (Figure 3.2). We see that there are only about 200 artists of 2978 artists who have performed in over 50 concerts. To consider this aspect while measuring coverage, we compute the *overlap* as defined in Eq. 3.1 through different subsets of the artists in Kutcheris.com, sweeping over the number of concerts they have performed. Furthermore, we perform this analysis for different categories of the artists in the corpus as mentioned above.

In Figure 3.3, we show the *overlap* between the artists in the corpus and the ones listed in Kutcheris.com for different sets of artists based on the number of performed concerts. We compute the *overlap* curve for all three categories of the artists in the corpus. We notice that the *overlap* increases as we consider the more frequently performing artists. We also observe that the *overlap* saturates and becomes constant. This can

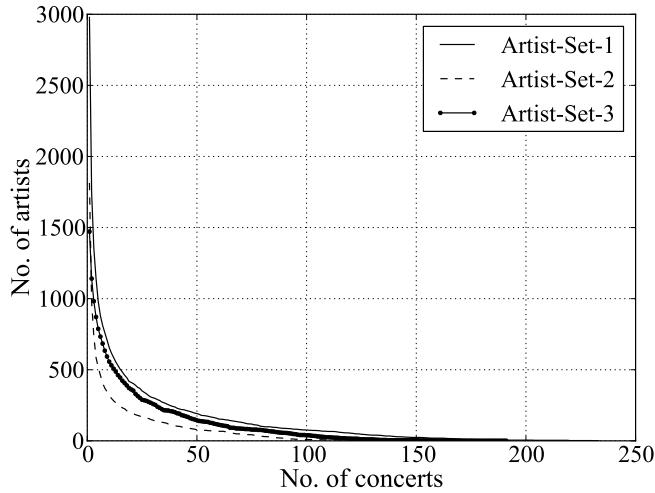


Figure 3.2: The number of artists versus number of their concerts. The information about these artists is obtained from Kutcheris.com.

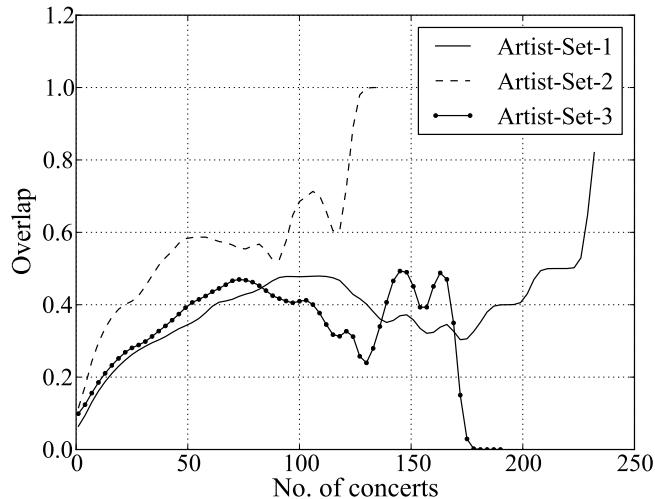


Figure 3.3: The *overlap* between the artists in the corpus and those listed in Kutcheris.com for different sets of artists determined by the number of performed concerts. The analysis is performed for every category of artists in the corpus.

be attributed to the fact that the most frequently performing artists are accompanists, and they are few when compared to the number of lead artists, as they accompany multiple lead artists. Since the number of artists with more than 150 concerts is very less, the *overlap* values become unreliable. Overall, we notice that, the *overlap* is better for artists in the category Artist-Set-2 than in Artist-Set-1 and in Artist-Set-3. This also indicates that the corpus has a better coverage of the artists from evening

concerts round the year.

3.2.2.2 Completeness of the Carnatic Music Corpus

As defined earlier, completeness of the corpus in the context of the CompMusic corpora refers mainly to the completeness of the associated metadata for each recording. The editorial metadata is stored at and accessed from MusicBrainz, as explained in Section 3.2.5. There can be multiple reasons for missing and erroneous editorial metadata in the MusicBrainz. Many times commercially released CDs do not provide all the relevant editorial metadata on the cover-art. In several cases there is no mention of the accompanying artists or of the *rāga* and *tāla* of the musical piece. Very often composition information is missing on the CD cover. Another reason for incomplete metadata can be that the editorial metadata is not completely entered into the MusicBrainz. This happens quite frequently for the fields such as recording relationships. Several times the metadata entered is erroneous. This is either due to a mistake done by the person uploading the metadata to the MusicBrainz or that the editorial metadata provided on the CD cover itself is wrong. Multiplicity of languages used in Carnatic music further adds to these inconsistencies. There has been some effort to automatically complete the missing metadata based on the relationships on the release and the recordings using semantic web approaches. The missing metadata due to transliteration errors also has been addressed to an extent by making curated list of entities such as *rāga* and *tāla*, and using robust algorithm for matching and linking metadata. Despite these efforts, there are still a number of recordings and releases for which the metadata is incomplete.

In Table 3.2, we show the completeness of the recordings in the Carnatic music corpus. We see that all the recordings are at least labeled with a lead artist, but about a quarter of the recordings (429/1650) do not have any accompanying artist information. *Rāga*, *tāla* and work (composition) labels are available for more than half the number of recordings. Note that there are some recordings that have the required editorial metadata but deemed incomplete because the names could not be accurately matched to any entity in the curated list.

3.2.3 Hindustani Music Corpus

Similar to Carnatic music, *rāga* and *tāla* are the fundamental music concepts with which to describe melodic and rhythmic aspects of Hindustani music. They thus become the primary consideration while building the Hindustani music corpus as well. In Hindustani music also, vocal music is predominant. However, compared to Carnatic music, the instrumental performances in Hindustani music are much more popular and prevalent. Hindustani music tradition as compared to Carnatic music is much more diverse and heterogeneous. One of the reasons for this can be the geographical spread of this music tradition. Hindustani music thus presents a significant challenge in compilation of a good research corpus. Another major difference

Metadata	#Recordings	completeness (%)
Lead artist	1650	100
Accompanying artist	1221	74
Rāgas	959	58
Tālas	917	56
Work (compositions)	989	60

Table 3.2: Completeness of the Carnatic music corpus showing the number of recordings for which the corresponding metadata is available and the percentage (%) of such recordings. The percentage values are rounded off to the nearest integer.

between the two music traditions is that in Hindustani music, the compositions are very short. The compositions basically act as a base for improvisation, which is the main focus in a performance of Hindustani music. For compilation of the Hindustani music corpus we focus on two prominent vocal music styles, *dhrupad* and *khyāl* (Section 2.3.1).

There are many institutions that have compiled large audio archives of Hindustani music. The prominent ones among them are the ITC Sangeet Research Academy (ITC-SRA), Sangeet Natak Academy, and the All India Radio (AIR). Each of these institutions own thousands of hours of expert curated music recordings that represent the performance practices in Hindustani music. ITC-SRA is a premier music academy of Hindustani music and has taken up major efforts in the archival of music. Sangeet Natak Academy is India's national academy for music, drama and dance. AIR is the largest public broadcaster in India and has a large archive of Hindustani music curated over many decades. AIR awards grades to performing musicians and its archives can be considered as a reference for Hindustani music. Like in most of the cases, none of these archives is publicly available. In such a situation, we gathered commercially released audio recordings from several music labels and compiled our own corpus using these institutions as a reference. During this process we also consulted expert musicians and musicologists, such as Dr. Suvarnalata Rao at the National Centre for the Performing Arts (NCPA), Mumbai, India, to curate the audio collection in the corpus.

The Hindustani music corpus primarily comprises *khyāl* and *dhrupad* vocal music releases, though a significant number of instrumental music releases are also present. There are 236 releases with over 1100 recordings spanning nearly 300 hours of audio data. The details of the Hindustani music corpus in terms of the unique number of recordings, releases, artists, *rāgas*, *tālas* and compositions is shown in Figure 3.4.

As mentioned before, compositions in Hindustani music are short and they basically act as a base for improvisation. The music performances mainly comprise improvised music material. Due to these factors lyrics and scores are not very relevant to the computational analysis of Hindustani music. There exists few repositories such as Bhatkhande (Bhatkhande, 1990) and Ramashray Jha (Jha, 2001) who have compiled

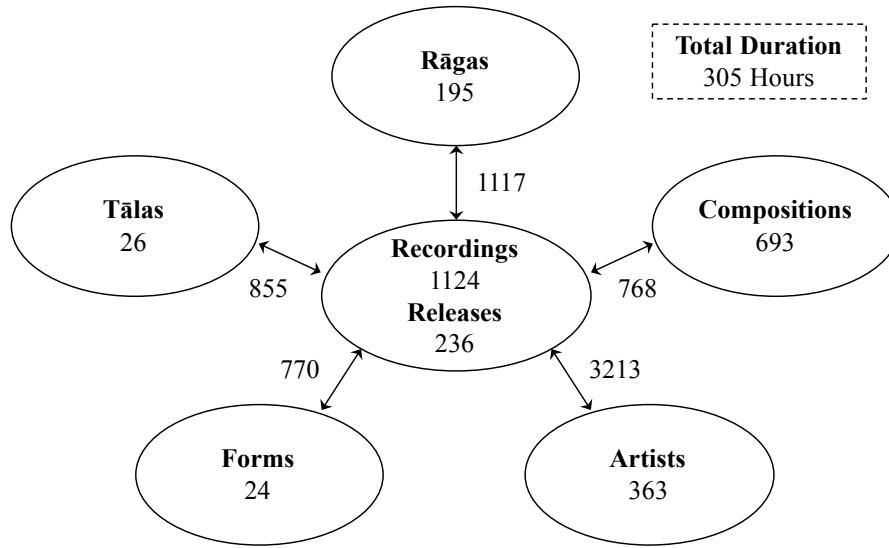


Figure 3.4: Details of the Hindustani music corpus in terms of the number of different musical entities and relationships between them.

lyrics and scores of several bandishes (compositions in Hindustani music) using a standard notation for Hindustani music. However, they are not available in a machine readable format. In addition to lyrics and scores, there are some repositories such as Swarganga Music Foundation²⁷ for musical concepts such as *rāga*, *tāla* and bandish. The category of Hindustani music on Wikipedia²⁸ is a good source of contextual information including music concepts of Hindustani music.

3.2.3.1 Coverage of Hindustani Music Corpus

In order to perform the coverage analysis of the Hindustani music corpus we follow the same methodology as that for the Carnatic music corpus. The analysis is done for artists, *rāgas*, *tālas* and compositions. Large geographical spread, lack of dedicated record labels and heterogeneous nature of the music make the coverage analysis of Hindustani music more complex compared to Carnatic music. Therefore, it is challenging to do a comprehensive artist coverage analysis like the one presented for Carnatic music. For each of the entities, we choose two main institutional references, ITC-SRA and Swarganga. In Table 3.3, we show the coverage of the Hindustani music corpus. We see that even though the corpus and the chosen references have comparable number of entities, the *overlap* between them is less. This can be attributed to the differences in the purpose of creating the music collection. We mainly focused on recordings made in the last 20-30 years to ensure good recording

²⁷<https://www.swarganga.org/>

²⁸https://en.wikipedia.org/wiki/Category:Hindustani_music

	Corpus	ITC-SRA	Swarganga
Artists	360	240 (19%)	629 (14%)
Rāgas	176	185 (48%)	534 (13%)
Tālas	32	N/A	59 (37%)
Works	685	N/A	1957

Table 3.3: Coverage of the Hindustani music corpus. The numbers in the parenthesis indicate the computed *overlap* measure. N/A denotes data not available.

Metadata	#Recordings	completeness (%)
Lead artist	1096	100
Accompanying artist	658	60
Rāgas	960	88
Tālas	627	57
Work (Bandish)	576	53

Table 3.4: Completeness of the Hindustani music corpus showing the number of recordings for which the corresponding metadata is available and the percentage (%) of such recordings. The percentage values are rounded off to the nearest integer.

quality and to reflect current performance practices. On the other hand, both the references focus primarily on archiving Hindustani music and hence consist of several generations of artists, infrequent rāgas and tālas, and a more comprehensive list of compositions. Furthermore, in our Hindustani music corpus we focus on vocal music recordings of only two styles, khyāl and dhrupad. The reference archives additionally include instrumental music and several other styles within Hindustani music.

3.2.3.2 Completeness of Hindustani Music Corpus

In Table 3.4, we show the completeness of the editorial metadata for Hindustani music. We see that the editorial metadata for all the recordings at least includes a lead artist, and for more than half of the collection, the accompanying artists. Roughly 90% of the corpus is annotated with rāga label and more than half with tāla label. Work (bandish) labels are present for nearly half of the collection. Ālāp performances in Hindustani music are completely improvisatory musical pieces and are not based on compositions. Also, they are unmetered in nature, and hence they are not assigned any tāla label. Ideally, such music pieces should be discounted while assessing the completeness of the work and the tāla metadata. However, due to the unavailability of the ālāp labels on these recordings, such performances are also included in the assessment, and hence, work and tāla completeness is an underestimate.

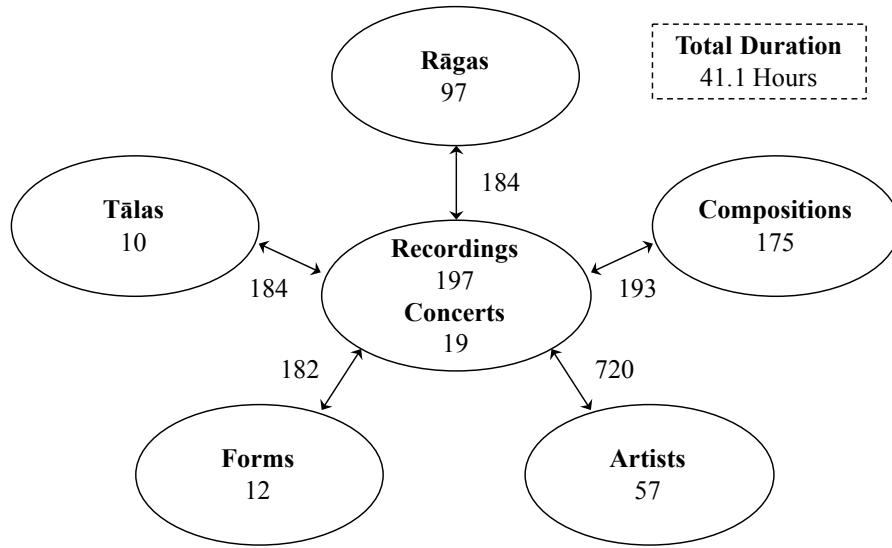


Figure 3.5: Details of the open-access Carnatic music corpus in terms of the number of different musical entities and relationships between them.

3.2.4 Open-access Music Corpus

The audio recordings in both Hindustani and Carnatic music corpus are ripped from commercially released music CDs. The copyright on these recordings does not allow for a redistribution of the music content, and therefore, they cannot be made publicly available. There has been an effort to compile an open-access music collection in the CompMusic project in order to promote the idea of open-access of the research corpora, and reproducibility of the research results. This open-access corpus comprises both Hindustani and Carnatic music. Like the other research corpora described in the previous sections, this corpus contains audio recordings and the associated editorial metadata. In addition, this corpus also contains several annotations of different music attributes such as melodic phrases, sama locations and sections. Due permissions are taken from the artists for redistribution of these audio recordings. As a result of which, the corpus is made publicly available under creative commons license (CC BY-NC 4.0) (Appendix B). The audio recordings in this corpus are hosted on Internet Archive²⁹ and are made accessible through the Dunya web API (Section 7.2).

The Carnatic music part of the open-access music corpus contains 19 releases comprising 197 recordings spanning over 41 hours of audio (see Figure 3.5 for details). A majority of these releases comprise recordings of the music concerts performed in the **Arkay Convention Center (ACC)** in Chennai, India. These are multi-track recordings later mixed and mastered by a professional. Individual music pieces of a concert are split into separate recordings that then constitute a release. In addition to the content

²⁹<https://archive.org/>

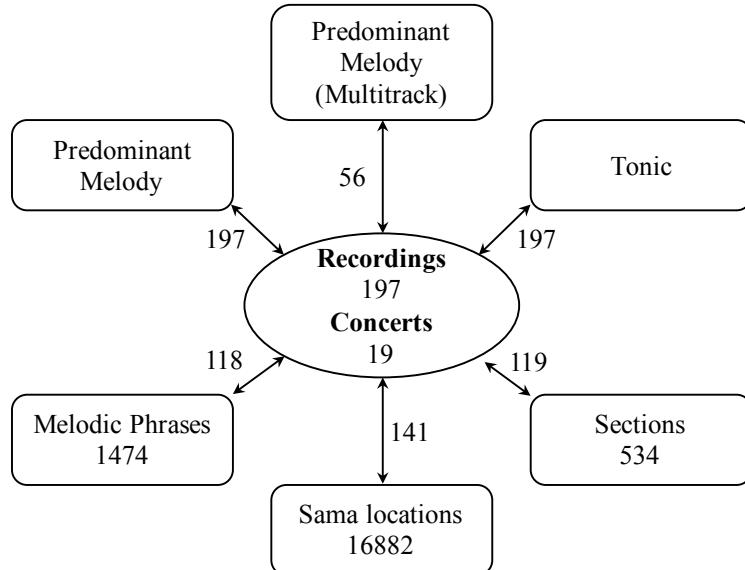


Figure 3.6: Details of the open-access Carnatic music corpus in terms of the number of annotations of different musical attributes and audio features.

procured from the ACC, a number of releases in this corpus are commercially released CDs with artists' due permissions to make them publicly available. Along with the audio recordings and the accompanying editorial metadata, this corpus contains carefully done annotations corresponding to the melodic, rhythmic and sectional aspects of the music. Manually done annotations include characteristic melodic phrases and sections within each audio recording. Semi-automatically done annotations include time-aligned sama locations and tempo in a recording. Finally, automatically extracted mid-level audio features include predominant pitch contour and tonic frequency used in the recording by the lead artist. Since for several concerts their multi-track recordings are available, along with the predominant pitch estimated from the mix-down track we also compute pitch using the solo vocal track. These pitch contours from the solo vocal tracks can serve as ground-truth to evaluate pitch estimation algorithms for IAM. In Figure 3.6, we show the number of available annotations of different musical attributes and audio features for the Carnatic music part of the corpus.

The Hindustani music part of the open-access music corpus contains 36 releases comprising 108 recordings that span 43.6 hours of music (see Figure 3.7 for details). A significant portion of this collection comprises commercial releases by maestros such as Pandit Ajoy Chakraborty and Pandit Kumar Gandharva. The other portion of the collection comprises individual recordings procured from different professional musicians and grouped together into meaningful releases for each artist. The grouping is based on the common performance practices, clubbing together the music pieces in the rāgas that are performed together in a concert. This corpus also contains annota-

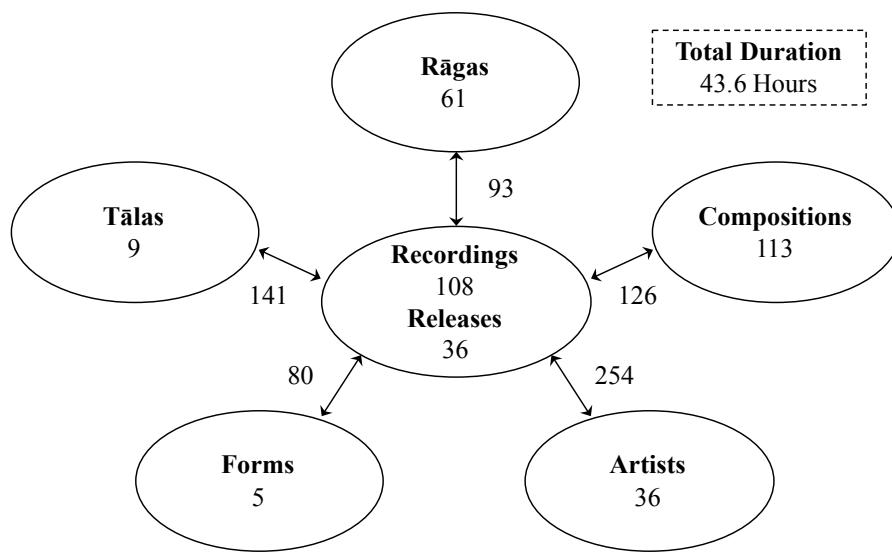


Figure 3.7: Details of the open-access Hindustani music corpus in terms of the number of different musical entities and relationships between them.

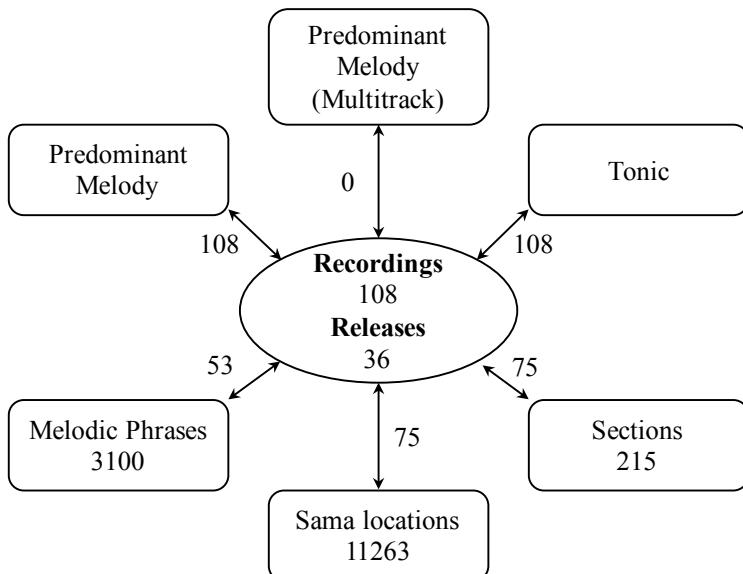


Figure 3.8: Details of the open-access Hindustani music corpus in terms of the number of annotations of different musical attributes and audio features.

tions for characteristic melodic phrases, sections, time-aligned sama locations and tempo. It also contains mid-level audio features, which include the predominant pitch contour and the tonic frequency for each recording. In Figure 3.8, we show the number of available annotations of different musical attributes and audio features for the Hindustani music part of the corpus.

3.2.5 Storage and Access of Corpus

The audio corresponding to the corpora are stereo recordings sampled at 44.1 kHz. For a few performances in the Carnatic open-access music corpus multi-track recordings are also available. Most of the audio recordings in the corpora are ripped from the commercially released CDs. These recordings are compressed and stored as 160 kbps MP3 files. The audio recordings for the open-access music corpus are hosted on the Internet Archive. The rest of the recordings are hosted on the servers in Universitat Pompeu Fabra. In addition to the audio data, the corpora also contain different melody and rhythm related audio features and annotations such as the predominant pitch contours, tonic and tempo value of the music recordings. These are also hosted on the university servers along with the audio recordings.

As mentioned before, every audio recording in the corpora is accompanied by its associated editorial metadata. It comprises the lead artist, accompanying artists, instrument played by these artists, *rāga* and *tāla* of the music piece, music form, and the name of the composition. For the case of Hindustani music it also includes the lay information. All this metadata is stored in MusicBrainz. Every music concept in MusicBrainz has a unique identifier, which facilitates the processing of this metadata. There are several other advantages of using the MusicBrainz repository for storing editorial metadata, such as its ability to publish its database into Linked Data³⁰, mapping of entity concepts to Music Ontology³¹, and its API for direct access to its data. Above all, it is a community driven open-content initiative and has its majority of the data in public domain.

All the data associated with an audio recording in the corpora: the audio file, extracted set of features, the accompanying metadata and annotations can be accessed through the Dunya platform³² (see Section 7.2). Dunya makes this data available in two ways: through a web-based graphical user interface and through a RESTful API. A detailed description of the ways to access the data is provided in Section 7.2. A direct link to several resources related with the corpora is provided in Appendix B.

³⁰<http://linkeddata.org/>

³¹<http://musicontology.com/>

³²<http://dunya.compmusic.upf.edu/>

3.3 Test Datasets

As explained earlier, a test dataset is a subset of a corpus used for studying a specific task. Typically a dataset is specific to an experiment and may contain additional information such as annotations. We here describe different datasets that are used to develop and evaluate computational approaches presented in this thesis. Though these datasets are compiled for studying specific research problems, some of them are essentially audio collections with comprehensive editorial metadata. And hence, we envision their usage in other melodic analysis tasks beyond the ones addressed in this thesis. Since a majority of these datasets are a subset of the corpora described in the previous sections (with the exception of the dataset described in Section 3.3.3), the details regarding the quality of the audio recordings and the editorial metadata remain the same. In case of exceptions, these details are provided in their respective sections.

3.3.1 Tonic Identification Datasets

We use six different datasets with varied musical characteristics for a comparative evaluation of approaches for tonic identification in **IAM**. The comparative evaluation is presented in Chapter 4. A summary of the tonic datasets in terms of different attributes of the comprising excerpts is provided in Table 3.5. In the subsequent paragraphs we describe each of these datasets in detail. Note that, all our datasets are made publicly available online for research purposes (Appendix B).

3.3.1.1 CompMusic Tonic Identification Datasets

The first three datasets shown in Table 3.5: **TID_{CM1}**, **TID_{CM2}** and **TID_{CM3}** are compiled by the author as a part of this thesis. They are derived from the Carnatic and the Hindustani research corpora described in the previous sections. These datasets contain audio excerpts, associated metadata (lead artist, lead instrument and gender of the lead artist in case of a vocal performance) and tonic frequency annotation for every audio recording. The main differences across these three datasets are in terms of the duration of the excerpts and the type of the music performance (vocal vs instrumental). These datasets comprise a diverse set of artists and music material such as gender of the singer, set of *rāgas* and music forms. Due to the diversity present in the datasets and the fact that these excerpts are taken from the commercial releases, they can be regarded as a representative collection of **IAM** performances for tonic identification.

TID_{CM1} and **TID_{CM2}** comprise three minute long audio excerpts extracted from full length recordings in the Hindustani and Carnatic music corpus. If a recording is longer than 12 minutes, we extracted 3 excerpts from the beginning, middle and end of the recording. If the recording was shorter than 12 minutes only one excerpt from the beginning was extracted. By taking excerpts from different sections of a song, we ensure that the datasets are representative, since the musical characteristics can change

Dataset	Avg. length (min)	#Excerpts	Hi. (%)	Ca (%)	Voc. (M/F %)	Inst. (%)	#Urec	#Uartists
TIDCM1	3	271	41	59	0	100	169	33
TIDCM2	3	935	45	55	100 (68 / 32)	0	547	81
TIDCM3	14.8	428	45	55	100 (72 / 28)	0	428	71
TIDITM1	144.6	38	0	100	89 (79 / 21)	11	N/A	22
TIDITM2	12.3	472	0	100	92 (77 / 23)	8	472	22
TIDuSc	7.4	55	0	100	100 (80 / 20)	0	55	5

Table 3.5: Summary of the tonic identification datasets, including average excerpt length (Avg. length), number of excerpts (#Excerpts), percentage of Hindustani music (Hi), Carnatic music (Ca), vocal excerpts (Voc.), instrumental excerpts (Inst.), number of unique recordings (#Urec) and number of unique artists (#Uartists) in each dataset. For vocal excerpts we also provide the breakdown into male (M) and female (F) singers. Percentage (%) values are rounded to the nearest integer.

significantly between different parts of a recording. TID_{CM1} contains exclusively instrumental performances, and does not overlap with TID_{CM2} and TID_{CM3} . The latter two contain only vocal performances, where TID_{CM3} contains full performances and TID_{CM2} contains excerpts taken from these performances. Overall, the total duration of the unique audio recordings in the instrumental dataset (TID_{CM1}) and the vocal datasets (TID_{CM2} and TID_{CM3}) is 35.5 and 132.5 hours, respectively.

The tonic pitch for the vocal performances and tonic pitch-class for the instrumental performances was manually annotated for each excerpt by the author. All the annotations were later verified by a professional Carnatic musician and the number of discrepancies was very small and later corrected. To assist the annotation process, we used the tonic candidate generation part of the approach proposed by Salamon et al. (2012). For every excerpt, the top 10 tonic candidates were synthesized and played together with the original audio file to help identify and label the correct candidate. Note that, the correct tonic pitch was always present amongst the top 10 candidates. A detailed description of this procedure is provided in Gulati (2012).

3.3.1.2 IITM Tonic Identification Datasets

Datasets TID_{IITM1} and TID_{IITM2} summarized in Table 3.5 are compiled by Bellur et al. (2012). These datasets were compiled by selecting 40 concerts from a private collection of hundreds of live concert recordings. These 40 concerts comprise 472 music pieces of Carnatic music. In order to study the robustness of tonic identification methods, the concerts that were selected range from artists from the 1960's to present day artists. The quality of the recordings vary from poor to good, usually depending on the period in which they were made. TID_{IITM1} comprises 38 of these full-length concerts. TID_{IITM2} comprises full length music pieces extracted from the 40 selected concert recordings. These performances are of varying duration, ranging from 46 seconds to 85 minutes. The tonic pitch for TID_{IITM1} and TID_{IITM1} was manually annotated by a professional Carnatic musician.

3.3.1.3 IISc Tonic Identification Dataset

Dataset TID_{IISc} is compiled by Ranjani et al. (2011). It comprises audio recordings obtained from an online Carnatic music archive³³. The archive is compiled by Carnatic musician and enthusiast Dr. Shivkumar Kalyanaraman for the benefit of music amateurs and hobbyists as an online educational resource. The archive includes various forms of Carnatic music. TID_{IISc} consists of 55 music pieces in the *ālāpna* form, recorded by five singers across seven *rāgas*. The total duration of the dataset is 6.75 hours. It includes recordings from the last 50 years, many of which were recorded live on analog audio tapes. The overall quality of the recordings is not very good. This makes it a challenging dataset for evaluating the accuracy of tonic identification

³³<http://www.shivkumar.org/music/index.html>

approaches. The tonic pitch for recordings in this dataset was manually annotated by two professional musicians, S. Raman and S. Vijayalakshmi.

These six datasets represent the diversities present in the **IAM** audio repertoire. To the best of our knowledge, these are the largest and the most comprehensive datasets available for studying the task of tonic identification in **IAM**.

3.3.2 Nyās Dataset

Nyās dataset (**NDD_{CM}**) is used for evaluating our proposed approach to identify **nyās svara** segments in melodies of **IAM** (Chapter 4). **NDD_{CM}** comprises 20 audio recordings of total duration of 1.5 hours. All these recordings are of vocal **ālāp** performances of Hindustani music. **Ālāp** is an unmetered melodic improvisatory section, usually performed at the opening of a **rāga** rendition. We selected only **ālāp** performances because the concept of **nyās** is emphasized in these sections during a **rāga** rendition. Of the 20 audio recordings, 15 are commercially available polyphonic recordings taken from the Hindustani music research corpus (Section 3.2.3). The other 5 audio recordings are monophonic in-house studio recordings done by a professional singer of Hindustani music. These in-house recordings are available under Creative Commons (CC) license in Freesound³⁴. In total, we have performances by 8 artists in 16 different **rāgas**. To the best of our knowledge, there does not exist any other dataset compiled and annotated for studying **nyās** segmentation in **IAM**.

Nyās segments were annotated by a performing artist of Hindustani music (vocalist) who has received over 15 years of formal musical training. The musician marked all the **nyās** segment boundaries in the recordings and labeled them appropriately. This dataset contains 1257 **nyās svara** segments. The duration of these segments vary from 150 ms to 16.7 s with a mean of 2.46 s and a median of 1.47 s.

3.3.3 Melodic Similarity Dataset

Melodic similarity dataset (**MSD**) is built for evaluating approaches for computing similarity between short-duration melodic fragments in **IAM**. Since the melodic characteristics across Carnatic and Hindustani music differ considerably, it is preferred to perform evaluations on each music tradition separately. Therefore, this dataset is divided into two parts: Carnatic music melodic similarity dataset (**MSD_{iitm}^{cmd}**) and Hindustani music melodic similarity dataset (**MSD_{iitb}^{hmd}**). Both these datasets contain audio recordings, bare minimum metadata (lead singer and **rāga** label for each recording) and time-aligned annotations of characteristic melodic phrases. **MSD_{iitb}^{hmd}** also contains the predominant pitch in the audio recording estimated using a semi-automatic approach proposed by Rao & Rao (2010); Rao et al. (2009). **MSD_{iitm}^{cmd}** was compiled at DONLab in Indian Institute of Technology Madras, Chennai, India. This dataset was introduced by Ishwar et al. (2013), and since then it has undergone several

³⁴<http://www.freesound.org/people/sankalp/packs/12292/>

Dataset	Rec.	PT	Rāgas	Artists	Duration (hr)
$\text{MSD}_{\text{iitm}}^{\text{cmd}}$	23	5	5	14	3.82
$\text{MSD}_{\text{iitb}}^{\text{hmd}}$	10	5	2	8	1.92

Table 3.6: Details of the melodic similarity datasets in terms of the total number of recordings (Rec.), number of annotated pattern types (PT), number of rāgas, unique number of artists and total duration of the dataset.

changes. $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ was compiled at Digital Audio Processing Lab in Indian Institute of Technology Bombay, Mumbai, India. This dataset was introduced by Ross et al. (2012), and has also undergone several changes since then. Both these datasets have been used in several studies for a similar task Ishwar et al. (2013); Ross et al. (2012); Rao et al. (2014). We describe and share the version of these datasets used in our experiments (Section 5.2).

$\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ comprise polyphonic vocal music recordings of renowned artists in both Carnatic and Hindustani music. In Table 3.6, we summarize the relevant details for both the datasets. We see that these datasets are diverse in terms of the number of artists.

The melodic phrases in **MSD** are annotated by two professional musicians (one for each music tradition) who have received over 15 years of formal music training. All the annotated phrases are the characteristic melodic phrases of the rāgas. These characteristic melodic phrases are distinctly recognized by musicians, as a result of which, the ambiguity involved in the judgment of melodic similarity is minimized. For both the datasets occurrences of five different melodic phrases are annotated. In Table 3.7, we summarize the relevant details for every category of the annotated melodic phrases for both the datasets. From the table we get an idea about the length of these phrases across their occurrences for each phrase type. In general, we see that the phrases in Hindustani music as compared to Carnatic music have a higher degree of variation in terms of their length.

In **MSD**, we discovered a number of discrepancies where several occurrences of the melodic phrases considered in these datasets were not annotated. This can potentially degrade the precision measurement of the method. When these cases were shown to musicians who annotated the dataset, they were surprised to know the errors in their annotations. Both the musicians commented that a possible reason why they missed marking such phrases was the melodic context in which these phrases occurred. When these phrases are played in isolation (without any melodic context) they appear to belong to the phrase categories considered in **MSD**. However, when played with the melodic context (i.e. including audio from a few seconds before the onset of the phrase), they are masked, and their segmentation and identification becomes harder. Many such missed occurrences of the melodic phrases were added to **MSD** to build

Dataset	PT	#Occ	L_{mean}	L_{std}	L_{median}	L_{min}	L_{max}
$\text{MSD}_{\text{iitm}}^{\text{cmd}}$	C_1	31	1.41	0.24	1.44	0.99	1.94
	C_2	33	1.28	0.21	1.26	0.91	1.91
	C_3	32	1.22	0.25	1.15	0.74	1.71
	C_4	26	1.12	0.17	1.06	0.9	1.6
	C_5	35	0.75	0.09	0.74	0.63	0.98
Overall		157	1.15	0.31	1.12	0.63	1.94
$\text{MSD}_{\text{iitb}}^{\text{hmd}}$	H_1	41	1.80	1.06	1.44	0.73	5.26
	H_2	139	1.33	0.74	1.22	0.38	5.23
	H_3	21	1.24	0.62	1.16	0.53	2.82
	H_4	61	2.25	1.30	1.74	0.51	5.93
	H_5	78	1.15	0.32	1.13	0.416	2.64
Overall		340	1.51	0.92	1.23	0.38	5.93

Table 3.7: Details of the annotated characteristic melodic patterns in **MSD** dataset. PT: phrase type, #Occ: number of annotated occurrences of patterns of a PT, and L_{mean} , L_{std} , L_{median} , L_{min} and L_{max} are the mean, standard deviation, median, minimum and maximum value of the lengths of the phrases of a PT (in seconds).

a revised version of the dataset, which we denote by **MSD_{CM}**. **MSD_{CM}** comprises exactly the same set of audio recordings as **MSD**. The only difference is in terms of the occurrences of the melodic phrases. In Table 3.8, we summarize the relevant details for every category of the annotated melodic phrases in both the datasets in **MSD_{CM}**. Comparing Table 3.7 and Table 3.8, we notice that in the new dataset nearly 25% of the melodic phrases are added.

3.3.4 Rāga Recognition Datasets

In our review of the existing studies on automatic rāga recognition we observed that every approach was evaluated on a different dataset (Section 2.4.3). In addition, there is no sizable dataset available, which can be shared and used across studies. We therefore build two sizable rāga recognition datasets, Carnatic music rāga recognition dataset (**RRD_{CMD}**) and Hindustani music rāga recognition dataset (**RRD_{HMD}**), which we make publicly available online (Appendix B).

As mentioned before, there are considerable differences in the melodic characteristics of Carnatic and Hindustani music. We therefore consider two separate datasets,

Dataset	PT	#Occ	L_{mean}	L_{std}	L_{median}	L_{min}	L_{max}
$\text{MSD}_{\text{CM}}^{\text{cmd}}$	C_1	39	1.38	0.25	1.35	0.87	1.94
	C_2	46	1.25	0.21	1.25	0.81	1.9
	C_3	38	1.23	0.24	1.16	0.74	1.7
	C_4	31	1.1	0.17	1.07	0.8	1.61
	C_5	45	0.76	0.08	0.76	0.62	0.98
Overall		199	1.14	0.3	1.12	0.63	1.94
$\text{MSD}_{\text{CM}}^{\text{hmd}}$	H_1	62	1.93	0.98	1.61	0.73	4.52
	H_2	154	1.4	0.8	1.22	0.38	5.23
	H_3	47	1.3	0.8	1.08	0.53	4.49
	H_4	76	2.38	1.34	1.89	0.5	5.93
	H_5	87	1.17	0.37	1.14	0.42	2.64
Overall		426	1.6	0.99	1.27	0.38	5.93

Table 3.8: Details of the annotated characteristic melodic patterns in MSD_{CM} dataset. PT: phrase type, #Occ: number of annotated occurrences of patterns of a PT, and L_{mean} , L_{std} , L_{median} , L_{min} and L_{max} are the mean, standard deviation, median, minimum and maximum value of the lengths of the phrases of a PT (in seconds).

one for each music tradition. RRD_{CMD} is a subset of Carnatic music research corpus (Section 3.2.2) and was introduced in Gulati et al. (2016b). It primarily consists of vocal performances. RRD_{CMD} comprises 124 hours of audio recordings and editorial metadata that includes carefully curated and verified *rāga* labels. It contains 480 recordings belonging to 40 *rāgas* with 12 recordings per *rāga*. In Figure 3.9, we show an overall summary of the relevant details of this dataset in terms of the different musical attributes. In Table C.3, we further provide the details for each constituent *rāga*. There are a total of 311 different compositions belonging to diverse forms in Carnatic music (for example, *kirtana*, *varnam*, *virtuttam*). In Table C.4 we list all the *rāgas* in RRD_{CMD} and their comprising set of *svaras*. We see that the chosen *rāgas* contain diverse set of *svaras*, both in terms of the number of *svaras* and their pitch-classes (*svarasthānās*). Several selected *rāgas* share a common set of *svaras*. This increases the complexity of the task, since the discrimination between these *rāgas* is mainly based on subtle melodic nuances and highly characteristic melodic phrases of *rāgas* (Section 2.3.2). To have diversity in the dataset we selected both the phrase-based and the scale-based *rāgas* (Krishna & Ishwar, 2012; Meer, 1980).

RRD_{HMD} is a subset of Hindustani music research corpus (Section 3.2.3) and was first

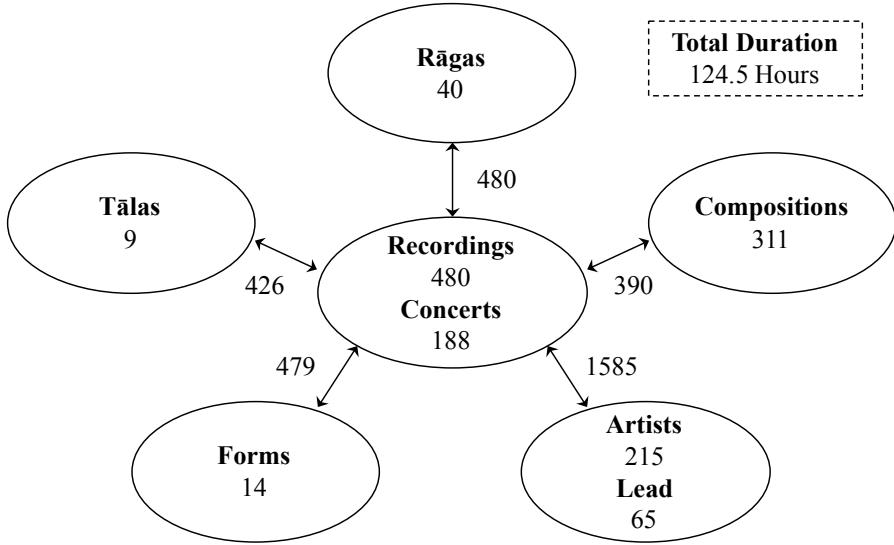


Figure 3.9: Details of the *rāga* recognition dataset (RRD_{CMD}) comprising Carnatic music recordings in terms of the number of different musical entities and relationships between them.

introduced in Gulati et al. (2016a). It comprises nearly 116 hours of audio recordings and editorial metadata. RRD_{HMD} contains full-length recordings of 300 Hindustani music performances belonging to 30 *rāgas* with 10 music pieces per *rāga*. In Figure 3.10, we show an overall summary of the relevant details of this dataset in terms of the different musical attributes. In Table C.3, we further provide the details for each constituent *rāga*. We see that the dataset is diverse in terms of the number of artists, the number of forms, and the number of compositions. In Table C.2 we list all the *rāgas* in RRD_{HMD} and their comprising set of *svaras*. Similar to the case in RRD_{CMD} , we see that in RRD_{HMD} as well, several chosen *rāgas* share a common set of *svaras*.

Given the diversities present in RRD_{CMD} and RRD_{HMD} , they can be regarded as the representative collections of IAM. To the best of our knowledge, these are the largest and the most comprehensive (in terms of the available metadata) datasets ever used for studying the task of automatic *rāga* recognition.

3.4 Summary

In this chapter, we described the music corpora and the test datasets that were compiled and curated as a part of our work within the CompMusic project. We started by enumerating the design criteria that we followed to gather and curate the corpora, and subsequently, evaluated the corpora in terms of these criteria. We saw that the musical concepts and entities in our corpora have a good coverage with respect to the reference institutes, and the corpora is representative of the audio collections in

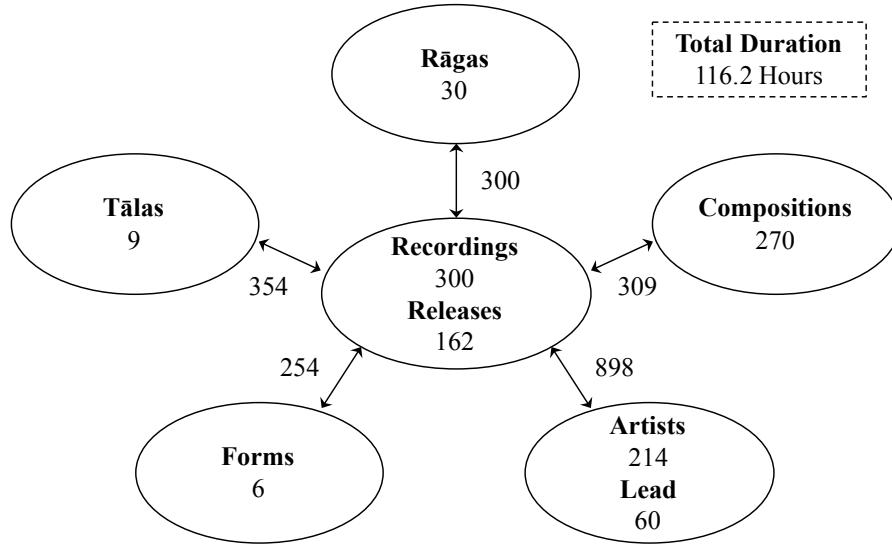


Figure 3.10: Details of the rāga recognition dataset (RRD_{HMD}) comprising Hindustani music recordings in terms of the number of different musical entities and relationships between them.

IAM. In addition to the corpora, we presented four different test datasets that we built to develop and evaluate the approaches described in this thesis. To the best of our knowledge, these are the largest corpora ever compiled for studying melodic aspects of **IAM** from a computational perspective. Both the corpora and the test datasets, are made publicly available ensuring easy access (Appendix B).

Chapter 4

Melody Descriptors and Representations

4.1 Introduction

In this chapter, we describe methods for extracting relevant melodic descriptors and low-level melody representation from raw audio signals. These melodic features are used as inputs to perform higher level melodic analyses by the methods described in the subsequent chapters. Since these features are used in a number of computational tasks, we consolidate and present these common processing blocks in the current chapter. This chapter is largely based on our published work presented in Gulati et al. (2014a,b,c).

There are four sections in this chapter, and in each section, we describe the extraction of a different melodic descriptor or a representation.

- In Section 4.2, we focus on the task of automatically identifying the tonic pitch in an audio recording. Our main objective is to perform a comparative evaluation of different existing methods and select the best method to be used in our work.
- In Section 4.3, we present the method used to extract the predominant pitch from audio signals, and describe the post-processing steps to reduce frequently occurring errors.
- In Section 4.4, we describe the process of segmenting the solo percussion regions (Tani sections) in the audio recordings of Carnatic music.
- In Section 4.5, we describe our *nyās*-based approach for segmenting melodies in Hindustani music.

4.2 Tonic Identification: Approaches and Comparative Evaluation

The tonic pitch of a lead artist in **IAM** is the base frequency that serves as a reference and foundation for melodic integration throughout the performance. All the tones in the musical progression are always in reference and related to the tonic pitch (Section 2.3.2). Therefore, for a meaningful comparison of melodies across recordings, it is important that the melodic representation is normalized by the tonic pitch in the recording. Identification of the tonic pitch in an audio recording thus becomes a crucial first step in melodic analysis of **IAM**.

There exist a number of approaches for identifying tonic pitch in an audio recording of **IAM** (Salamon et al., 2012; Gulati et al., 2012; Bellur et al., 2012; Ranjani et al., 2011; Sengupta et al., 2005; Chordia & Şentürk, 2013). We present a detailed review of these approaches in Section 2.4.1. In our previous studies on tonic identification we showed promising results with identification accuracies close to 90% (Salamon et al., 2012; Gulati et al., 2012). Similar accuracies are reported in other studies (Ranjani et al., 2011; Bellur et al., 2012). As seen in our review (Section 2.4.1), it is misleading to draw a consensus on the best performing method just based on the accuracy reported in the publications. It is because these approaches are evaluated using different datasets with varied musical characteristics and under different experimental setup.

In order to identify the most reliable, robust, and scalable approach, we perform their comparative evaluation using the same music collection and the same experimental setup. For this, we use a number of sizable datasets with varied musical and acoustical characteristics that are representative of the audio collections in **IAM**. In the subsequent sections, we describe the comparative evaluation, which is based on our earlier work presented in Gulati et al. (2014a). The experimental setup and the datasets used for the evaluation are described in Section 4.2.1. A detailed discussion of the results of the evaluation along with an in depth error analysis is presented in Section 4.2.2. We then explain a heuristic-based majority voting process to correct the frequently occurring errors in tonic identification (Section 4.2.4).

4.2.1 Comparative Evaluation Setup

We evaluate seven of the nine tonic identification methods listed in Table 2.1. The methods selected for evaluation are denoted by M_{JS} (Salamon et al., 2012), M_{SG} (Gulati et al., 2012), M_{RH1} and M_{RH2} (Ranjani et al., 2011), M_{AB1} , M_{AB2} and M_{AB3} (Bellur et al., 2012). Note that M_{RS} (Sengupta et al., 2005) was not available for evaluation and M_{CS} (Chordia & Şentürk, 2013) was not available when the experiments were conducted.

Each of the method mentioned above is evaluated using six different datasets, TID_{CM1} , TID_{CM2} , TID_{CM3} , TID_{IITM1} , TID_{IITM2} and TID_{IISc} . A detailed description of these datasets in terms the musical material and the audio quality is given in Section 3.3.1.

A summary of the datasets can be obtained from Table 3.5. The diversities present in IAM repertoire in terms of the musical and acoustical characteristics are well captured in these six datasets. Note that M_{AB1} method requires several excerpts from the same concert as an input, which is only available in the case of TID_{IITM1} dataset. Hence, M_{AB1} is only evaluated using TID_{IITM1} dataset.

For vocal performances we evaluate the accuracy of correctly identifying the tonic pitch, whereas, for instrumental music we evaluate the accuracy of identifying the tonic pitch-class only (i.e. the identified tonic pitch is allowed to be in any octave). This is because whilst for vocal music the concept of the tonic pitch being in a specific octave is clearly defined (because it is restricted by the pitch range of the singer), this notion is not as clear for instrumental music. For vocal performances, the tonic identified by a method is considered as correct if it lies within 50 Cents of the ground truth annotation. For instrumental music, output of a method is considered as correct if it is within 50 Cents of the correct tonic pitch-class.

Classification-based methods (M_{JS} and M_{SG}) are evaluated by using 10-fold cross-validation methodology on every dataset. The experiments are repeated 10 times and the mean accuracy over the 10 repetitions are reported. Parameters for all the methods are kept fixed across all datasets. Since the tonic pitch range for male and female singers, and for instrumental music is different, editorial metadata regarding the gender of the singer and the type of music excerpt (vocal or instrumental) can be used to improve the accuracy of tonic identification. We therefore perform two sets of experiments, first one, using only the audio excerpts, and the other, in which the methods are also given information about the gender of the singer and the type of excerpt (vocal or instrumental). The results of these evaluations are presented in the subsequent section. We remind again that a brief description of the methods considered in this evaluation is provided in Section 2.4.1.

4.2.2 Results and Discussion

In this section, we present the results of the comparative evaluation. We compare the performance of different tonic identification approaches, highlight their shortcomings and discuss various types of errors made by them. The section is divided into three parts. In Section 4.2.2.1, we present the results obtained when only the audio data is used and no additional metadata is provided to the methods. Subsequently, we report the performance accuracy obtained when information regarding the gender of the singer (male or female) and performance type (instrumental or vocal) is also provided to the methods in addition to the audio data (Section 4.2.2.2). Finally, in Section 4.2.2.3, we present an analysis of the most common mistakes made by the methods and make some general observations regarding their performances.

4.2.2.1 Results Obtained Using Only Audio Data

In Table 4.1, we summarize the identification accuracies (in percentage) for tonic pitch (TP) and tonic pitch-class (TPC) obtained by seven methods on six datasets, using only audio data.

From Table 4.1, we see that most of the methods perform well on all the datasets, and the accuracy of the best performing method on each dataset ranges from 84-97%. We note that the identification accuracy obtained for instrumental music (TID_{CM1}) by each method is comparable to the accuracy obtained for vocal music, meaning the approaches are equally suitable for vocal and instrumental music. The approaches based on multipitch analysis and classification (M_{JS} and M_{SG}) are more consistent and perform better across different datasets compared to the approaches based only on the predominant pitch (with the exception of TID_{IISc} , which is due its poor recording quality). We see that the performance of the two multipitch-based approaches is comparable. As could be expected, a simple maximum peak selection approach employed by M_{AB1} and M_{AB3} is too simplistic and the template matching approach employed in M_{AB2} yields better results in most cases.

M_{SG} obtains the best results for the instrumental dataset TID_{CM1} , with M_{AB2} and M_{JS} reporting comparable accuracies. For the TID_{CM2} and TID_{CM3} datasets, we see that the multipitch-based approaches (M_{JS} and M_{SG}) obtain the best performance, whilst the predominant pitch-based methods exhibit a considerable difference between the TP and TPC accuracies. This means that in many cases these approaches are able to identify the tonic pitch-class correctly but fail to identify the correct octave of the tonic pitch. In the case of M_{RH1} , M_{RH2} , M_{AB2} and M_{AB3} , this can be attributed primarily to the tonic selection procedure employed by these approaches. The group-delay processing used in M_{AB2} and M_{AB3} , and the estimators used in M_{RH1} and M_{RH2} , accentuate the peaks corresponding to all *svaras* that have a low degree of pitch variance. This includes both the lower and higher octave Sa and Pa *svaras* in addition to the middle octave Sa *svara* (the tonic pitch). Furthermore, the magnitude of the peaks corresponding to the Sa *svara* in higher and lower octave is sometimes further accentuated by pitch halving and doubling errors produced by the pitch extraction algorithm. This makes identification of the correct tonic octave more difficult, and as seen in Table 4.1, it results in a higher degree of octave errors.

Analyzing the results for TID_{IISc} dataset, we note that the performance drops for all the methods. The main reason for this is the poor audio quality of the excerpts in this collection. The recordings are relatively old and noisy, and contain a humming sound in the background. This makes pitch tracking very difficult. Furthermore, the drone sound in the recordings is very weak compared to the lead artist, which explains the drop in performance for the multipitch-based approaches. On the other hand, analyzing the results for TID_{ITM1} dataset, we see that all methods perform very well. This is because each excerpt in this dataset is a full concert, which includes many performances in different *rāgas*. Usually different set of *svaras* are used in different

Methods	TID _{CMI}		TID _{CM2}		TID _{CM3}		TID _{Disc}		TID _{ITM1}		TID _{ITM2}	
	TP	TPC	TP	TPC	TP	TPC	TP	TPC	TP	TPC	TP	TPC
M _{RH1}	-	81.4	69.6	84.9	73.2	90.8	81.8	83.6	92.1	97.4	80.2	86.9
M _{RH2}	-	63.2	65.7	78.2	68.5	83.5	83.6	83.6	94.7	97.4	83.8	88.8
M _{AB1}	-	-	-	-	-	-	-	-	89.5	89.5	-	-
M _{AB2}	-	88.9	74.5	82.9	78.5	83.4	72.7	76.4	92.1	92.1	86.6	89.1
M _{AB3}	-	86	61.1	80.5	67.8	79.9	72.7	72.7	94.7	94.7	85	86.6
M _{JS}	-	88.9	87.4	90.1	88.4	91	75.6	77.5	89.5	97.4	90.8	94.1
M _{SG}	-	92.2	87.8	90.9	87.7	90.5	79.8	85.3	97.4	97.4	93.6	93.6

Table 4.1: Accuracies (%) for tonic pitch (TP) and tonic pitch-class (TPC) identification by seven methods on six different datasets using only audio data. The best accuracy obtained for each dataset is highlighted using bold text. The dashed horizontal line divides the methods based on supervised learning (M_{JS} and M_{SG}) and those based on expert knowledge (M_{RH1}, M_{RH2}, M_{AB1}, M_{AB2} and M_{AB3}). TP column for TID_{CM1} is marked as ‘-’, because it consists of only instrumental excerpts for which we not evaluate tonic pitch accuracy. M_{AB1} is only evaluated on TID_{ITM1} since it works on the whole concert recording.

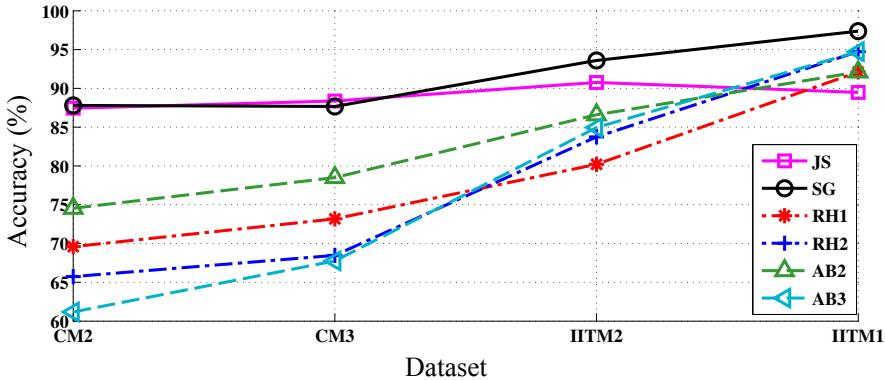


Figure 4.1: Accuracy (%) of different methods on four datasets arranged by increasing order of mean duration. Only the subscript in the names of the methods and datasets is used for labeling the axis and legends.

performances, but with the same tonic pitch throughout the concert. As a result, the pitch histogram contains a high peak corresponding to the Sa svara, which makes the identification of the tonic pitch easier.

We now proceed to analyze the tonic identification accuracy as a function of the excerpt duration. As shown in Table 3.5, different datasets contain audio excerpts of different lengths. In order to investigate a possible correlation between the accuracy of a method and the length of an audio excerpt, in Figure 4.1, we plot the identification accuracies of different methods for four of the six datasets ordered by the mean duration of the excerpts: $\text{TID}_{\text{CM}2}$ (3 min), $\text{TID}_{\text{CM}3}$ (full song), $\text{TID}_{\text{IITM}2}$ (full song) and $\text{TID}_{\text{IITM}1}$ (full concert). $\text{TID}_{\text{CM}1}$ and TID_{IISc} are excluded because the characteristics of these datasets are very different compared to the rest of the datasets ($\text{TID}_{\text{CM}1}$ contains only instrumental performances and TID_{IISc} has poor quality audio). As could be expected, from Figure 4.1 we notice that practically for all methods there is an improvement in the performance as we increase the duration of the excerpts. Interestingly, the improvement is very significant for the predominant pitch-based methods ($M_{\text{RH}1}$, $M_{\text{RH}2}$, $M_{\text{AB}2}$ and $M_{\text{AB}3}$) compared to the multipitch-based methods (M_{JS} and M_{SG}). This shows that the latter approaches, which exploit the pitch information of the drone instrument require less amount of audio data to perform this task. Notably, the accuracy of M_{JS} remains nearly the same across all the datasets, which is because the normalized distribution of the tones corresponding to the drone sound remains the same throughout the excerpt. Since this method does not utilize the predominant pitch information, it does not benefit much from the long duration audio excerpts. Note that, while the TP accuracy by M_{JS} is the lowest compared to the other methods on $\text{TID}_{\text{IITM}1}$ dataset, the TPC accuracy is amongst the highest.

In addition to analyzing the performance accuracy for the whole dataset, we also examine the results as a function of different musical attributes of a dataset, namely

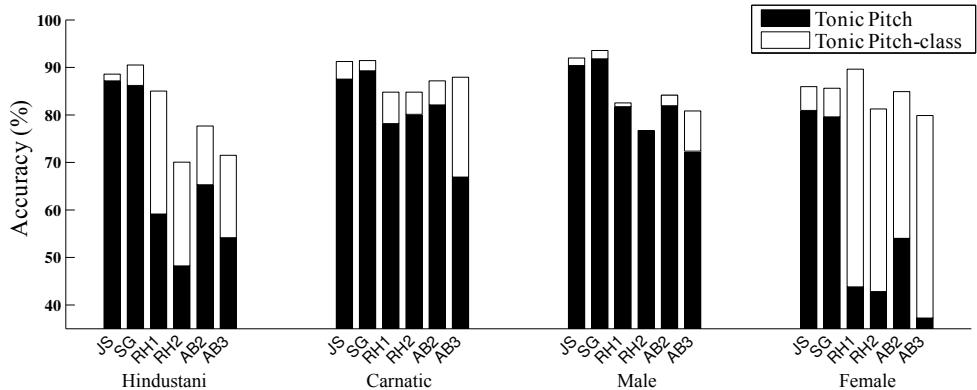


Figure 4.2: Accuracy (%) as a function of different attributes (Hindustani, Carnatic, male, female). Only the subscript in the names of the methods is used for labeling the axis.

music tradition (Hindustani or Carnatic) and the gender of the lead singer (male or female). For this analysis, we use the TID_{CM2} dataset, as it has the most balanced representation of excerpts from the different categories. In Figure 4.2, we show the accuracies obtained by the different methods as a function of the different attributes. We see that the performance of the multipitch-based approaches (M_{JS} and M_{SG}) is relatively independent of the music tradition (Hindustani or Carnatic). On the other hand, for the predominant pitch-based approaches there is a significant difference in performance for Hindustani and Carnatic music. They obtain considerably better results on Carnatic music. The most notable difference for these approaches is the increased amount of octave errors made for Hindustani music compared to Carnatic music (deduced from the difference seen in tonic pitch-class and the tonic pitch accuracies). A possible reason for this is that in the Hindustani recordings the *tānpura* is generally more salient compared to the Carnatic recordings. This results in the monophonic pitch estimators tracking the *tānpura* in some frames, in particular, when the lead artist is not singing. As a result, the pitch histogram includes high peaks at octave multiples or sub-multiples of the correct tonic pitch. In the case of M_{AB2} , M_{AB3} , M_{RH1} and M_{RH2} , most octave errors were found to be sub-multiples of the tonic pitch, possibly caused by the salient lower Sa played by the drone instrument.

Now we turn to examine the performance as a function of the gender of the lead artist (male or female). We see that in general, all the approaches function better for the performances by the male singers compared to those by the female singers. Similar to the case across music traditions, the difference is more significant for the predominant pitch-based methods, which make a large number of octave errors for the performances by the female singers. As noted earlier (Section 2.4.1.3), in methods M_{RH1} , M_{RH2} , M_{AB2} and M_{AB3} a range of 100–250 Hz is considered for finding the tonic pitch when no additional metadata about the artists is available. In the case of female singers, the tonic usually resides in the higher end of this range. However,

Methods	TID _{CM1}	TID _{CM2}	TID _{CM3}	TID _{IISc}	TID _{IITM1}	TID _{IITM2}
M _{RH1}	87.7	83.5	88.9	87.3	97.4	91.7
M _{RH2}	79.55	76.3	82	85.5	97.4	91.5
M _{AB1}	-	-	-	-	97.4	-
M _{AB2}	92.3	91.5	94.2	81.8	97.4	91.1
M _{AB3}	87.5	86.7	90.9	81.8	94.7	89.9
<hr/>						
M _{JS}	88.9	93.6	92.4	80.9	97.4	92.3
M _{SG}	92.2	90.9	90.5	85.3	97.4	93.6

Table 4.2: Accuracies (tonic pitch-class (%)) when additional information regarding the gender of the lead singer (male/female) and performance type (vocal/instrumental) is used. The dashed horizontal line divides the methods based on supervised learning (M_{JS} and M_{SG}) and those based on expert knowledge (M_{RH1}, M_{RH2}, M_{AB1}, M_{AB2} and M_{AB3}).

the presence of the drone, the tonal sounds produced by percussive instruments and the octave errors produced by the pitch tracker, all contribute to the appearance of a high peak one octave below the tonic of the female singers. This is especially the case for three minute excerpts, wherein a limited amount of vocal pitch information is available. In the case of the approaches based on multipitch analysis and classification (M_{JS} and M_{SG}), a probable reason for obtaining better performance for male singers is the larger number of excerpts from male singers in the database. As a result, it is possible that the rules learned by the classifier are slightly biased towards the performances of male singers.

4.2.2.2 Results Obtained Using Metadata Together with the Audio

One of the ways to reduce the amount of octave errors in tonic identification is to restrict the frequency range of the allowed tonic pitches. The frequency range can be optimized based on the additional information regarding the gender of the singer (when available) to guide the method. In this section, we analyze the effect of including information regarding the gender of the singer and the performance type (vocal or instrumental) on the identification accuracy obtained by the methods.

In Table 4.2, we present the identification accuracies obtained when associated metadata is available to the methods in addition to the audio data. Note for this evaluation we only report the tonic pitch accuracy for vocal excerpts (and not pitch-class accuracy) since when this metadata is available the pitch range of the tonic is known and limited to a single octave, meaning the TP and TPC accuracies will be the same.

Comparing the identification accuracies summarized in Table 4.2 with the ones in Table 4.1, we see that the accuracies for all methods are higher when gender and

performance metadata is available. With the additional information the performance of the predominant pitch-based approaches (M_{AB2} , M_{AB3} and M_{RH1}) becomes closer to that of the multipitch-based approaches (M_{JS} and M_{SG}). Whilst the performance of all methods is improved, the increase in accuracy is more considerable for the predominant pitch-based approaches which use template matching (in particular M_{AB2} and M_{AB3}) compared to classification-based approaches (M_{JS} and M_{SG}). This possibly indicate that the rules learned automatically using machine learning are more complete compared to the relatively simple Sa-Pa templates, meaning that the classification-based approaches can correctly identify the octave of the tonic even without using gender metadata. That is, since both male and female excerpts are used during training, the influence of the gender of the singer on the pitch features is implicitly learned by the classifier, thus producing rules that can handle both male and female performances, even without explicit metadata about the gender of the singer. On the other hand, manually defined template-based approaches require this extra information to fine-tune the frequency range considered for the tonic, after which they obtain comparable performance to that of the classification-based methods.

A potential advantage of the template-based approaches is that they do not require training. This, in theory, could make them more generalizable compared to the classification-based methods. To assess this, we ran an experiment in which the classification-based approaches were trained on one dataset and tested on a different dataset (TID_{CM2} and TID_{IITM2}). We found that the results only went down by approximately 2% compared to the results obtained using 10-fold cross validation on a single dataset. Furthermore, the datasets used for this experiment contained relatively different music material (percentage of Carnatic music excerpts and length of the audio files). This suggests that for tonic identification the rules learned by the classification-based approaches are generalizable and can be used to obtain high identification accuracies on diverse and sizable music collections of **IAM**.

4.2.2.3 Error Analysis

We now turn to analyze the different types of errors made by the methods, both with and without using additional metadata for each dataset. Overall, three common types of errors are identified: Pa errors, where the fifth (Pa) is selected instead of the tonic, Ma errors, where the fourth (Ma) is selected instead of the tonic, and the previously mentioned octave errors, where the correct pitch is identified but in the wrong octave (usually one octave above or below the tonic). Since the octave errors are already discussed at length in the previous paragraphs, here we focus on all other types of errors, which we divide into three categories: Pa (for Pa errors), Ma (for Ma errors) and “Other”, which includes all errors that are neither Pa, Ma nor octave errors (e.g. selecting the seventh ($N\bar{i}$) instead of the tonic Sa).

In Figure 4.3, for each dataset, we show the percentage of excerpts containing each of the three categories of errors for every method (when no additional metadata is

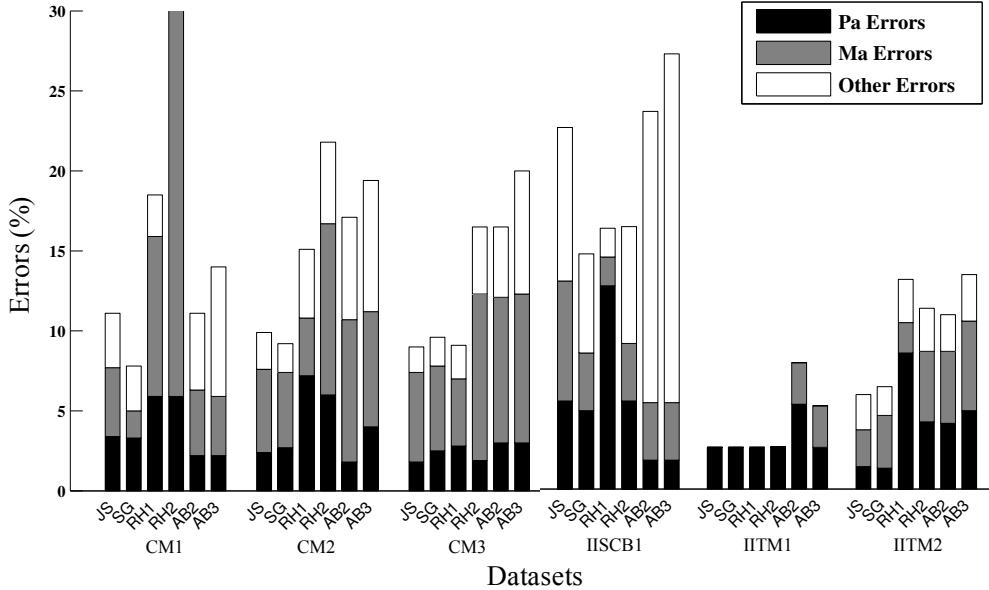


Figure 4.3: The percentage of audio excerpts containing each of the three different categories of errors (excluding octave errors): Pa, Ma and Other, when no additional metadata is used. Only the subscript in the names of the methods and datasets is used for labeling the axis.

used). We see that for most datasets Pa and Ma errors constitute a large proportion of the total amount of errors made by each method. These confusions make sense from a musical perspective, since in every performance of **IAM** one of these two **svaras** (Pa or Ma) is always present in a melody in addition to Sa (the tonic pitch-class). Furthermore, the pitch distance between Sa and Pa (fifth) is the same as the distance between Ma and higher Sa, and the pitch distance between Sa and Ma (one fourth) is same as the distance between Pa and higher Sa. Since most approaches are based on templates or rules that consider the pitch distance between the peaks of the feature histogram, these equivalences can cause four types of confusions: considering a Sa-Pa pair to be Ma-Sa leading to a Pa error, considering Ma-Sa to be Sa-Pa leading to a Ma error, considering Sa-Ma to be Pa-Sa leading to a Ma error and considering Pa-Sa to be Sa-Ma leading to a Pa error.

For the approaches based on multipitch analysis (**M_{JS}** and **M_{SG}**), we observe that the only case where we get more ‘Other’ errors compared to Pa and Ma errors is for the **TID_{IISCB}** dataset. Since the drone sound is very weak in the excerpts of this dataset, there are cases in which the prominent peaks of the multipitch histogram correspond to **svaras** other than Sa, Ma and Pa (which depends on the choice of the **rāga**). Since these approaches assume that the multipitch histogram represents the **svaras** of the drone instrument, the peaks of the histogram are mistakenly identified as Sa and Pa or Sa and Ma, leading to an error in identification. For these specific type of excerpts

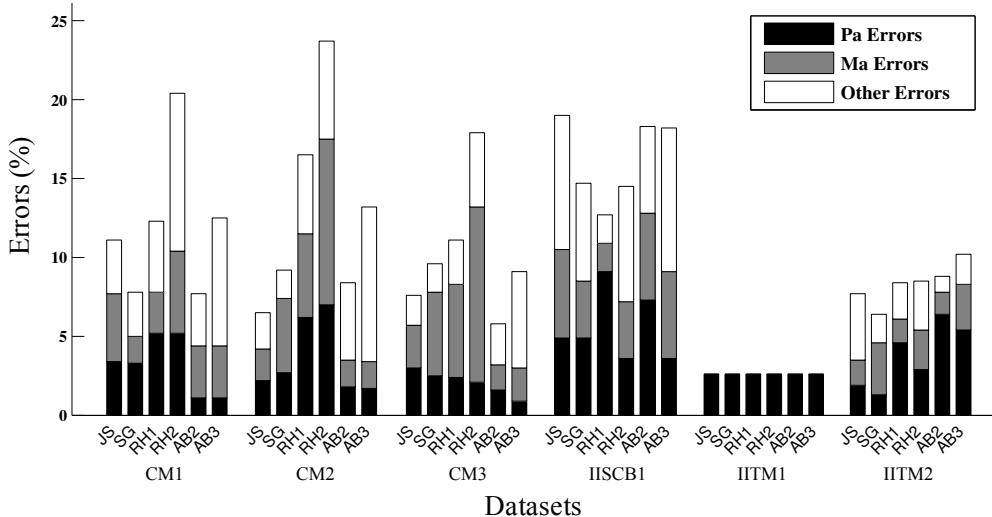


Figure 4.4: Percentage of different type of errors (Pa, Ma and Others) by different methods on all the datasets using information regarding the gender of the singer and performance type. Only the subscript in the names of the methods and datasets is used for labeling the axis.

the **M_{RH1}** method produces slightly better results, as the Sa svara is not inflected (i.e. there is little pitch variation within the **svara**) regardless of the **rāga**.

In many cases we observe that the percentage of Ma errors is greater than the percentage of Pa errors. For the classification-based approaches, this can be attributed to the fact that in most excerpts the drone instrument is tuned to Pa tuning (lower Sa, middle Sa, middle Sa, lower Pa). This creates a bias in the training set and the rules learned by the classifier work better for Pa tuning. Ma errors are also common in **M_{RH2}**, as the estimator looks for a Sa-Pa-higher Sa pitch relation, which would also fit a Ma-tuned performance. **M_{RH1}** on the other hand does not search for a Sa-Pa-Sa template, resulting in a low proportion of Ma errors compared to the other methods. Finally, we note that most methods do not make any Ma errors on the **TID_{IITM1}** dataset. This is because the items in this dataset are full concerts, each concert consisting of several pieces. Whilst Ma may be included in the melody of some of the pieces, Pa and Sa are always present. As a result, the pitch histogram for the complete concert does not contain a prominent Ma peak, meaning that it is highly unlikely for it to be selected as the tonic.

We now examine how the errors are affected once we allow the methods to use gender and performance metadata (Figure 4.4). In comparison to the method variants that do not utilize such metadata (Figure 4.3), we see that Ma and Pa errors are reduced more than “Other” errors. By restricting the tonic frequency range to a single octave we prevent the appearance of a high Sa peak, thus avoiding the possible confusion between fourths and fifths explained earlier and reducing the amount of Pa and Ma

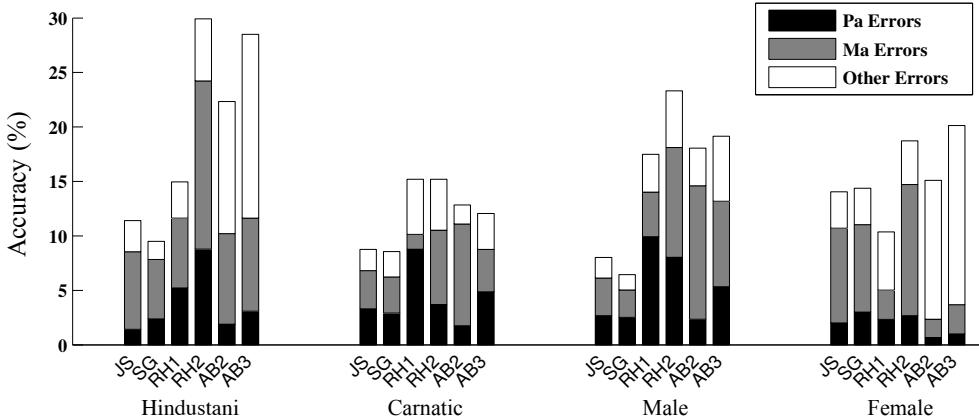


Figure 4.5: The percentage of audio excerpts with the different categories of errors (Pa, Ma and Others) for every method as a function of different excerpt attributes (Hindustani, Carnatic, male, female). Only the subscript in the names of the methods is used for labeling the axis.

errors.

For M_{RH1} and M_{RH2} the percentage of Ma errors actually increases slightly after including male/female information. A large proportion of these errors were observed in excerpts with female singers. For these excerpts, the range for *sadja* candidates is limited to 130-250 Hz. For this range, candidates fitting a lower Ma-middle Sa-middle Ma template would also satisfy the minimization criterion used in M_{RH2} . In the case of M_{RH1} , the reduced frequency range results in relatively weak peaks also being considered, and their small pitch variance can result in the wrong candidate being selected during the minimization process.

Finally, we analyze the errors as a function of the different attributes of the excerpts (Hindustani versus Carnatic, male versus female). As in Section 4.2.2.1, we use the TID_{CM2} dataset for this analysis because it is the most balanced dataset in terms of these attributes. Note that the methods are not provided with any metadata in addition to the audio signal. The percentage of excerpts containing each of the three categories of errors (Pa, Ma and Other) for every approach as a function of the different excerpt attributes is shown in Figure 4.5. We see that for the classification-based methods, the proportion of Ma errors is much higher in performances by female singers compared to performances by male singers. The pitch range of the tonic for female singers is such that the lower Ma resides in the frequency range where the tonic of most male singers lies. Thus, the lower Ma-middle Sa (fifth) relationship for female singers is often confused with middle Sa-Pa relationship for male singers, resulting in a high number of Ma errors. For further details and insights regarding the types of error made by the different methods and their underlying causes we refer the reader to the publications where these methods are discussed in depth (Salamon et al., 2012;

Gulati, 2012; Bellur et al., 2012; Ranjani et al., 2011).

4.2.3 Summary of the Comparative Evaluation

We evaluated seven tonic identification methods on six different and diverse datasets of **IAM**. The evaluation was performed in two scenarios: first, when only audio data is used as input to the methods, and second, when the additional information about the gender of the singer (male or female) and the performance type (vocal or instrumental) is given in addition to audio data. The results obtained are analyzed per dataset, and for different characteristics of the music material. Along with the accuracies of the methods, we presented an in-depth error analysis, describing the different kinds of errors for different scenarios and provided plausible explanations for them.

Overall, we see that methods **M_{JS}** and **M_{SG}** that use multipitch-based audio feature and classification methodology for selecting tonic pitch perform better than the rest. Their performance is better on all but one of the considered datasets (the exceptional dataset is small in size and contains poor quality audio recordings). In addition, the performance of these two methods appears to be the most consistent across music traditions, gender of the singer (male or female) and type of music performance (vocal or instrumental).

Finally, we select **M_{JS}** method to identify tonic pitch from audio recordings in all the studies conducted as a part of this thesis. **M_{JS}** produces comparable results to **M_{SG}** and is relatively simpler to implement owing to a single stage processing. Furthermore, **M_{JS}** does not require any estimate of the predominant pitch, which makes it independent of the performance of the pitch estimation algorithms. There are two implementations of this method, which we make publicly available online (Appendix B).

4.2.4 Correcting Common Errors in Tonic Identification

As mentioned above, identification of the tonic pitch is a crucial first step required for nearly all meaningful melodic analyses of **IAM**. Any error in this step propagates through the processing chain and adversely affect the output of the melodic analyses. Though the accuracy of the most successful tonic identification method is nearly 90% (Table 4.1), even a 10% error in tonic pitches can deteriorate the performance of melodic analyses described in the subsequent chapters. Moreover, it leads to an uncertainty, whether an error in the final output is caused due to a wrong tonic pitch or it is because of an error made in the later stages of the methods.

We here present a heuristic-based approach to correct frequently occurring errors in tonic identification. From Table 4.3, we see that the majority of the errors are Ma and Pa errors. We know from the music knowledge that the tonic pitch chosen by the lead performers in **IAM** does not change drastically over the years and it typically remains within two semitones (200 Cents). Since the accuracy of the tonic identification is nearly 90%, for the recordings of an artist in our music corpus there might only be a

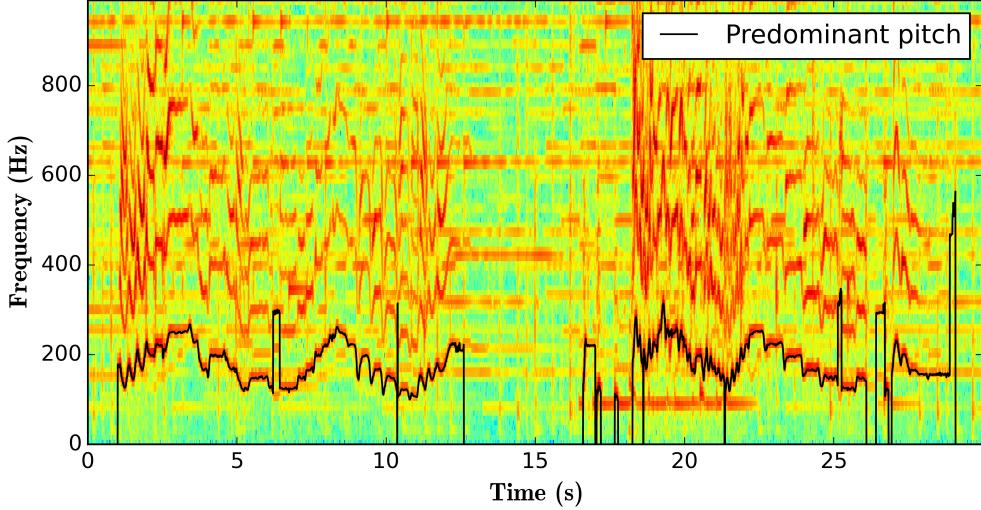


Figure 4.6: Example of a continuous pitch contour corresponding to the predominant melodic source extracted from a polyphonic audio excerpt.

handful of recordings for which the tonic values are incorrectly identified. Moreover, these wrong estimates are either Ma or Pa. These errors typically correspond to a pitch value that cannot possibly be the true tonic of the singer. Thus, these erroneous cases can be automatically detected by employing a majority voting across tonic values. Once detected, the wrongly estimated tonic values can be transposed (either by fifth, or fourth) to match the most frequent tonic value of that artist in the corpus. After we perform this step, we achieve close to perfect tonic estimates for the music collection.

4.3 Melody Processing

In this section, we present all the procedures applied in relation to the extraction and processing of a melody representation from raw audio signals. These processing steps can be grouped into three main categories: predominant pitch estimation (sometimes also referred to as predominant melody extraction), pitch post-processing, and melody representation. All these three steps are described at length in the subsequent sections.

4.3.1 Predominant Pitch Estimation

In Section 2.2, we presented our working definition of melody. In a nutshell, we consider the continuous pitch contour corresponding to the predominant melodic source (typically, the lead artist) in the audio recording as the low-level representation of melody. An example is shown in Figure 4.6, wherein the contour represents the pitch of the lead singer at each instance in time. Usage of such a melody representation is a common practice in MIR, and is done for a variety of melody processing tasks across

different music traditions (Dutta & Murthy, 2014a; Ishwar et al., 2013; Rao et al., 2014; Koduri et al., 2014; Şentürk et al., 2013; Pikrakis et al., 2012, 2003; Moelants et al., 2009).

Pitch estimation, also commonly known as pitch tracking from audio signals has been an active research topic since several decades (Salamon, 2013). The challenges in the estimation of a reliable pitch contour differ across the type of audio music signals (monophonic or polyphonic). As a result of which the performance of these algorithms vary significantly across different music genres, which also seems to be correlated with the extent of the polyphony in the music. In comparison to several western popular music genres, a fewer number of simultaneously playing instruments, and the prominence of the voice of the lead artist in performances of **IAM** lessens the complexity of estimating the predominant pitch from audio recordings. This trend is clearly visible from the past MIREX (an international MIR evaluation campaign) results³⁵. For example, compare the accuracy obtained by different algorithms on INDIAN08³⁶, MIREX05³⁷ and MIREX09 0dB³⁸ datasets from MIREX-2011.

In order to estimate the predominant pitch (denoted hereafter by p) in the audio recordings of **IAM** we use **Melodia** algorithm, a state-of-the-art melody extraction method proposed by Salamon & Gómez (2012). This method performed favorably in MIREX 2011 on a variety of music genres, including **IAM**³⁹. This method is used in several other studies that analyze melodies extracted from audio signals (Dutta & Murthy, 2014a; Ishwar et al., 2013; Rao et al., 2014; Koduri et al., 2014; Şentürk et al., 2013; Pikrakis et al., 2012).

Another motivation to use this algorithm is that it works on continuous pitch contours and employs auditory streaming constraints to ensure a continuity in the output pitch. As a result of this processing, it does not produce octave errors at the frame-level. Noticeably, this predominant pitch estimation algorithm also performs voicing detection. This means that the algorithm in addition to estimating the predominant pitch also detects the time segments where the voice is absent.

Currently, there are two implementations of **Melodia** algorithm that are publicly available. We use its implementation as available in **Essentia** (Bogdanov et al., 2013). **Essentia**⁴⁰ is an open-source C++ library for audio analysis and content-based MIR. We use the default values of the parameters, except for the frame and hop sizes, which are set to 46 and 2.9 ms, respectively. The other implementation of **Melodia** is available as a Vamp plug-in⁴¹.

³⁵http://www.music-ir.org/mirex/wiki/MIREX_HOME

³⁶http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/indian08/summary.html

³⁷http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/mirex05/summary.html

³⁸http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/mirex09_0dB/summary.html

³⁹http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/indian08/summary.html

⁴⁰<https://github.com/MTG/essentia>

⁴¹<http://mtg.upf.edu/technologies/melodia>

Prior to the predominant pitch estimation we apply an equal-loudness filter⁴² (Bogdanov et al., 2013) to make the processing perceptually relevant. For this operation as well we use Essentia with the default values of the parameters.

In every computational task addressed in this thesis we use the Melodia algorithm to estimate the predominant pitch for all the datasets. There is only one exception, the MSD_{itb}^{hmd} dataset, which was introduced in Ross et al. (2012) for computing melodic similarity. The MSD_{itb}^{hmd} dataset, along with its audio recordings and melodic phrase annotations, also includes semi-automatically extracted predominant pitch contours (Section 3.3.3). Using the pitch contours provided along with the dataset allows us to compare the output of our method with other studies. In addition, since these pitch contours are nearly free from any octave errors, we avoid propagating errors to subsequent stages, and thus, evaluate the task of melodic similarity more reliably.

In this thesis, we consider only the pitch dimension of melody in its representation. However, loudness and timbral dimensions of melody that largely capture the phonetics and expressive aspects are also informative and can be helpful in several tasks including computation of melodic similarity. To give an idea about the type of information captured in the loudness and timbral dimensions and their usefulness, we take an example. We synthesize the harmonic series corresponding to a voice extracted from an excerpt of Carnatic music. During the synthesis we force the pitch of the voice to a monotone while keeping the other aspects (loudness and timbre) intact. The original excerpt⁴³, synthesized predominant pitch⁴⁴ and the synthesized monotone voice⁴⁵ is made available for listening. After listening to this example, we get an idea about the nature of the melodic characteristics that are not utilized if we only consider the pitch dimension of melody in the analysis. Exploiting loudness and timbral dimensions for melodic analysis shall be taken up in the future endeavors.

4.3.2 Pitch Post-processing

Predominant pitch estimation in polyphonic audio recordings has not been solved yet, and the output pitch contour is still far from being perfect (MIREX-2011 results⁴⁶, MIREX-2016 results⁴⁷). While these algorithms get better over years, a number of errors in the predominant pitch estimation can be alleviated by post-processing the pitch contour. In post-processing, our main objective is to correct the spurious pitch octave jumps and smoothen the extracted pitch contour. In addition, for certain tasks described in the subsequent chapters, we also interpolate the unvoiced regions in the

⁴²http://wiki.hydrogenaud.io/index.php?title=ReplayGain_1.0_specification

⁴³<http://www.freesound.org/people/sankalp/sounds/352810/>

⁴⁴<http://www.freesound.org/people/sankalp/sounds/352809/>

⁴⁵<http://www.freesound.org/people/sankalp/sounds/352808/>

⁴⁶http://nema.lis.illinois.edu/nema_out/mirex2011/results/ame/indian08/summary.html

⁴⁷http://nema.lis.illinois.edu/nema_out/mirex2016/results/ame/ind08/summary.html

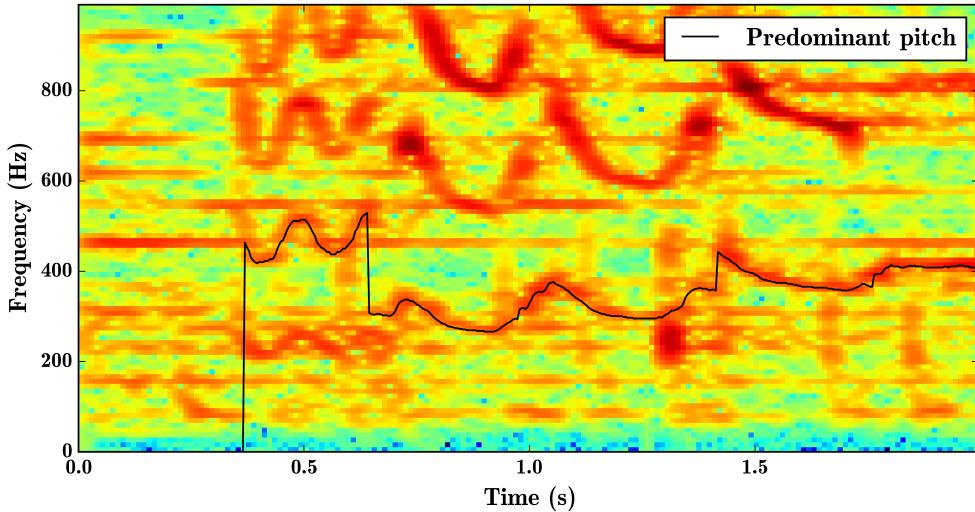


Figure 4.7: Example of an octave error in predominant pitch contour.

melody that correspond to the short-duration breath pauses taken by the artists. We now describe these post-processing steps in detail.

4.3.2.1 Correcting Spurious Pitch Jumps

One of the most frequently occurring errors in pitch estimation is its detection in a wrong octave, commonly referred to as an octave error. Identifying and correcting such errors during a post-processing step for the case of a monophonic signal (containing a single harmonic series) is fairly simple. One can compare the energy of the partials with the total energy of the audio frame to infer if the estimated pitch value corresponds to the actual fundamental frequency. However, in the case of a polyphonic music signal it becomes a challenging task.

We alleviate this problem to an extent by restricting ourselves to a specific type of pitch octave error occurring over a short duration of time. Instead of identifying an octave error for an individual audio frame, we exploit the temporal melodic continuity to detect such errors. We explain the intuition behind our method with the help of an example shown in Figure 4.7. If we analyze an isolated audio frame at 0.5 s, identifying a pitch octave error computationally becomes a challenging task. However, by analyzing the pitch continuity we can easily detect an anomalous pitch jump of roughly 1200 Cents at time 0.65 s. This is due to the fact that such drastic pitch jumps across frames do not occur naturally in singing voice. By analyzing the amount of frequency difference across the pitch transition we can infer to an extent the type of pitch error and can subsequently correct it. Since the detected pitch jump is around 1200 Cents, it is highly likely to be an octave error. In this example voice starts at around 0.4 s, which we consider as a positive pitch jump. Knowing that there is a

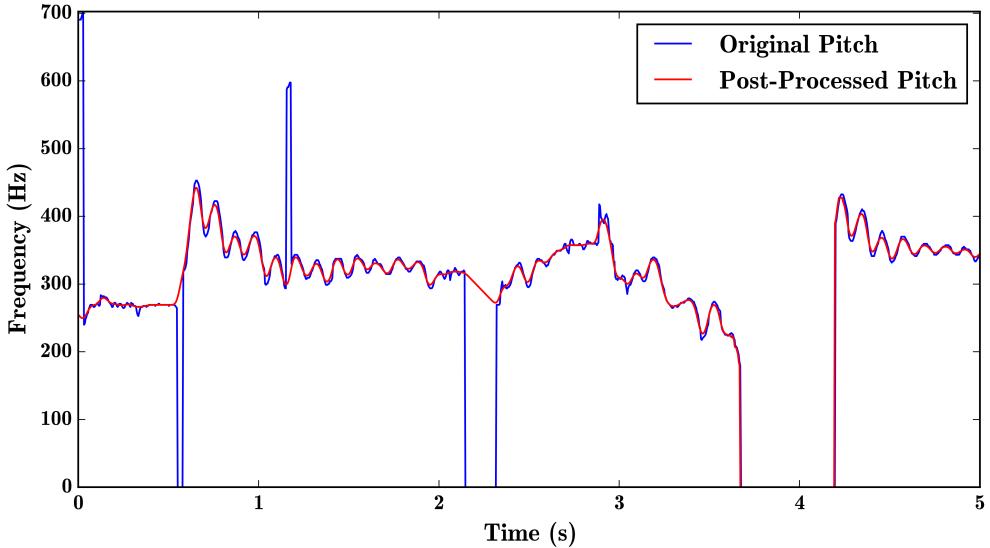


Figure 4.8: Example of a pitch segment before and after post-processing, which involve median filtering, Gaussian filtering and interpolation of the short unvoiced regions.

omalous pitch jump in the other direction (negative pitch jump) at time 0.65 s makes it highly probable that the pitch segment between time 0.4 s and 0.65 s suffers from an octave error.

We describe this heuristic-based approach in Algorithm 2. In this algorithm `windowSize` is set to 1 s and `silenceCentValue` is set to $1200 * \log_2(\varepsilon)$, where ε is of the order of 10^{-17} .

4.3.2.2 Pitch Smoothening

This processing step aims to remove the spurious pitch jumps lasting over a few frames and to smooth the pitch contours. We start by performing a median filtering on the estimated pitch contour. The window length chosen for median filtering is 50 ms. Subsequently, to smooth the pitch contour we apply a low-pass filtering by using a Gaussian window. The window size and the standard deviation of the Gaussian window is set to 50 ms and 10 ms, respectively. In Figure 4.8 we show an example of a pitch segment before and after applying median filtering and Gaussian smoothening. We see that the spurious pitch jumps are removed and the pitch contour appears smooth.

4.3.2.3 Pitch Interpolation

In the rendition of melodic phrases in IAM, there are often unvoiced segments lasting over a small time interval. These unvoiced segments may either correspond to

Algorithm 2 Correcting spurious pitch octave jumps

Input: pitch sequence (p) of length N samples in Cents scale
 $\text{jumpType} = \text{zeros}(N)$

```

for  $ii=0$ ;  $ii < N$ ;  $ii++$  do                                // Detecting type of pitch transition
     $diff = p[ii+1] - p[ii]$ 
    if  $abs(diff) \% 1200 \leq 300$  then
        if  $diff > 0$  then
             $\text{jumpType}[ii+1] = 3$                          // Positive octave jump
        else if  $diff < 0$  then
             $\text{jumpType}[ii] = 4$                            // Negative octave jump
    else if  $abs(diff) \geq 600$  then
        if  $p[ii] = \text{silenceCentValue}$  then
             $\text{jumpType}[ii+1] = 1$                          // Unvoiced to voiced
        else if  $p[ii+1] = \text{silenceCentValue}$  then
             $\text{jumpType}[ii] = 2$                            // Voiced to unvoiced
        else if  $diff > 0$  then
             $\text{jumpType}[ii+1] = 5$                          // Other positive pitch jump
        else
             $\text{jumpType}[ii] = 6$                            // Other negative pitch jump
    for  $ii=0$ ;  $ii < N - \text{winSize}$ ;  $ii++$  do          // Fixing pitch octave jumps
         $shift = 0$ 
         $indJumps = \text{where}(\text{jumpType}[ii:ii+\text{winSize}] \neq 0)$ 
        if  $\text{len}(indJumps) == 2$  then                      // Process only when two jumps are detected
             $i1 = indJumps[0] + ii$ 
             $i2 = indJumps[1] + ii$ 
            if  $\text{jumpType}[i1] + \text{jumpType}[i2] == 3$  then
                 $\text{jumpType}[i1] = 0$ 
                continue                                     // Do nothing for unvoiced to voiced
            if  $\text{jumpType}[i1] \% 2 = 1$  and  $\text{jumpType}[i2] \% 2 = 0$  then
                if  $\text{jumpType}[i1] == 3$  and  $\text{jumpType}[i2] \neq 6$  then
                     $shift = 1200 * \text{round}((p[i1] - p[i1-1]) / 1200)$ 
                else if  $\text{jumpType}[i2] == 4$  and  $\text{jumpType}[i1] \neq 5$  then
                     $shift = 1200 * \text{round}((p[i2] - p[i2+1]) / 1200)$ 
                else if  $\text{jumpType}[i1] == 1$  then
                     $shift = 100 * \text{round}((p[i2] - p[i2+1]) / 100)$ 
                else if  $\text{jumpType}[i2] == 2$  then
                     $shift = 100 * \text{round}((p[i1] - p[i1-1]) / 100)$ 
             $p[i1:i2+1] = p[i1:i2+1] - shift$ 
             $\text{jumpType}[i1] = 0$ 
             $\text{jumpType}[i2] = 0$ 

```

short breath-pauses taken by the vocalists or to the consonants in the lyrics, which are unvoiced in nature. These short unvoiced segments pose a difficulty in the computation of melodic similarity since they do not exist in all the occurrences of a melodic phrase. In such situations assignment of a meaningful numerical value to these unvoiced melodic regions is desired. Therefore, in order to avoid the complexities arising from these unvoiced melodic segments, we interpolate these regions. We perform a linear interpolation across all the unvoiced segments that last for less than 300 ms. An example of an interpolated unvoiced segment is shown in Figure 4.8, between time range of 0 to 1 s, and 2 to 3 s.

4.3.2.4 Pitch Resampling

The optimal sampling rate of the predominant pitch might depend on the particular computational task under study. For tasks such as analysis of melodic ornaments in IAM, a high sampling rate might be desired (for example, 5 ms), whereas, for computationally complex tasks such as melodic pattern discovery and search, a sampling rate as low as possible but sufficient to capture melodic nuances might be preferred. In order to avoid predominant melody computation for different sampling rates used across experiments, we resample the predominant melody contours extracted once at a high sampling rate. Pitch contours are decimated to a lower sampling rate by simply downsampling them by an integer factor. The downsampling factor is decided based on the sampling rate used in a particular experiment.

Note that, not all the post-processing steps described in this section are employed in all the experiments. The description of the specific methods in the subsequent chapters will contain details about the post-processing steps used in their respective experiments.

4.3.3 Melody Representation

4.3.3.1 Hertz to Cent Conversion

The perception of melodic intervals for human beings is logarithmic in nature with respect to pitch (or fundamental frequency in Hz). Thus, in order for the pitch representation to be musically meaningful, we convert the estimated predominant pitch values from Hertz to Cents (logarithmic scale) following the equation below.

$$\hat{p}_i = 1200 \log_2 \left(\frac{p_i}{f_r} \right), \quad (4.1)$$

where p_i is the i^{th} sample of the predominant pitch in Hertz-scale, \hat{p}_i is the i^{th} sample of the predominant pitch in Cent-scale, f_r is the reference tuning frequency which typically in the case of western popular (or even classical) music is set to an integer multiple or sub-multiple of 440 Hz. We consider $f_r = 55$ Hz, although as we will notice in the next section that this choice is inconsequential.

4.3.3.2 Tonic Normalization

As described in Section 2.3.2 and in Section 4.2, in a performance of **IAM**, the tonic pitch of the lead performer serves as the reference frequency. Every lead artist chooses a tonic pitch, using which the **tānpura** and the rest of the instruments are tuned. To analyze a melody in the tonal context established by the tonic pitch in a recording (\mathcal{T}), we normalize the predominant pitch by this frequency. We perform this normalization by considering the reference frequency $f_r = \mathcal{T}$ in Eq. 4.1, which is estimated for every recording using **M_{JS}** method (Section 4.2.3).

4.4 Tani Segmentation

A concert of Carnatic music typically contains a solo percussion section towards the end of the concert, referred to as **tani avartanam** or **tani** in short. The duration of this section typically varies from 2 to 25 min depending on the artist and the context. The main percussion instrument in Carnatic music is **mṛdaṅgam**, which becomes the lead instrument during this section. Along with the **mṛdaṅgam** there are other percussion instruments often played in **tani** section such as **kanjira** and **ghatam**. Though **mṛdaṅgam** is a percussion instrument and is basically a kind of a barrel drum, its sound has tonal characteristics. Like the other accompanying instruments in Carnatic music such as **tānpura** and violin, **mṛdaṅgam** is also tuned at the tonic of the lead artist. Due to the tonal acoustical characteristics of **mṛdaṅgam** sound, and because of the long duration of **tani** sections, the pitch estimation algorithm detects and tracks pitch contours during the **tani** sections, instead of detecting these sections as non-voiced segments. Notice that the **mṛdaṅgam** strokes during short (lasting over roughly 1-20 seconds) unvoiced segments such as breath-pauses are correctly detected as non-voiced by the predominant pitch estimation algorithm. This is due to the fact that during such sections the predominant pitch corresponds to the voice, which has significantly higher energy compared to the background percussion signal.

Detecting pitch corresponding to the **mṛdaṅgam** strokes as predominant pitch in the audio poses several challenges in melodic analyses, specifically in the discovery of melodic patterns. Percussion patterns and rhythm cycles contain recurring strokes (or patterns), and therefore, pitch fragments from the **tani** sections are frequently discovered as highly similar patterns. An example of a discovered pair of pitch patterns that correspond to **mṛdaṅgam** strokes is shown in Figure 4.9. We see that the patterns are close to exact repetitions. Since we aim to discover patterns in melodies of **IAM** (Chapter 5), patterns discovered from the **tani** sections are undesired. There can be two approaches to avoid such unwanted patterns in the final output: 1) post-process the discovered pitch patterns and detect if they correspond to **mṛdaṅgam** strokes, 2) discard pitch contours corresponding to the **tani** section in the pre-processing step. We follow the second approach and discard segments of the pitch contours that correspond to the **tani** sections from the input given to the pattern discovery method. Dis-

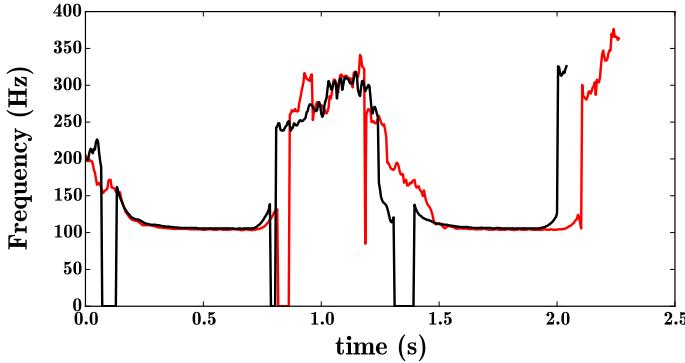


Figure 4.9: Pitch contours of a pair of highly similar patterns corresponding to the *mṛdaṅgam* strokes in a *tāni* section.

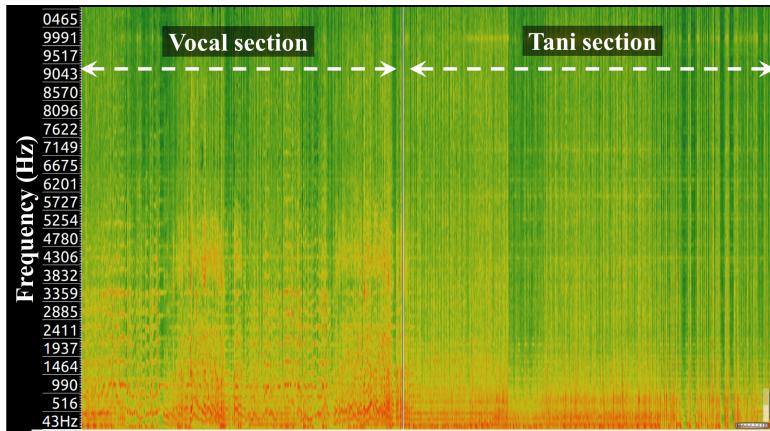


Figure 4.10: The spectrogram of a voice (with percussion) and a *tāni* section in an audio recording of Carnatic music. The excerpt spans 3 minutes of audio.

carding such segments upfront has several advantages. First, detection of the *tāni* sections in audio recording appears to be an easier task compared to characterizing pitch contours as belonging to the melody or *mṛdaṅgam* strokes. This is mainly because of the considerable amount of timbral differences across the sections where melody is present and where it is not. In addition, such computational tasks of classifying an audio segment based on its timbral characteristics is well studied in MIR (Herrera-Boyer et al., 2003). In Figure 4.10, we show the spectrogram of a voice section and a *tāni* section. We see that the timbral characteristics between the two types of sections are considerably different. Another benefit of discarding the *tāni* sections upfront is that it reduces the computational complexity of the pattern discovery task (*tāni* sections may last up to 2-25 minutes).

We detect *tāni* sections in audio recordings using a classification-based approach. To

feed the classifiers we extract 13 MFCC coefficients, spectral centroid, spectral flatness and pitch salience from the audio signal using Essentia (Bogdanov et al., 2013) library. We iterate over 23, 46 and 92 ms frame sizes and chose the one that results in the best classification accuracy. We set the hop size as half the frame size and all other parameters to their default values. Next, we compute the means and the variances of these features over 2 s non-overlapping segments (sometimes referred to as texture window). For training, we use a labeled audio music dataset containing 1.5 hours of mixed voice and violin recordings and 1.5 hours of solo percussion recordings. To assess the performance of the extracted features, we perform a 10-fold cross-validation procedure and repeat the experiment 10 times. We experiment with five different algorithms exploiting diverse classification strategies (Hastie et al., 2009a): decision trees (**Tree**), *k*-NN, naive Bayes (**NB**), logistic regression (**LR**), and **SVM** with a radial basis function kernel. We use the implementations of the classifiers as available in scikit-learn version 0.14.1 (Pedregosa et al., 2011). We used the default set of parameters with few exceptions to avoid over-fitting and to compensate for the uneven number of instances per class. We set `min_samples_split=10` for **Tree**, `fit_prior=False` for **NB**, `n_neighbors=5` for *k*-NN, and for **LR** and **SVM** `class_weight='auto'`.

The combination of the frame size of 46 ms and the **SVM** classifier yielded the best performance (96% accuracy), with no statistically significant difference to the performance with the **Tree** (95.5%) and the *k*-NN (95%), for the same frame size. We finally chose *k*-NN because of its low complexity. Detection accuracy of the tani sections can be improved further by a simple post-processing step. Since **tani** is a single continuous section in an audio recording, class labels of the texture windows predicted by the classifier can be median filtered to remove the spurious labels lasting over a few frames. This ensures continuity of the class labels across texture windows and avoids short non-voiced regions during the vocal sections being classified as tani segments.

4.5 Nyās Svara Segmentation

Musical melodies contain hierarchically organized events that follow a specific grammar (Patel, 2007). Some of these events are musically more salient than others and act as melodic landmarks. Cadential notes in western classical music (Rockstro et al., 2001) or *kārvai* regions in Carnatic music (Sambamoorthy, 1998) are examples of such landmarks. While some of these landmarks can be identified based on a fixed set of rules, others do not follow any explicit set of rules and are learned implicitly by a musician through years of music training. A computational analysis of these landmarks can discover some of these implicitly learned rules and help in developing musically aware tools for music exploration, understanding and education.

Occurrence of a **nyās** in Hindustani music melodies is an example of such a melodic landmark that we investigate in this section. A detailed description of **nyās** is provided

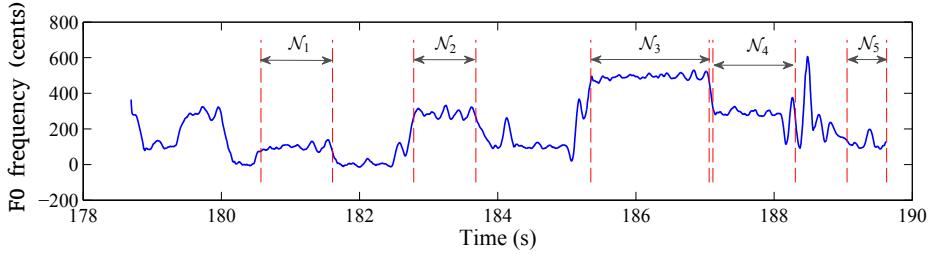


Figure 4.11: A fragment of a pitch contour showing *nyās* segments denoted by \mathcal{N}_i ($i = 1\dots 5$).

in Section 2.3.2. Typically, occurrence of a *nyās* delimits melodic phrases, which constitute one of the most important characteristics of a *rāga*. Analysis of *nyās* is thus a crucial step towards the melodic analysis of Hindustani music. In particular, automatically detecting occurrences of *nyās* (from now on referred as *nyās* segments) will aid in melody segmentation, which is a crucial step in melodic phrase discovery. However, detection of *nyās* segments is a challenging computational task, as the prescriptive definition of *nyās* is very broad, and there are no fixed set of explicit rules to quantify this concept (Dey, 2008, p. 73). It is through rigorous practice that a seasoned artist acquires perfection in the usage of *nyās*, complying with the *rāga* grammar and exploring creativity through improvisation at the same time.

From a computational perspective, the detection of *nyās* segments is challenging due to the variability in segment length, melodic characteristics and the different melodic contexts in which *nyās* is rendered. To illustrate this point we show a fragment of pitch contour in Figure 4.11, annotated with *nyās* segments denoted by \mathcal{N}_i ($i = 1\dots 5$). We see that the *nyās* segment length is highly varied, where \mathcal{N}_5 is the smallest *nyās* segment (even smaller than many non-*nyās* segments) and \mathcal{N}_3 is the longest *nyās* segment. In addition, pitch contour characteristics also vary a lot due to the presence of alankār (Section 2.3.2). The pitch characteristics of a segment depend on the *rāga* and scale degree of the *nyās*, and adds further complexity to the task (Bagchee, 1998). For example, in Figure 4.11, \mathcal{N}_1 and \mathcal{N}_3 have a small pitch deviation from the mean svara frequency, whereas, \mathcal{N}_2 and \mathcal{N}_4 have a significant pitch deviation (close to 100 Cents in \mathcal{N}_5). Large pitch deviations also pose a challenge in segmentation process. Furthermore, melodic context such as the relative position with respect to a non-voiced or a long held svara region plays a crucial role in determining a *nyās* segment. Because of these factors the task of *nyās* segment detection becomes challenging and requires sophisticated learning techniques along with musically meaningful domain specific features.

In the computational analysis of IAM, *nyās* segment detection has not received much attention in the past. To the best of our knowledge, only one study with the final goal of spotting melodic motifs has indirectly dealt with this task (Ross & Rao, 2012). In it, the authors considered performances of a single *rāga* and focused on a very specific *nyās* svara, corresponding to a single scale degree: the fifth with respect to the

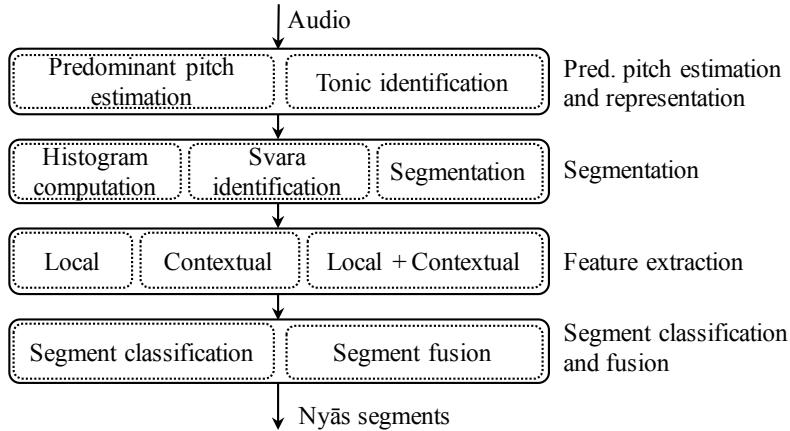


Figure 4.12: Block diagram of the proposed approach for nyās segmentation.

tonic, Pa svara. This svara is considered as one of the most stable svaras, and has minimal pitch deviations. Thus, focusing on it oversimplified the methodology developed in Ross & Rao (2012) for nyās segment detection. Note that the concept of landmark has been used elsewhere, with related but different notions and purposes. That is the case with time series similarity (Perng et al., 2000), speech recognition (Jansen & Niyogi, 2008; Chen et al., 2012), or audio identification (Duong & Thudor, 2013).

In this section, we describe our method for detecting occurrences of nyās svara in Hindustani music melodies. The description is based on our work presented in Gullati et al. (2014b). The method consists of two main steps: segmentation based on domain knowledge, and segment classification-based on a set of musically motivated pitch contour features. There are three main reasons for selecting this approach over a standard pattern detection technique (for example DTW). First, the pitch contour of a nyās segment obeys no explicit patterns, hence, the contour characteristics have to be abstracted. Second, information regarding the melodic context of a segment can be easily interpreted in terms of discrete features. Third, we aim to measure the contribution of a specific feature in the overall classification accuracy (for example, if contour variance and length are the most important features for the classification). This is important in order to corroborate the results obtained from such data driven approaches with that from musicological studies. In the subsequent sections we first describe our method (Section 4.5.1), present the methodology used for evaluation (Section 4.5.2), discuss the results and summarize our findings (Section 4.5.3).

4.5.1 Method

The block diagram of the proposed method for nyās segmentation is shown in Figure 4.12. It consists of four main processing blocks: predominant pitch estimation and representation, segmentation, feature extraction, and segment classification and

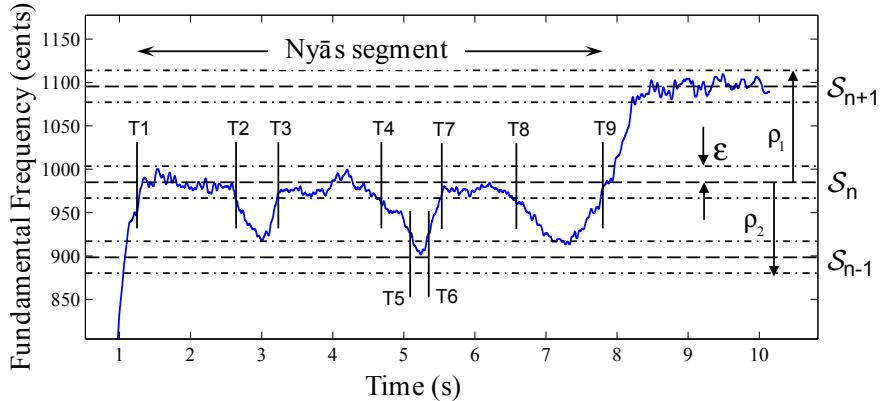


Figure 4.13: Fragment of a pitch contour containing a **nyās** segment ($T_1 - T_9$), where T_i s denote time stamps and S_n s denote mean **svara** frequencies. The pitch deviation within the **nyās** segment ($T_1 - T_9$) is almost 100 Cents.

fusion. These processing blocks are described in the following sections.

4.5.1.1 Predominant Pitch Estimation and Representation

To estimate pitch of the predominant melodic source we use the procedure described in Section 4.3.1 with the same set of parameters. We do not perform any post-processing on the estimated pitch contours. Pitch estimated in Hertz is converted to Cent-scale and is normalized by the tonic of the lead artist in the recording as described in Section 4.3.3.1 and Section 4.3.3.2. Tonic pitch of an audio recording is estimated using **M_{JS}** method (Section 4.2.3).

4.5.1.2 Segmentation

Nyās segment is a rendition of a single **svara** and the aim of the segmentation process is to detect the **svara** boundaries. However, **svaras** contain different **alankārs**, where the pitch deviation with respect to the mean **svara** frequency can go roughly up to 200 Cents (Section 2.3.2). This characteristic of a **svara** in Hindustani music poses a challenge to segmentation. To illustrate this, in Figure 4.13 we present an example of a **nyās** segment (between $T_1 - T_9$, centered around mean **svara** frequency $S_n = 990$ Cents). The pitch deviation in this **nyās** segment with respect to the mean **svara** frequency reaches almost 100 Cents (between $T_5 - T_6$). Note that here the reference frequency, i.e. 0 Cent correspond to the tonic pitch of the lead singer.

We experiment with two different methods for segmenting melodies: piece-wise linear segmentation (**PLS**), a classical, generic approach used for the segmentation of time series data (Keogh et al., 2004), and our proposed method, which incorporates domain knowledge to facilitate the detection of **nyās** boundaries. For **PLS** we use

a bottom-up segmentation strategy as described by Keogh et al. (2004). Bottom-up segmentation methods involve computation of residual error incrementally for each sample of time series. When the residual error satisfies a pre-defined criterion a new segment is created. Out of the two typical criteria used for segmentation, namely average and maximum error, we choose the latter because, ideally, a new segment should be created as soon as the melody progresses from one **svara** to the other. In order to select the optimal value of the allowed maximum error, which we denote by ζ , we iterated over four different values and chose the one which resulted in the best performance. Specifically, for $\zeta = \{10, 25, 50, 75\}$, $\zeta = 75$ Cents yielded the best performance. We rejected $\zeta \geq 100$ Cents in early experimentation stages because few svaras of a *rāga* are separated by an interval of 100 Cents and, therefore, the segmentation output was clearly unsatisfactory.

To make the segmentation process robust to pitch deviations, we propose a method based on empirically-derived thresholds. Unlike **PLS**, our proposed method computes a pitch histogram and uses that to estimate mean **svara** frequencies before the computation of residual error. This allows us to compute the residual error with respect to the mean **svara** frequency instead of computing it with respect to the previous segment boundary, as done in **PLS**. In this way our proposed method utilizes the fact that the time series being segmented is a pitch contour where the values of the time series hover around mean **svara** frequencies. The mean **svara** frequencies for an excerpt are estimated as the peaks of the histogram computed from the estimated pitch values. An octave folded pitch histogram is computed using a 10 Cent resolution and subsequently smoothed using a Gaussian window with a variance of 15 Cents. Only the peaks of the normalized pitch histogram which have at least one peak-to-valley ratio greater than 0.01 are considered as **svara** locations. As peaks and valleys we simply take all local maximas and minimas over the whole histogram. In Figure 4.14 we show an example of an octave folded normalized pitch histogram used for estimating mean **svara** frequencies. The estimated mean **svara** frequencies are indicated by circles. We also notice that the pitch values corresponding to a **svara** span a frequency region and not a single value.

After we estimate mean frequencies of all the **svaras** in a piece, we proceed with their refinement. For the n -th **svara** S_n , we search for contiguous segments within a deviation of ε from S_n , that is, $|S_n - \hat{p}_i| < \varepsilon$, for $i \in [1, N]$, where \hat{p}_i is the fundamental frequency value (in Cents) of the i -th sample of a segment of length N . In Figure 4.13, this corresponds to segments $[T_1, T_2]$, $[T_3, T_4]$, and $[T_7, T_8]$.

Next, we concatenate two segments $[T_a, T_b]$ and $[T_e, T_f]$ if two conditions are met:

1. $\hat{p}_i - S_n < \rho_1$ and $S_n - \hat{p}_i < \rho_2$, for $i \in [T_b, T_e]$, where $\rho_1 = S_{n+1} - S_n + \varepsilon$ and $\rho_2 = S_n - S_{n-1} + \varepsilon$.
2. $T_c - T_d < \psi$, where ψ is a temporal threshold and $[T_c, T_d]$ is a segment between T_b, T_e such that $|S_m - \hat{p}_i| < \varepsilon$ for $i \in [T_c, T_d]$ for $m \in [n-1, n+1]$ and $m \neq n$.

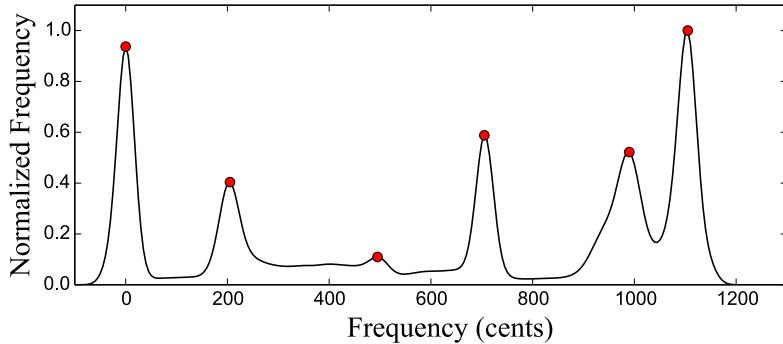


Figure 4.14: Normalized octave folded pitch histogram used for estimating mean svara frequencies. Estimated mean svara frequencies are indicated by circles.

In simple terms, we concatenate two segments if the fundamental frequency values between them do not deviate a lot (less than ρ_1 and ρ_2) and the time duration of the melody in close vicinity (less than ε) of neighboring svaras is not too large (less than ψ). We repeat this process for all svara locations. In our experiments, we use $\varepsilon = 25$ Cents and $\psi = 50$ ms, which were empirically obtained. Notice that we can already derive a simple binary flatness measure v for $[T_a, T_b]$, $v = 1$ if $|\mathcal{S}_n - \hat{p}_i| < \varepsilon$ for $i \in [T_a, T_b]$ for any n and $v = 0$ otherwise.

4.5.1.3 Feature Extraction

We extract musically motivated melodic features for segment classification, which resulted out of discussions with musicians. For every segment obtained following the process mentioned above, three sets of melodic features are computed: local features (\mathcal{F}_L), which capture the pitch contour characteristics of the segment, contextual features (\mathcal{F}_C), which capture the melodic context of the segment, and a third set combining both of them ($\mathcal{F}_L + \mathcal{F}_C$) in order to analyze if they complement each other. Initially, we considered 9 local features and 24 contextual features:

Local Features: segment length, mean and variance of the pitch values in a segment, mean and variance of the differences in adjacent peak locations of the pitch sequence in a segment, mean and variance of the peak amplitudes of the pitch sequence in a segment, temporal centroid of the pitch sequence in a segment normalized by its length, and the above-mentioned flatness measure v (we use the average segmentation error for the case of PLS).

Contextual Features: segment length normalized by the length of the longest segment within the same breath phrase⁴⁸, segment length normalized by the length

⁴⁸Melody segment between consecutive breath pauses of a singer. We consider every unvoiced segment (i.e., a value of 0 in the pitch sequence) greater than 100 ms as breath pause.

of the breath phrase, length normalized with the length of the previous segment, length normalized by the length of the following segment, duration between the ending of the segment and succeeding silence, duration between the starting of the segment and preceding silence, and all the local features of the adjacent segments.

However, after preliminary analysis, we reduced these features to 3 local features and 15 contextual features. As local features we selected length, variance, and flatness measure (v). As contextual features we selected all of them except the local features of the posterior segment. This feature selection was done manually, performing different preliminary experiments with a subset of the data, using different combinations of features and selecting the ones that yielded the best accuracies.

4.5.1.4 Classification and Segment Fusion

Each segment obtained in Section 4.5.1.2 is classified into nyās or non-nyās based on the extracted features described above. To demonstrate that the predictive power of the considered features is generic and independent of a particular classification scheme, we employ five different algorithms exploiting diverse classification strategies (Hastie et al., 2009a): decision tree (Tree), k -NN, NB, LR, and SVM with a radial basis function kernel. We use the implementations available in scikit-learn (Pedregosa et al., 2011), version 0.14.1. We use the default set of parameters with few exceptions in order to avoid over-fitting and to compensate for the uneven number of instances per class. Specifically, we set `min_samples_split=10` for Tree, `fit_prior=False` for NB, `n_neighbors=5` for k -NN, and for LR and SVM `class_weight='auto'`.

For out-of-sample testing we implement a cross-fold validation procedure. We split the dataset into folds that contain an equal number of nyās segments, the minimum number of nyās segments in a musical excerpt. Furthermore, we make sure that no instance from the same artist and rāga is used for training and testing in the same fold.

After classification, boundaries of nyās and non-nyās segments are obtained by merging all the consecutive segments with the same segment label. During this step, the segments corresponding to the silence regions in the melody, which were removed during classification, are regarded as non-nyās segments.

4.5.2 Experimental Setup

4.5.2.1 Music Collection and Annotations

For evaluation of this task we use the nyās dataset NDD_{CM} as described in Section 3.3.2. As explained, this dataset contains both commercially released polyphonic music recordings and in-house monophonic recordings. As melodic characteristics of a nyās segment might depend on the artist and the chosen rāga, the dataset includes

performances by 8 artists in 16 different rāgas to ensure diversity and representativeness.

4.5.2.2 Evaluation Measures and Statistical Significance

We evaluate two tasks, nyās segment boundary annotation, and nyās and non-nyās segment label annotation. For the evaluation of nyās boundary annotations we use hit rates as in a typical music structure boundary detection task (Ong & Herrera, 2005). While calculating hit rate, segment boundaries are considered as correct if they fall within a certain threshold of a boundary in the ground-truth annotation. Using matched hits, we compute standard precision, recall, and F-score for every fold and average them over the whole dataset. The choice of a threshold however depends on the specific application. Due to the lack of scientific studies on the just noticeable differences of nyās svara boundaries, we computed results using an arbitrary selected threshold of 100 ms. Label annotations are evaluated using standard pairwise frame clustering method as described in Levy & Sandler (2008). Frames with same duration as threshold value for the boundary evaluation (i.e. 100 ms) are considered while computing precision, recall, and F-score.

For assessing statistical significance we use the Mann-Whitney U test (Mann & Whitney, 1947) with $p < 0.05$ and assuming an asymptotic normal distribution of the evaluation measures. To compensate for multiple comparisons we apply the Holm-Bonferroni method (Holm, 1979), a powerful method that also controls the so-called family-wise error rate. Thus, we end up using a much more stringent criterion than $p < 0.05$ for measuring statistical significance.

4.5.2.3 Baselines

Apart from reporting the accuracies for the proposed method and its variants, we compare against some baseline approaches. In particular, we consider DTW together with a *k*-NN classifier ($K = 5$). For every segment, we compute its distance from all other segments and assign a label to it based on the labels of its K nearest neighbors, using majority voting. As the proposed method also exploits contextual information, in order to make the comparison more meaningful, we consider the adjacent segments in the distance computation with linearly interpolated values in the region corresponding to the segment. For comparing with the variant of the proposed method that uses a combination of the local and contextual features, we consider adjacent segments together with the actual segment in the distance computation. As this approach does not consider any features, it will help us in estimating the benefits of extracting musically-relevant features from nyās segments.

In addition, to quantify the limitations of the adopted evaluation measures, we compute a few random baselines. The first one (\mathfrak{B}_{R1}) is calculated by randomly planting boundaries (starting at 0 s) according to the distribution of inter boundary intervals obtained using the ground-truth annotations. For each segment we assign the labels

‘nyās’ with a priori probability (same for all excerpts) computed using ground truth annotations of the whole dataset. The second one (\mathcal{B}_{R2}) is calculated by planting boundaries (starting at 0 s) at even intervals of 100 ms and assigning class labels as in \mathcal{B}_{R1} . Finally, the third one (\mathcal{B}_{R3}) considers the exact ground-truth boundaries and assigns the class labels randomly as in \mathcal{B}_{R1} and \mathcal{B}_{R2} . Thus, with \mathcal{B}_{R3} we can directly assess the impact of the considered classification algorithms. We found that \mathcal{B}_{R2} achieves the best accuracy and therefore, for all the following comparisons we only consider \mathcal{B}_{R2} .

4.5.3 Results and Discussion

We evaluate two tasks, nyās segment boundary annotation, and nyās and non-nyās segment label annotation. For both the tasks, we report results obtained using two different segmentation methods (PLS and the proposed segmentation method), five classifiers (Tree, *k*-NN, NB, LR, SVM), and three set of features (local (\mathcal{F}_L), contextual (\mathcal{F}_C) and local together with contextual ($\mathcal{F}_L + \mathcal{F}_C$)). In addition, we report results obtained using a baseline method (DTW) and a random baseline (\mathcal{B}_{R2}).

In Table 4.3, we show the results of nyās boundary annotations. First, we see that every variant performs significantly better than the best random baseline. \mathcal{B}_{R2} yields an F-score of 0.184 while the worst variant tested reaches 0.248. Next, we see that the proposed method achieves a notably higher accuracy compared to the DTW baseline. Such difference is found to be statistically significant, with the only exception of the NB classifier. For a given feature set, the performance differences across classifiers are not statistically significant. The only exceptions are Tree and NB, which yield relatively poor and inconsistent performances over different feature sets. Therefore, we opted to not consider these two classifiers in the following comparisons. Amongst the feature sets, the performance differences are not statistically significant between PLS variants (Table 4.3, top rows), whereas for the case of the proposed segmentation method (Table 4.3, bottom rows), we find that the local features perform significantly better than the contextual features and their combination does not yield consistent improvements. Finally, we see that the best results are obtained using the proposed segmentation method together with the local features, with a statistically significant difference to its competitors. Furthermore, the worst accuracy obtained using the proposed segmentation method is notably higher than the best accuracy using PLS method, again with a statistically significant difference.

In Table 4.4, we show the results for nyās and non-nyās label annotations. We can draw similar conclusions as with Table 4.3: (1) all the method variants perform significantly better than the random baselines, (2) all the proposed method variants yield significant accuracy increments over the DTW baseline, and (3) no statistically significant differences between classifiers (with the aforementioned exceptions). In the label annotations, unlike the boundary annotations, we find that though the local features perform better than the contextual features, the differences are not statistically

Segmentation	Feat.	DTW	Tree	<i>k</i> -NN	NB	LR	SVM
PLS	\mathcal{F}_L	0.356	0.407	0.447	0.248	0.449	0.453
	\mathcal{F}_C	0.284	0.394	0.387	0.383	0.389	0.406
	$\mathcal{F}_L + \mathcal{F}_C$	0.289	0.414	0.426	0.409	0.432	0.437
Proposed	\mathcal{F}_L	0.524	0.672	0.719	0.491	0.736	0.749
	\mathcal{F}_C	0.436	0.629	0.615	0.641	0.621	0.673
	$\mathcal{F}_L + \mathcal{F}_C$	0.446	0.682	0.708	0.591	0.725	0.735

Table 4.3: F-scores for *nyās* boundary detection using PLS method and the proposed segmentation method. Results are shown for different classifiers (Tree, *k*-NN, NB, LR, SVM) and local (\mathcal{F}_L), contextual (\mathcal{F}_C) and local together with contextual ($\mathcal{F}_L + \mathcal{F}_C$) features. DTW is the baseline method used for comparison. F-score for the random baseline obtained using \mathfrak{B}_{R2} is 0.184.

Segmentation	Feat.	DTW	Tree	<i>k</i> -NN	NB	LR	SVM
PLS	\mathcal{F}_L	0.553	0.685	0.723	0.621	0.727	0.722
	\mathcal{F}_C	0.251	0.639	0.631	0.690	0.688	0.674
	$\mathcal{F}_L + \mathcal{F}_C$	0.389	0.694	0.693	0.708	0.722	0.706
Proposed	\mathcal{F}_L	0.546	0.708	0.754	0.714	0.749	0.758
	\mathcal{F}_C	0.281	0.671	0.611	0.697	0.689	0.697
	$\mathcal{F}_L + \mathcal{F}_C$	0.332	0.672	0.710	0.730	0.743	0.731

Table 4.4: F-scores for *nyās* and non-*nyās* label annotation task using PLS method and the proposed segmentation method. Results are shown for different classifiers (Tree, *k*-NN, NB, LR, SVM) and local (\mathcal{F}_L), contextual (\mathcal{F}_C) and local together with contextual ($\mathcal{F}_L + \mathcal{F}_C$) features. DTW is the baseline method used for comparison. The best random baseline F-score is 0.153 obtained using \mathfrak{B}_{R2} .

significant for all the proposed method variants. Furthermore, we also see that the proposed segmentation method consistently performs better than PLS. However, the differences are not always statistically significant.

In addition, we also investigate per-class accuracies for the label annotations. We find that the performance for the *nyās* segments is considerably better than the non-*nyās* segments. This could be attributed to the fact that even though the segment classification accuracy is balanced across classes, the differences in segment length of *nyās* and non-*nyās* segments (*nyās* segments being considerably longer than non-*nyās* segments) can result in more number of matched pairs for *nyās* segments.

In general, we see that the proposed segmentation method improves the performance over PLS method in both the tasks, wherein the differences are statistically significant

in the former case. Furthermore, the local feature set, when combined with the proposed segmentation method, yields the best accuracies. We also find that the contextual features do not complement the local features to further improve the performance. However, interestingly, they perform reasonably good considering that they only use contextual information.

4.5.4 Summary

We described a method for detecting *nyās* segments in melodies of Hindustani music. We divided the task into two broad steps: melody segmentation and segment classification. For melody segmentation we proposed a method which incorporates domain knowledge to facilitate *nyās* boundary annotations. We evaluated three feature sets: local, contextual and the combination of both. We showed that the performance of the proposed method is significantly better compared to a baseline method that uses a DTW-based distance and a *k*-NN classifier. Furthermore, we showed that the proposed segmentation method outperforms a standard approach based on PLS. A feature set that includes only the local features was found to perform best. However, we showed that using just the contextual information we could also achieve a reasonable accuracy. This indicates that *nyās* segments have a defined melodic context which can be learned automatically.

4.6 Summary

In this chapter, we described methods to obtain relevant melodic descriptors and melody representations from audio recordings. We presented algorithms used to extract and post-process the predominant pitch from these recordings. To match the tonal context of melodies across performances, the predominant pitch is normalized by the tonic used in the recordings. We presented an exhaustive evaluation of a number of tonic identification approaches, wherein we analyzed and compared their performance on different music material such as Hindustani and Carnatic music, male and female singers, and for instrumental and vocal music. We showed that our multipitch approach consistently outperformed all the other methods. Subsequently, we explained our methodology for identifying *tani* sections in recordings of Carnatic music. We showed that a classification-based approach that uses timbral features can successfully identify *tani* sections with accuracy. Finally, we described our *nyās*-based approach for segmenting melodies in Hindustani music. We saw that a knowledge driven segmentation approach performs better than a generic method for segmenting time-series data. In addition, we showed that our classification-based approach that uses abstracted melodic features performs better than a string matching based method for labeling *nyās* segments. Furthermore, we demonstrated the utility of the context-based melodic features for *nyās* detection task.

Chapter 5

Melodic Pattern Processing: Similarity, Discovery and Characterization

5.1 Introduction

In this chapter, we present our methodology for discovering musically relevant melodic patterns in sizable audio collections of **IAM**. We address three main computational tasks involved in this process: melodic similarity, pattern discovery and characterization of the discovered melodic patterns. We refer to these different tasks jointly as melodic pattern processing.

“Only by repetition can a series of tones be characterized as something definite. Only repetition can demarcate a series of tones and its purpose. Repetition thus is the basis of music as an art”

(Schenker et al., 1980)

Repeating patterns are at the core of music. Consequently, analysis of patterns is fundamental in music analysis. In **IAM**, recurring melodic patterns are the building blocks of melodic structures. They provide a base for improvisation and composition, and thus, are crucial to the analysis and description of *rāgas*, compositions, and artists in this music tradition. A detailed account of the importance of melodic patterns in **IAM** is provided in Section 2.3.2.

To recapitulate, from the literature review presented in Section 2.4.2 and Section 2.5.2 we see that the approaches for pattern processing in music can be broadly put into two categories (Figure 5.1). One of the types of approaches perform pattern detection (or matching) and follow a supervised methodology. In these approaches the system knows *a priori* the pattern to be extracted. Typically such a system is fed with exemplar patterns or queries and is expected to extract all of their occurrences in a piece of

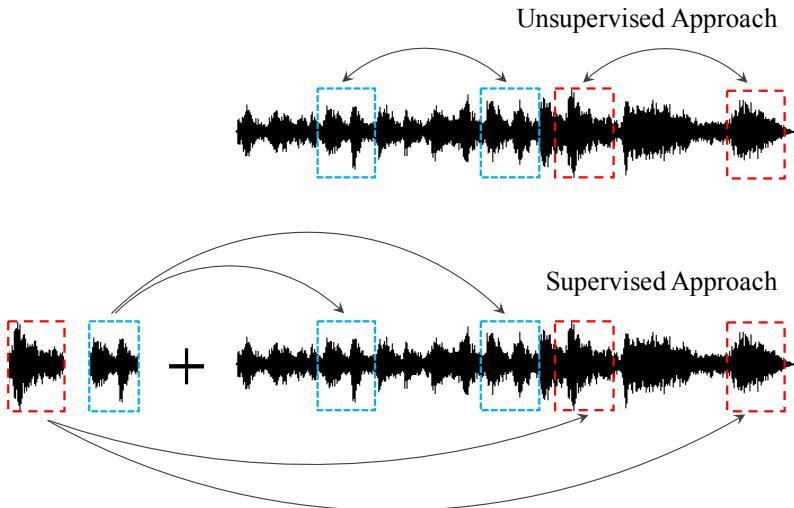


Figure 5.1: Two types of approaches for pattern extraction in music recordings.

music or in a music collection. In the context of melodies in **IAM**, this would mean that musicians provide examples of melodic patterns, which are then used to retrieve all of their occurrences in an audio collection. Nearly all the methods proposed for pattern processing in **IAM** take a supervised approach (Section 2.4.2). Note that this research problem is also addressed to an extent in a related task of query-by-humming (QBH) (Section 2.5.2.3).

We identify several caveats in the supervised methodology described above in the context of pattern processing for melodies in **IAM**. We believe that using only a supervised methodology severely limits the potential of a pattern-based melodic analysis in **IAM**. This is primarily because of the following reasons:

- **Dataset size:** Manually annotating instances of melodic patterns for hundreds of hours of music is a cumbersome process. Since the task of melodic segmentation and similarity is to an extent subjective, ideally speaking, the annotations should be done by multiple domain experts, which makes this task even more challenging. Thus, building a representative and comprehensive corpus of musically meaningful melodic patterns becomes practically unfeasible. This is clearly evident from the size of the datasets used in the existing studies (Section 2.4.2, Table 2.2). The existing datasets typically comprise only a handful of *rāgas*, tens of music pieces, tens of unique number of melodic phrases and a single annotator.
- **Knowledge bias:** The process of annotating melodic patterns in audio recordings of **IAM** essentially boils down to marking regions in time where a known

melodic phrase (*rāga* motif, or *mukhda*) occurs. With long audio recordings (some lasting up to an hour) and improvisatory characteristics of this music, annotating every repeated melodic pattern in the melody (using the concept of parallelism) is close to impossible. This can be attributed to the limited memory of human listeners. Thus, an annotated corpus of melodic patterns suffers from a bias (only the patterns known to an annotator are marked), and does not contain all possible repeating patterns. This is also evident from the datasets used in the existing studies (Section 2.4.2, Table 2.2). The annotated melodic patterns either correspond to *mukhda* phrase or to few well known *rāga* motifs.

- **Human errors:** We found that even when expert listeners are annotating known melodic phrases they are susceptible to making errors in judgment. One of the possible reasons we discovered is the influence of the local melodic context on phrase segmentation. In one of the pattern datasets, **MSD** (Section 3.3.3), we found several instances where many repetitions of the melodic patterns were missed by professional musicians. When these missed instances of the patterns were heard in isolation by the same set of musicians, they were correctly identified. The number of the missed pattern instances was significant, nearly 25% of the total number of annotated patterns. In our conversations with some of these musicians, they commented that the local melodic context masked these patterns and influenced their segmentation. While this phenomenon is yet to be scientifically studied for **IAM**, we at least know that melodic pattern annotations done by even professional musicians are prone to such errors.

One way to circumvent the issues enumerated above is to follow an unsupervised methodology for pattern processing (Figure 5.1). In such an approach a system discovers patterns in the data without any training examples from an expert or a query pattern. In the context of our work, such a system will not require any examples of the annotated melodic patterns from musicians, and thus can be robust to the issues enumerated above. Also, from our literature review presented in Section 2.4.2, we notice that there are only a few studies that address the task of discovering short-time melodic patterns in audio music collections. Based on these considerations, in this thesis, we follow an unsupervised approach for discovering melodic patterns in sizable audio collections of **IAM**.

The arguments presented above brings us to an interesting question, what do we mean by a melodic pattern given we do not have any example of it provided by an expert? Recall (Section 2.2), in the scope of this thesis any recurring melodic fragment is considered as a melodic pattern. We believe that since it repeats, it is not just a coincidence, but it has some significance as the artist rendered the same melodic fragment multiple times. It should be noted that by a recurrence we do not mean a numerical repetition of the pitch sequence. By a recurrence in this context we refer to a musical recurrence of a melodic pattern, which can undergo several melodic variations allowed in the music tradition without loosing its musical identity. In addition, we

carefully differentiate the term melodic pattern from melodic motif, wherein the latter is used to refer to a musically significant pattern that is characteristic of a *rāga* or a composition (Section 2.2).

Along with the aforementioned advantages, there are several challenges in following an unsupervised approach for pattern extraction. One of the most challenging aspects of such approaches is its quantitative evaluation. Due to the absence of ground-truth annotations evaluating such approaches becomes difficult. This becomes even more challenging for a data-driven research methodology since the system parameters are often optimized based on the data. After our first study presented in Gulati et al. (2014c) on mining melodic patterns, we realized that selecting optimal values of the system parameters is notoriously difficult in a completely unsupervised setup. As a result, we bootstrap our methodology by first improving one of the core processing blocks in this task, the computation of melodic similarity, within a supervised setup.

This chapter is divided into four sections, each of which focuses on an important aspect related with pattern processing in IAM. The contents of these sections are:

- In Section 5.2, we address the task of computing melodic similarity. The main objective is to perform an exhaustive evaluation of different methodologies and parameter settings in order to study their influence on the computation of melodic similarity in IAM. This section is based on our work presented in Gulati et al. (2015b).
- In Section 5.3, we focus on improving melodic similarity in IAM by exploiting the culture specific characteristics of the melodies in Carnatic and Hindustani music. This section is based on the study presented in Gulati et al. (2015c).
- In Section 5.4, we describe our methodology for discovering melodic patterns in audio music collections of IAM. This section is largely based on our work presented in Gulati et al. (2014c).
- In Section 5.5, we describe our approach to characterize the discovered melodic patterns in order to identify the ones that correspond to the *rāga* motifs. This section is based on our published work in Gulati et al. (2016c).

5.2 Melodic Similarity: Approaches and Evaluation

In this thesis, we regard repeating melodic fragments as melodic patterns. As explained above, the concept of repetition or recurrence in this context is not just a numerical reiteration of exactly the same pitch sequence. But, it is musical in nature, allowing for permitted melodic variations. Thus, the idea of repetition is closely linked to that of perceptual melodic similarity, which is influenced by several factors including the specificities of a music culture. Our objective in this study is to model this

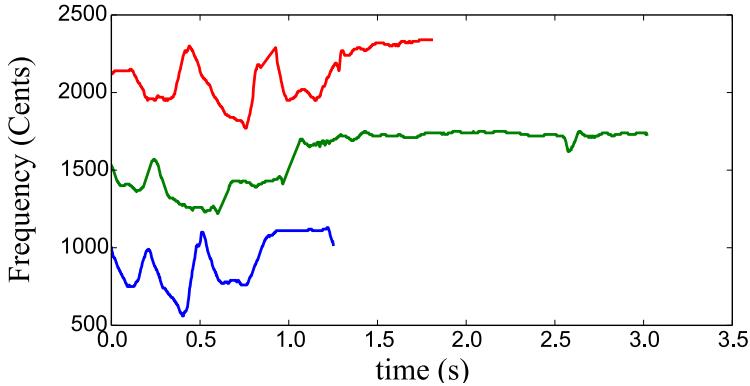


Figure 5.2: Melodic fragments corresponding to three different renditions of the same characteristic *rāga* phrase in Hindustani music. For a better visualization, the patterns are transposed by a frequency offset of 600 Cents between them.

perceptual similarity between short-duration melodic fragments, which will eventually be used for extracting melodic patterns in audio collections of **IAM**.

Studying computational models for melodic similarity is particularly interesting in the context of **IAM**. It is mainly because this music is inherently improvisatory in nature, and the melodies are largely based around melodic patterns. Due to melodic improvisation, which is largely governed by *rāga* grammar, the surface representation of melodic patterns varies significantly across occurrences. This is specifically true for the characteristic melodic patterns of *rāgas* (Section 2.3.2). These patterns constitute the artists' ground for expressing creativity through improvisation. Hence, even when two melodic patterns are perceptually the same for a musician, their surface melodic representation can be drastically different. To illustrate this, we present an example in Figure 5.2, where three melodic fragments corresponding to the same characteristic *rāga* phrase in Hindustani music are shown. We notice that, despite these patterns being the occurrences of the same melodic phrase, they differ considerably in terms of their surface melody representation. In addition to improvisation, peculiar characteristics of melodies in **IAM** such as the usage of *gamakas* in Carnatic music, the long held steady *svaras* and melodic ornaments in Hindustani music further adds to the complexity of the task.

Melodic similarity computation is a critical processing block within pattern processing in melodic sequences. In Section 2.4.2 and Section 2.5.2.3, we review the existing approaches for melodic pattern processing, which directly or indirectly address the task of melodic similarity. From our review we see that the computational methods to assess melodic similarity have received considerable attention from the **MIR** community for a long time. A large number of studies focusing on melodic similarity work with a symbolic representation of music (Section 2.5.2.1). When working with audio music signals, this topic has been studied in depth within the tasks of **query-by-**

humming (QBH) and pattern detection in audio recordings (see Section 2.5.2.3 and Section 2.4.2).

The approaches proposed for computing melodic similarity primarily differ in the choices made for the main processing steps comprising: melody representation, melody segmentation, pitch transposition invariance, tempo or timing invariance, and distance measure (Section 2.5.2.3). In addition, every method has a set of additional choices for selecting the optimal parameters at each processing step. Since the melodic characteristics across music traditions vary considerably, these procedures and parameter choices cannot be generalized to all music traditions. Therefore, studies that perform a comparative evaluation of different methods and analyse the effect of different parameter settings for a specific type of music material are valuable to the community (Dannenberg et al., 2007; Rao et al., 2014; Serra, 2011).

During the course of this thesis work several approaches have been proposed for computing melodic similarity in **IAM** (Section 2.4.2). However, a consensus on the best approach has yet not been reached. This is mainly because these approaches are evaluated using different datasets and under different experimental setup. To the best of our knowledge there has not been any study that systematically evaluates the influence of different choices of procedures and parameter values involved in similarity computation in melodies of **IAM**.

In this section, we describe a string matching-based approach for computing similarity between two short-time melodic patterns in **IAM**. Our objective is to perform a comprehensive evaluation of different variants of this approach to study the influence of different system parameters and procedures on melodic similarity for **IAM**. We evaluate 560 variants put together by combining different choices of the sampling rate of the melody representation, pitch quantization levels, melody normalization techniques, uniform time-scaling and distance measures. We believe the findings of this study will pave the way for developing unsupervised melodic pattern discovery approaches as described in the subsequent sections, whose evaluation is a challenging and, many times, ill-defined task. The current section is based on our published work presented in Gulati et al. (2015b).

5.2.1 Method

The block diagram for computing melodic similarity is shown in Figure 5.3. There are four main processing blocks involved in this task: predominant pitch estimation, pitch normalization, uniform time-scaling, and distance computation. We explore different combinations of the choices made for the processing steps and the parameter settings.

5.2.1.1 Predominant Pitch Estimation

We follow common practice and represent melody by the predominant pitch of an audio signal. For Carnatic music, we use a state-of-the-art melody extraction method

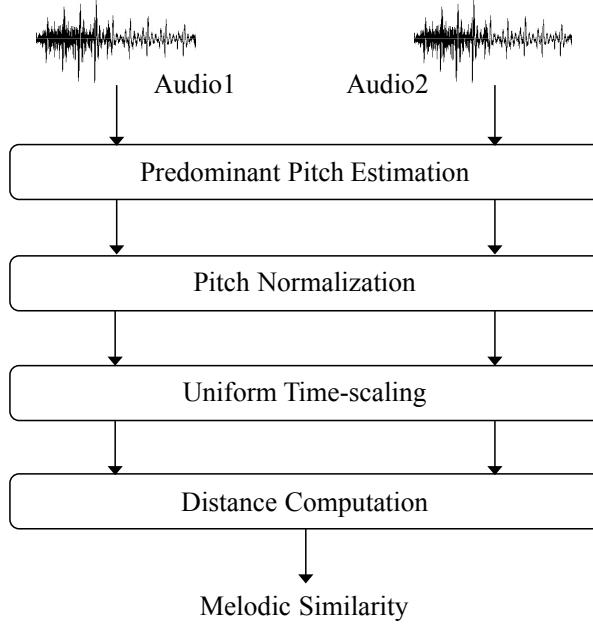


Figure 5.3: Block diagram of the methodology followed for computing melodic similarity from audio recordings.

proposed by Salamon & Gómez (2012) as described in Section 4.3.1. We do not perform any post-processing on the pitch contours except for interpolating short unvoiced segments (Section 4.3.2.3). For Hindustani music we use semi-automatically extracted predominant pitch contours included in the dataset that we use to perform evaluations (Section 3.3.3). This dataset along with these pitch contours has been used in several studies on similar topics (Rao et al., 2014; Ross et al., 2012; Ross & Rao, 2012). We convert the pitch values from Hertz to Cents in order to make the representation musically more relevant (Section 4.3.3.1).

Since automatic assessment of melodic similarity is a computationally expensive task, particularly when done on large audio archives, we desire the minimum possible sampling rate of the melody without compromising the accuracy. We therefore analyse the effect of different sampling rates of the melody representation on melodic similarity. We consider 5 sampling rates 100, 67, 50, 40 and 33 Hz of predominant pitch, implemented by down-sampling the original melody sequence as described in Section 4.3.2.4. We denote these parameter settings by ω_{100} , ω_{67} , ω_{50} , ω_{40} and ω_{33} . Note that in our literature review we found that none of the existing approaches performed a systematic evaluation of this important parameter for melodies in IAM (Section 2.4.2).

5.2.1.2 Pitch Normalization

The absolute values of the pitch samples (in both Hertz and Cent scale) corresponding to different occurrences of a melodic pattern may differ across artists and even within the same recording. There are two main reasons for this. First, in IAM the reference frequency for a melody rendition is the tonic of the lead artist (Section 2.3.2), which typically varies across artists. And second, a melodic pattern may recur in a different octave within the same recording. Therefore, a meaningful comparison of the melodic patterns across different artists and across different sections of a recording is possible only when the similarity computation is invariant to certain pitch transpositions. Out of these two cases, the latter is relatively infrequent, and is ignored in most of the existing studies (Section 2.4.2).

We experiment with five different normalization techniques to achieve pitch transposition invariance. They are as follows.

1. Normalizing the pitch values of a melodic pattern by the tonic of the lead artist of the recording (Z_{tonic}). This is implemented by considering the tonic frequency of the lead artist as the base frequency in Hertz to Cents conversion (see Section 4.3.3.2)
2. Zero mean normalization (Z_{mean}), where mean of the melodic pattern is subtracted from each pitch sample so that the resulting mean of the pattern becomes zero
3. Zero median normalization (Z_{median}), where median of the melodic pattern is subtracted from each pitch sample so that the resulting median of the pattern becomes zero
4. Z-normalization (Z_{Znorm}), where from each pitch sample we subtract the mean of the melodic pattern and subsequently divide it by the standard deviation of the pattern
5. Median absolute deviation normalization (Z_{MAD}), where from each pitch sample we subtract the median of the melodic pattern and subsequently divide it by the median absolute deviation of the pattern

In the case of Z_{tonic} the tonic pitch of the lead artist is identified using **MJS** method, the best performing method resulted from our exhaustive evaluations (Section 4.2.3). Furthermore, since the reference frequency of the melody is known for this case, the pitch values can be quantized, as reported in other studies (Ross et al., 2012). Hence, we additionally experiment with two quantization levels: semitone level, quantizing pitch values to 100 Cents interval (Z_{tonicQ12}) and quarter-tone level, quantizing pitch values to 50 Cents interval (Z_{tonicQ24}). Note that the tonic normalization (Z_{tonic}) is

helpful only in the scenarios where the frequency transpositions are due to the different tonic frequencies of the lead artists across recordings. It does not handle the cases where a pattern recurs in a different octave or a tetra-chord within the same recording. In total, we consider 8 different normalization variants, including the one without any normalization (Z_{off}).

5.2.1.3 Uniform Time-scaling

In order to compensate for global tempo variations across occurrences of a melodic pattern, a typical approach is to consider multiple uniformly time-scaled versions of the patterns (Mazzoni & Dannenberg, 2001; Zhu & Shasha, 2003a; Kotsifakos et al., 2012). Such tempo differences if not handled can significantly degrade the performance in retrieval scenarios where fixed duration patterns are considered. We experiment with two possibilities: first, we do not apply any time-scaling to the patterns (Ω_{off}) and second, we generate 5 copies of every pattern before similarity computation by uniformly time-scaling it by a factor of 0.9, 0.95, 1, 1.05 and 1.1 (Ω_{on}). We implement uniform time-scaling using cubic interpolation.

5.2.1.4 Distance Computation

To measure the melodic similarity between two patterns we consider two categories of commonly used distance measures: Euclidean distance (D_{Euc}) and dynamic time warping (DTW)-based distance (see Section 2.4.2 and Section 2.5.2.3). Euclidean distance is a non-parametric distance measure (Section 2.6.1.1, Eq. 2.1), whereas, DTW-based distance has many variants and parameters to select (Section 2.6.2).

In this study we consider the whole sequence matching DTW variant and two possibilities of the local constraint (Section 2.6.2):

- Where the DTW step condition is $\{(1,0),(1,1),(0,1)\}$, which is without any local constraint. We denote this variant by $D_{\text{DTW_L0}}$. In this variant the DTW accumulated cost matrix is computed using Eq. 2.9.
- Where the DTW step condition is $\{(2,1),(1,1),(1,2)\}$, which is with a local constraint. We denote this variant by $D_{\text{DTW_L1}}$. In this variant the DTW accumulated cost matrix is computed using Eq. 2.11.

In addition, for both these DTW variants we also apply Sakoe-Chiba global band constraint (Sakoe & Chiba, 1978) with the width of the band as 5%, 10% and 90% of the pattern length. Note that 90% band width in the global constraint is to simulate the case of unconstrained DTW. We denote these combinations by $D_{\text{DTW_L0_G5}}$, $D_{\text{DTW_L0_G10}}$, $D_{\text{DTW_L0_G90}}$, $D_{\text{DTW_L1_G5}}$, $D_{\text{DTW_L1_G10}}$, and $D_{\text{DTW_L1_G90}}$, respectively. In total, we consider 7 variants of distance measures for melodic similarity computation.

Since the length of the melodic patterns are different, before computing similarity between two patterns we apply a uniform time-scaling to make their lengths equal. We select the maximum of the lengths of the two patterns as the final length. This operation is a must for the Euclidean distance and has been shown to have a slightly beneficial effect for DTW (Ratanamahatana & Keogh, 2004; Zhu & Shasha, 2003a).

5.2.2 Evaluation Methodology

We use **MSD** dataset for evaluating different variants of the method for computing melodic similarity (Section 3.3.3). Since melodic characteristics across Carnatic and Hindustani music differ considerably, **MSD** comprises two sub-datasets, $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{itb}}^{\text{hmd}}$. For evaluation of melodic similarity in Carnatic music we use $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ dataset, and in Hindustani music we use $\text{MSD}_{\text{itb}}^{\text{hmd}}$ dataset. Both these datasets contain annotations comprising occurrences of five different characteristic phrases of *rāgas* (Table 3.7). Note that, we regard each characteristic melodic phrase as a type of pattern.

We consider every annotated pattern as a query and perform an exhaustive search in the target search space comprising all the annotated patterns in the entire music collection. To make the experimental setup closer to the real world scenario, we add melodic segments other than the annotated patterns in the target search space, which act as noise (referred to as noise candidates). We generate these candidates by randomly selecting short fragments of the melodies from the dataset. The time stamps of the starting of these noise candidates are generated using a uniform distribution, and the lengths are determined using the distribution of the duration values of the annotated patterns. The total number of noise candidates added is 100 times the number of annotated patterns for each dataset. For every query, we order the search results by the similarity values and consider the top 1000 nearest neighbors for evaluation. A retrieved pattern is considered as a true hit only if it belongs to the same pattern type as the query pattern.

In the experimental setup described above, the segmentation of the melodic patterns is done using the ground-truth annotations. However, in several tasks such as in melodic pattern discovery the segmentation information might not be present. Thus, to simulate a retrieval scenario where the pattern boundaries are not known *a priori*, we also consider a simple extension to the experiment by assuming the target pattern length to be equal to the length of the query pattern. We perform all our experiments for both these setups.

We evaluate all possible combinations of the choices made at each step of the melodic similarity computation discussed in Section 5.2.1. We consider 5 different sampling rates of the melody representation, 8 different normalization scenarios, 2 possibilities of uniform time-scaling and 7 variants of the distance measures. In total, we evaluate 560 different variants.

To quantify the performance of a melodic similarity variant considered in this study

Dataset	MAP	Srate	Norm	TScale	Dist
$\text{MSD}_{\text{iitm}}^{\text{cmd}}$	0.413	ω_{67}	Z_{mean}	Ω_{off}	$D_{\text{DTW_L1_G90}}$
	0.412	ω_{67}	Z_{mean}	Ω_{on}	$D_{\text{DTW_L1_G10}}$
	0.411	ω_{100}	Z_{mean}	Ω_{off}	$D_{\text{DTW_L1_G90}}$
$\text{MSD}_{\text{iitb}}^{\text{hmd}}$	0.552	ω_{100}	Z_{tonic}	Ω_{off}	$D_{\text{DTW_L0_G90}}$
	0.551	ω_{67}	Z_{tonic}	Ω_{off}	$D_{\text{DTW_L0_G90}}$
	0.547	ω_{50}	Z_{tonic}	Ω_{off}	$D_{\text{DTW_L0_G90}}$

Table 5.1: MAP score and details of the parameter settings for the three best performing variants for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ dataset. Srate: sampling rate of the melody representation, Norm: normalization technique, TScale: uniform time-scaling and Dist: distance measure.

we use mean average precision (MAP), a typical evaluation measure in information retrieval (Manning et al., 2008). Mean average precision (MAP) is computed by taking the mean of the average precision values of each query in the dataset. This way, we have a single number to evaluate and compare the performance of a variant. In order to assess if the difference in the performance of any two variants is statistically significant, we use the Wilcoxon signed rank-test (Wilcoxon, 1945) with $p < 0.01$. To compensate for multiple comparisons, we apply the Holm-Bonferroni method (Holm, 1979). Thus, considering that we compare 560 different variants, we effectively use a much more stringent criterion than $p < 0.01$ for measuring statistical significance.

5.2.3 Results and Discussion

In this section, we present the results of our evaluation of the 560 variants for each of the datasets, $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$. We order the variants in the decreasing order of their MAP scores and present only the three best performing variants in Table 5.1. For complete results see the companion page for this study (Appendix A).

In Table 5.1 (top half), we show the MAP scores and details of the parameter settings for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ dataset. We see that the best performing variant has a MAP score of 0.413. Having a look at the whole list, we observe that for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$, in the ranked list of the 560 variants, top performing variants consistently use higher sampling rates (either ω_{100} , or ω_{67}). This can be attributed to the rapid oscillatory melodic movements present in Carnatic music, whose preservation requires a higher sampling rate. The top variants invariably use the zero mean normalization (Z_{mean}), suggesting that there are several repeated instances of the melodic patterns that are pitch transposed within the same recording. In addition, top variants also use DTW-based distance with local constraint (either $D_{\text{DTW_L1_G10}}$ or $D_{\text{DTW_L1_G90}}$), indicating that melodic fragments are prone to large pathological warping that can significantly degrade the performance. We do not observe any consistency in the usage of uniform time-scaling.

However, it strongly correlates with the global constraint in the **DTW** distance. In majority of the top ranked variants, Ω_{on} consistently occurs with $D_{\text{DTW_L1_G10}}$, and Ω_{off} consistently occurs with $D_{\text{DTW_L1_G90}}$. This suggests that a combination of uniform time-scaling and narrow global band constrained variant of **DTW** is able to handle the same degree of non-linear timing variations as can be handled by a globally unconstrained **DTW**. However, the former configuration is computationally more efficient than the latter. We also performed an analysis of several (small distance) false positives and found that the **MAP** scores for a number of queries were low because of the spurious errors in the predominant pitch.

The **MAP** scores and details of the parameter settings for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ dataset are shown in Table 5.1 (bottom half). Compared to $\text{MSD}_{\text{iitm}}^{\text{cmd}}$, the best **MAP** score for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ is higher (0.55). Amongst the top ranked variants there is no consensus on the sampling rate of the melody representation. All the top ranked variants have the same parameter values except the sampling rate. This suggests that the sampling rates considered in this study have no significant effect on the melodic similarity for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$. This can be attributed to the fact that the recordings in $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ are slow-medium tempo music pieces that do not have fast oscillatory melodic movements, as was the case with $\text{MSD}_{\text{iitm}}^{\text{cmd}}$. Furthermore, for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$, we observe that the variants using Z_{tonic} , Z_{tonicQ12} or Z_{tonicQ24} perform better than the ones using Z_{mean} , which is in contrast to the observation for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$. This is primarily because in Carnatic music in our datasets there are many cases where a pattern recurs in a different octave within the same recording, whereas, in Hindustani music, such cases are rare. In general, we see that the **DTW**-based distance performs better than the euclidean distance, and the **DTW** variant without a global constraint ($D_{\text{DTW_L1_G90}}$ or $D_{\text{DTW_L0_G90}}$) is preferred. This implies that the repeated instances of melodic patterns in **IAM** (specifically in Hindustani music) have large non linear timing variations.

To assess the statistical significance of the results we compare every possible pair of the variants (${}^{560}C_2 = 156520$ comparisons). The results are shown in Figure 5.4, where both the axes are the index of the variants in the ranked list. For every variant pair with index i and j , we mark the pixel (i, j) if the difference is statistically significant. From Figure 5.4 we see that a majority of variant pairs have a statistically significant difference in the **MAP** scores. This indicates that the task of computing melodic similarity is sensitive to the choice of parameters and processing steps, and a small change in the choices made in a variant can lead to a significantly different **MAP** score. Furthermore, as the marked pixels are higher in number for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$, this sensitivity is even higher for $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ compared to $\text{MSD}_{\text{iitm}}^{\text{cmd}}$.

To analyse the consistency in the performance across pattern types, we present the boxplot of the average precision values for each pattern type in Figure 5.5. For this, we consider only the top performing variant for each dataset. We see that the **MAP** scores vary considerably across pattern types for both $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$. Furthermore, we observe that the intra pattern type variance of the average precision values is higher for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ as compared to $\text{MSD}_{\text{iitb}}^{\text{hmd}}$. We observe that the pattern types H_2 and H_5

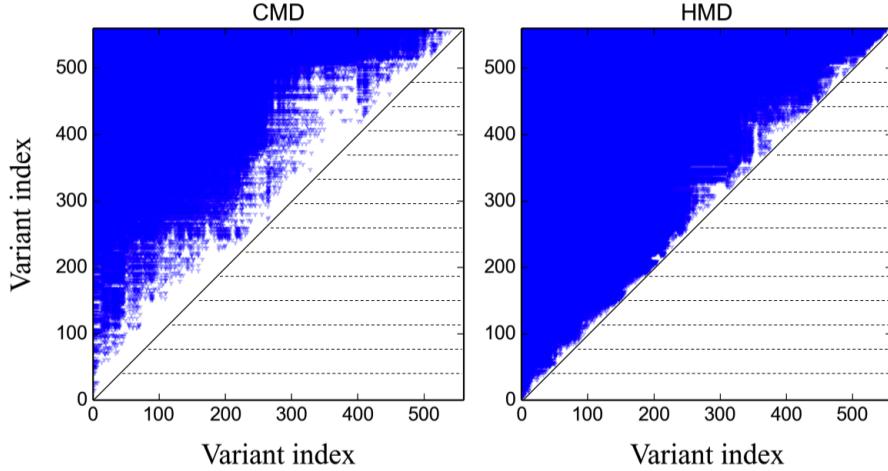


Figure 5.4: Matrix indicating the statistical significance of the performance difference between every variant pair for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitm}}^{\text{hmd}}$. Variant pairs, where the difference in the performance is statistically significant, are marked by blue dots. Only the superscript in the name of the dataset is used in the figure title.

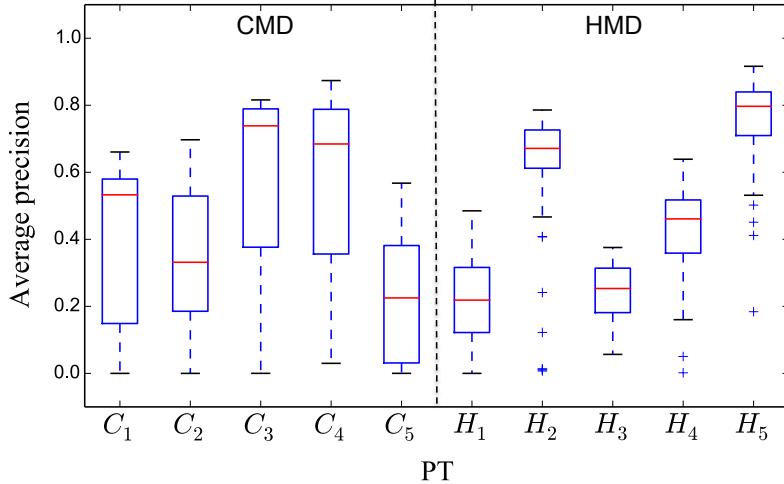


Figure 5.5: Boxplot of the average precision values for each pattern type (PT) in $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitm}}^{\text{hmd}}$. Only the superscript in the name of the dataset is used for labeling.

have a higher MAP score compared to other pattern types in $\text{MSD}_{\text{iitm}}^{\text{hmd}}$. Interestingly, H_2 and H_5 are also the pattern types for which the variance in the length is lower and the number of occurrences is higher than the other pattern types in $\text{MSD}_{\text{iitm}}^{\text{hmd}}$ dataset (Table 3.7). This correlation is not evident in $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ dataset.

So far we have seen the results in the experimental setup where the melodic patterns

Dataset	MAP	Srate	Norm	TScale	Dist
$\text{MSD}_{\text{iitm}}^{\text{cmd}}$	0.279	ω_{67}	Z_{mean}	Ω_{on}	$D_{\text{DTW_L1_G10}}$
	0.277	ω_{67}	Z_{tonicQ12}	Ω_{on}	$D_{\text{DTW_L1_G10}}$
	0.275	ω_{100}	Z_{tonicQ12}	Ω_{on}	$D_{\text{DTW_L1_G10}}$
$\text{MSD}_{\text{iitb}}^{\text{hmd}}$	0.259	ω_{40}	Z_{tonicQ12}	Ω_{on}	$D_{\text{DTW_L1_G90}}$
	0.259	ω_{100}	Z_{tonicQ12}	Ω_{on}	$D_{\text{DTW_L1_G90}}$
	0.259	ω_{67}	Z_{tonicQ12}	Ω_{on}	$D_{\text{DTW_L1_G90}}$

Table 5.2: MAP score and details of the parameter settings for the three best performing variants for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ dataset. These results are corresponding to the experimental setup where the target patterns' length is not based on the ground-truth annotations, but is taken to be the same as the query pattern length. Srate: sampling rate of the melody representation, Norm: normalization technique, TScale: uniform time-scaling and Dist: distance measure.

are segmented using the ground-truth annotations. We now present the results corresponding to the other experimental setup described in Section 5.2.2, where the target pattern lengths are considered to be the same as that of the query pattern. We evaluate all 560 variants of the method as done above on both the datasets under this experimental setup. In Table 5.2 we show the MAP scores of the top performing variants for both the datasets $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$. We find that the MAP score for the best performing variant decreases from 0.41 to 0.28 and 0.55 to 0.26 for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$, respectively. This indicates that the melodic similarity computation task becomes much more challenging in the absence of an accurate melodic segmentation method. For this experimental setup the trend in the sampling rate for Carnatic and Hindustani music remain the same as we saw in Table 5.1, which is that a higher sampling rate is desired for representing melodic patterns in Carnatic music. In terms of the normalization we see that surprisingly Z_{tonicQ12} is used by the top performing variants for both the datasets. Note that in other variants whose performance is not statistically significantly different from the ones reported in Table 5.2, Z_{mean} and Z_{tonic} normalization is also used for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ dataset respectively. An interesting observation is that in this experimental setup uniform time-scaling is consistently used by all the top performing variants. This indicates that such a time-scaling operation is immensely advantageous in the retrieval scenarios where the length of the target melodic patterns is taken to be the same as that of a query pattern (i.e. pattern segmentation is not known). We also see that $D_{\text{DTW_L1_G10}}$ and $D_{\text{DTW_L1_G90}}$ are always used for $\text{MSD}_{\text{iitm}}^{\text{cmd}}$ and $\text{MSD}_{\text{iitb}}^{\text{hmd}}$ datasets, respectively, indicating that applying a local constraint in DTW is critical for this experimental setup.

5.3 Improving Melodic Similarity

In the previous section we performed an exhaustive comparison of different procedures and parameter settings typically involved in the computation of melodic similarity (Section 5.2). We learned about the impact of the different methodologies on the accuracy of the system, and about the best set of parameter settings for computing melodic similarity in **IAM**. Results indicate that this task is challenging, and even in the case of the best methodology, there exists a large scope for improvement. In this section we build upon our findings in Section 5.2 and investigate the exploitation of specific melodic characteristics of Hindustani and Carnatic music to further improve melodic similarity.

From our literature review presented in Section 2.4 and Section 2.5 we see that the methodologies for computing melodic similarity varies depending on the type of music material (sheet music or audio, monophonic or polyphonic) (Marsden, 2012b; Meredith et al., 2002; Cambouropoulos, 2001b; Collins et al., 2014; Ghias et al., 1995; Dannenberg et al., 2007; Mazzoni & Dannenberg, 2001) and the music tradition (Juhász, 2009; Conklin & Anagnostopoulou, 2011; Lartillot & Ayari, 2006; Pikrakis et al., 2003). Existing literature also indicates that the important characteristics of several melody-dominant music traditions of the world such as Flamenco and **IAM** need dedicated research efforts to devise approaches for computing melodic similarity (Gómez et al., 2012; Pikrakis et al., 2012, 2016; Rao et al., 2014). With this spirit of devising culture specific approaches several methods for retrieving different types of melodic patterns have been proposed for **IAM** during the course of this dissertation (Ross et al., 2012; Ross & Rao, 2012; Ishwar et al., 2012; Rao et al., 2014; Ishwar et al., 2013; Dutta & Murthy, 2014a,b).

We recapitulate briefly the approaches reviewed in Section 2.4 that exploit specificities in **IAM**. Ishwar et al. (2013) propose a saddle point based representation of melody that exploits the presence of **gamakas** in Carnatic music. This representation is used in the first stage of a two-stage process to prune the target search space. Dutta & Murthy (2014b) propose to modify the intermediate steps involved in the computation of the **RLCS** distance to make it more suitable to the melodic sequences in Carnatic music. Ross et al. (2012) utilize the **sama** locations to reduce the search space for detecting the **mukhda** patterns of a composition in Hindustani music. Ross & Rao (2012) pruned the search space by employing a melodic landmark called **nyās svara**. Rao et al. (2014) address the challenge of a large within-class variability in the occurrences of the characteristic phrases of **rāgas**. They propose to use exemplar-based matching after vector quantization-based training to obtain multiple templates for a given phrase category. In addition, the authors also propose to learn an optimal **DTW** constraint for each phrase category in order to exploit the possible patterns in the duration variability of melodic phrases in Hindustani music.

The approaches mentioned above are directed towards developing culturally informed and knowledge-driven computational methodologies. However, as we notice, there is

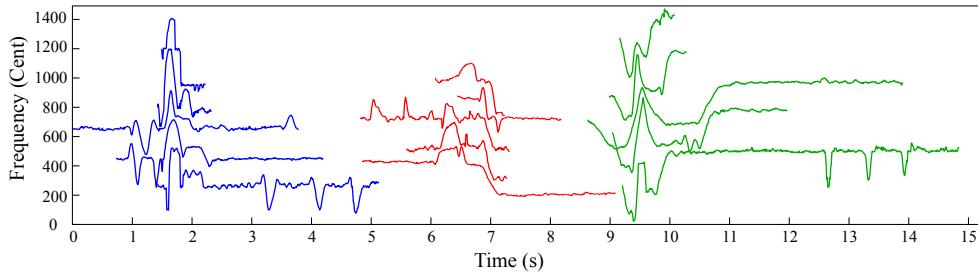


Figure 5.6: Pitch contours of occurrences of three different characteristic melodic phrases in Hindustani music. Contours are frequency transposed and time shifted for a better visualization.

a large scope for further improvement in this area. One of the main shortcomings of nearly all these existing approaches is their scalability to different musical forms and styles within IAM. Most of these approaches are proposed and evaluated on either Hindustani or Carnatic music. Even within these two music traditions they focus on a certain type of melodic patterns, musical style, and in some cases, to only slow tempo (*vilambit lay*) music compositions. For example, *sama* location can indicate roughly the onset of a *mukhda* phrase in an recording of Hindustani music (Ross et al., 2012). But, it has no musically established relationship with the location of the characteristic melodic phrases of *rāgas*. Similarly, the *Pa nyās* segmentation strategy followed in Ross & Rao (2012) can work only with the melodic phrases ending in the *Pa svara*, and mainly for slow tempo compositions where the concept of *nyās svara* is evident. Thus, these approaches do not generalize and scale to other types of melodic patterns and to large music collections of IAM. Moreover, detecting these landmarks is a challenging task in itself (Srinivasamurthy & Serra, 2014; Gulati et al., 2014b). Some of the above mentioned approaches also propose solutions to handle large within-class variability in melodic patterns, but they are suitable for a supervised analysis of melodic patterns and are clearly not applicable to unsupervised analysis. Our objective here is to devise an approach that can utilize specific melodic characteristics in IAM, and at the same time generalize to different musical forms and styles within this music tradition.

Before proceeding further it is worth revising the main challenges involved in the computation of melodic similarity for characteristic melodic phrases of *rāgas*. As already mentioned, the characteristic melodic phrases act as the basis for the artists to improvise, providing them with a medium to express creativity during a *rāga* rendition. Hence, the surface representation of these melodic phrases can vary a lot across their occurrences. This high degree of variability in terms of the duration of a phrase, non-linear time warping and the added melodic ornaments together pose a big challenge for melodic similarity computation. In Figure 5.6 we illustrate this variabil-

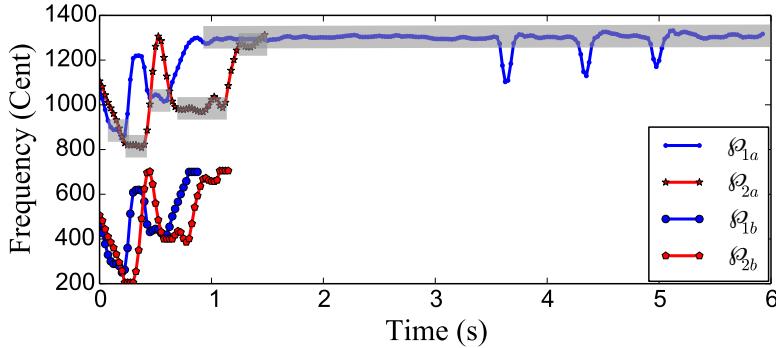


Figure 5.7: Original pitch contours (ϕ_{1a} , ϕ_{2a}) and duration truncated pitch contours (ϕ_{1b} , ϕ_{2b}) of two occurrences of a characteristic phrase of rāga Alahaiyā bilāval. The contours are transposed for a good visualization.

ity by showing the pitch contours of the different occurrences of three characteristic melodic phrases of the rāga Alahaiyā bilāval. We can clearly see that the duration of a phrase across its occurrences varies a lot and the steady melodic regions are highly varied in terms of the duration and the presence of melodic ornaments. Because of these factors detecting the occurrences of characteristic melodic phrases becomes a challenging task. Ideally, a melodic similarity measure should be robust to such high degree of melodic variations and, at the same time, it should be able to discriminate between different phrase categories and irrelevant melodic fragments (noise candidates).

In this section, we present two approaches that utilize specific melodic characteristics in IAM to improve melodic similarity. We describe a melodic abstraction process based on a partial transcription of melodies to handle large timing variations across occurrences of melodic phrases. Specifically for Carnatic music we also present a complexity weighting scheme that accounts for the differences in the melodic complexities of the phrases, a crucial aspect for melodic similarity in this music tradition. The following sections are based on our work presented in Gulati et al. (2015c).

5.3.1 Method

Before we present our approach in detail we first discuss the motivation and rationale behind it. A close examination of the occurrences of the characteristic melodic phrases in our dataset reveals that there is a pattern in the non-linear timing variations, which is also reported in Rao et al. (2014). In Figure 5.6 we show a few occurrences of three such melodic phrases. In particular, we see that the transient regions of a melodic phrase tend to span nearly the same time duration across different occurrences, whereas the stationary regions vary a lot in terms of the duration. In Figure 5.7 we further illustrate this by showing two occurrences of a melodic phrase (ϕ_{1a} and

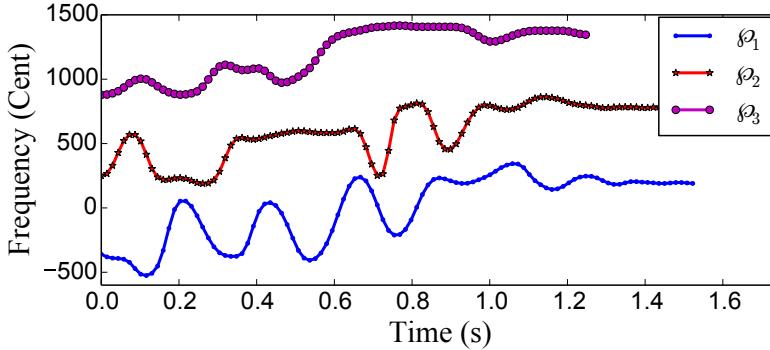


Figure 5.8: Pitch contours of three melodic phrases (ϕ_1 , ϕ_2 , ϕ_3). ϕ_1 and ϕ_2 are the occurrences of the same characteristic phrase and both are musically dissimilar to ϕ_3 .

ϕ_{2a}). The stationary **svara** regions are highlighted. We clearly see that the duration variation is prominent in the highlighted regions. To handle such large non-linear timing variations typically a non-constrained **DTW** distance measure is employed (Section 5.2.3). However, such a **DTW** variant is prone to noisy matches. Moreover, the absence of a band constraint renders it inefficient for computationally complex tasks such as pattern discovery (Section 5.4).

We put forward an approach that abstracts the melodic representation and reduces the extent of duration and pitch variations across the occurrences of a melodic phrase. Our approach is based on the partial transcription of the melodies. As mentioned earlier, melodic transcription in **IAM** is a challenging task. The main challenges arise due to the presence of non-discrete pitch movements such as smooth glides and **gamakas**. However, since the duration variation exists mainly during the steady **svara** regions, transcribing only the stable melodic regions might be sufficient. Once transcribed, we can then truncate the duration of these steady melodic regions and hence effectively reduce the amount of timing variations across the occurrences of a melodic phrase. Additionally, since the duration truncation also reduces the overall length of a pattern, the computational time for melodic similarity computation is also reduced substantially. Furthermore, this solution is independent of the distance measure used in the melodic similarity computation. Hence, it can be used even in the computationally complex tasks such as large scale pattern discovery, where the usage of distance lower bounds is imperative (Section 5.4).

The rapid oscillatory pitch movements (**gamakas**) in Carnatic music bring up another set of challenges for the melodic similarity computation. Very often, two musically dissimilar melodic phrases obtain a high similarity score owing to a similar pitch contour at a macro level. However, they differ significantly at a micro level. In Figure 5.8 we illustrate such a case where we show the pitch contours of three melodic phrases ϕ_1 , ϕ_2 and ϕ_3 , where ϕ_1 and ϕ_2 are the occurrences of the same melodic phrase and

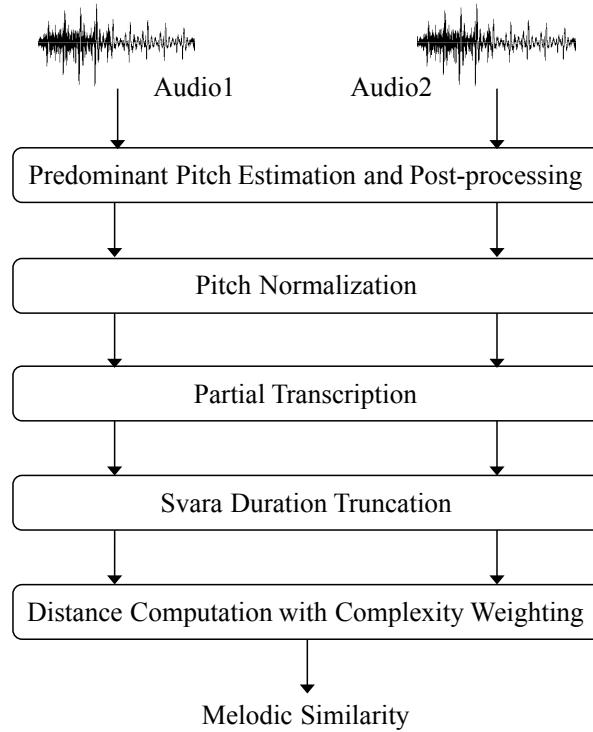


Figure 5.9: Block diagram of the improved methodology for computing melodic similarity.

both are musically dissimilar to ϕ_3 . Using the best performing variant of the similarity measure obtained in Section 5.2.3 (Table 5.1) we obtain a higher similarity score between the pairs (ϕ_1, ϕ_3) and (ϕ_2, ϕ_3) compared to the score between the pair (ϕ_1, ϕ_2) . This tendency of a high complexity time-series (higher degree of micro level variations) obtaining a high similarity score with another low complexity time-series is discussed in Batista et al. (2011). We follow their approach and apply a complexity weighting to account for the differences in the melodic complexities between phrases in the computation of melodic similarity.

We now proceed to describe our method in detail. The block diagram for computing melodic similarity is very similar to the one described in Section 5.2.1. The main differences are the added processing blocks for performing partial transcription, svara duration truncation and complexity weighting as shown in Figure 5.9. In the subsequent sections we describe every processing block in detail.

5.3.1.1 Predominant Pitch Estimation and post-processing

As done before, we represent melody by the predominant pitch in the audio recording. For its estimation, we follow exactly the same procedure and use the same para-

meter values as done in Section 5.2.1.1, which is explained in detail in Section 4.3.1. After estimating the predominant pitch we convert it from Hertz to Cent scale for the melody representation to be musically relevant (Section 4.3.3.1).

We proceed to post-process the pitch contours to remove the spurious pitch jumps lasting over a few frames as well as to smooth the pitch contours. We follow the procedure described in Section 4.3.2.2 and use exactly the same set of parameter values. The pitch contours are finally down-sampled to 67 Hz (Section 4.3.2.4). This sampling rate was found to be working well for both Carnatic and Hindustani music in our earlier study that evaluated five different sampling rates Section 5.2.3.

5.3.1.2 Pitch Normalization

We perform tonic normalization (Z_{tonic}) by considering the tonic of the lead artist as the reference frequency during the Hertz to Cent conversion as shown in Section 4.3.3.1. The tonic pitch is automatically identified by using M_{JS} method, which performed the best in our comparative evaluation (Section 4.2.2).

Tonic normalization does not account for the pitch of the octave transposed occurrences of a melodic phrase within a recording. In addition, estimated tonic pitch sometimes might be incorrect and a typical error is an offset of an octave or a fifth scale degree in some cases. To handle such cases, we propose a tetrachord normalization (Z_{tetra}). For this we analyse the difference (δ_m) in the mean frequency values of the two tonic normalized melodic phrases (ϕ_1, ϕ_2). We offset the pitch values of the phrase ϕ_1 by the frequency (Cents) in the set $\{-1200, -700, -500, 0, 500, 700, 1200, 1700, 1900\}$ that is closest to δ_m within a vicinity of 100 Cents. In addition to tetrachord normalization, we also experiment with mean normalization (Z_{mean}), which was reported to improve the performance in the case of Carnatic music (Section 5.2.3).

5.3.1.3 Partial Transcription

We perform a partial melody transcription to automatically segment and identify the steady **svara** regions in a melody. Note that even a partial transcription of the melodies is a non-trivial task, since we desire a segmentation that is robust to different melodic ornaments added to a **svara** where the pitch deviation from the mean **svara** frequency can be up to 200 Cents. In Figure 5.7 we show such an example of a steady **svara** region (ϕ_{1a} from 3-6 s) where the pitch deviation from the mean **svara** frequency is high due to added melodic ornaments. Ideally, the melodic region between 1 and 6 s should be detected as a single **svara** segment.

We segment the steady **svara** regions using a method described in Gulati et al. (2014b) (Section 4.5), which addresses the aforementioned challenges. A segmented **svara** region is then assigned a frequency value corresponding to the peak in an aggregated pitch histogram closest to the mean **svara** frequency. The pitch histogram is constructed for the entire recording and smoothed using a Gaussian window with a variance

of 15 Cents. As peaks of the normalized pitch histogram, we select all the local maxima where at least one peak-to-valley ratio is greater than 0.01. A detailed description of this method is provided in Section 4.5.

5.3.1.4 Svara Duration Truncation

After segmenting the steady *svara* regions in the melodies we proceed to truncate the duration of these regions. We hypothesize that, beyond a certain value Φ , the duration of these steady *svara* regions do not change the identity of a melodic phrase (i.e. the phrase category). We experiment with 7 different truncation durations $\Phi = \{0.1\text{ s}, 0.3\text{ s}, 0.5\text{ s}, 0.75\text{ s}, 1\text{ s}, 1.5\text{ s}, 2\text{ s}\}$ and select the one that results in the best performance. In Figure 5.7 we show an example of the occurrences of a melodic phrase both before ($\mathcal{P}_{1a}, \mathcal{P}_{2a}$) and after ($\mathcal{P}_{1b}, \mathcal{P}_{2b}$) the *svara* duration truncation using $\Phi = 0.1\text{ s}$. This example clearly illustrates that the occurrences of a melodic phrase after duration truncation exhibit lower degree of non-linear timing variations. We denote this method by M_{DT} .

5.3.1.5 Distance Computation

To measure the similarity between two melodic fragments we consider a DTW-based distance measure. Since the phrase segmentation is known beforehand, we use a whole sequence matching DTW variant. We consider the best performing DTW variant and the related parameter values for each music tradition as reported in Section 5.2.3. These variants were chosen based on an exhaustive grid search across all possible combinations and hence can be considered as optimal for this dataset. We use Sakoe-Chiba global band constraint Sakoe & Chiba (1978) with the width of the band as $\pm 10\%$ of the phrase length. Before computing the DTW distance we uniformly time-scale the two melodic fragments to the same length, which is the maximum of the lengths of the phrases. Notice that even though in Section 5.2.3 we found that a globally unconstrained DTW variant ($D_{DTW_L0_G90}$) is an optimal choice for computing melodic similarity in Hindustani music, we restrict ourselves to a band-width of 10% in this study. The main reason is because an unconstrained DTW variant due to its computational complexity is not suitable for a large scale pattern discovery task (Section 5.4). Since one the main objectives of studying melodic similarity within a supervised setup in this work is to be finally able to apply the findings in an unsupervised analysis, we now restrict to choices that are also available and feasible in an unsupervised analysis.

5.3.1.6 Complexity Weighting

The complexity weighting that we apply here to overcome the shortcoming of the distance measure in distinguishing two time series with different complexities is discussed in Batista et al. (2011). We apply a complexity weighting (Υ) to the DTW-

based distance (D_{DTW}) in order to compute the final similarity score $D_f = \Upsilon D_{DTW}$. We compute Υ as:

$$\Upsilon = \frac{\max(\zeta_i, \zeta_j)}{\min(\zeta_i, \zeta_j)} \quad (5.1)$$

$$\zeta_i = \sqrt[2]{\sum_{i=1}^{N-1} (\hat{p}_i - \hat{p}_{i+1})^2} \quad (5.2)$$

where, ζ_i is the complexity estimate of a melodic pattern of length N samples and \hat{p}_i is the pitch value of the i^{th} sample. We explore two variants of this complexity estimate. One of these variants is already proposed in Batista et al. (2011) and is described in Eq. 5.2. We denote this method variant by M_{CW1} . We propose another variant that utilizes melodic characteristics of Carnatic music. This variant takes the number of saddle points in the melodic phrase as the complexity estimate (Ishwar et al., 2013). This method variant is denoted by M_{CW2} . As saddle points we consider all the local minimas and the local maximas in the pitch contour which have at least one minima to maxima distance of half a semitone. Since such melodic characteristics are predominantly present in Carnatic music, the complexity weighting is not applicable for computing melodic similarity in Hindustani music.

5.3.2 Evaluation

5.3.2.1 Dataset and Annotations

For evaluation we use the same music collection as used in Section 5.2, which is described in Section 3.3.3. This collection enables a better comparison of the results with other studies since it has been used in several other studies for a similar task (Rao et al., 2014; Ross et al., 2012). However, we found a number of issues in the annotations of the melodic phrases, which we corrected and also extended the dataset by adding 25% more number of melodic phrase annotations as explained in Section 3.3.3. We denote this new dataset by MSD_{CM} and the comprising Carnatic and Hindustani sub-datasets by MSD_{CM}^{cmd} and MSD_{CM}^{hmd} , respectively. Similar to the way we perform evaluations in Section 5.2, we evaluate our approach separately on both Carnatic and Hindustani datasets. In Table 3.6 we summarize the relevant details of the dataset in terms of the number of artists, *rāgas*, audio recordings and total duration. In Table 3.8 we summarize the details of the annotated phrases in terms of their number of instances and basic statistics of the length of the phrases.

5.3.2.2 Setup, Measures and Statistical Significance

The experimental setup and evaluation measures used in this study are exactly the same as used for the comparative evaluation described in Section 5.2. We consider

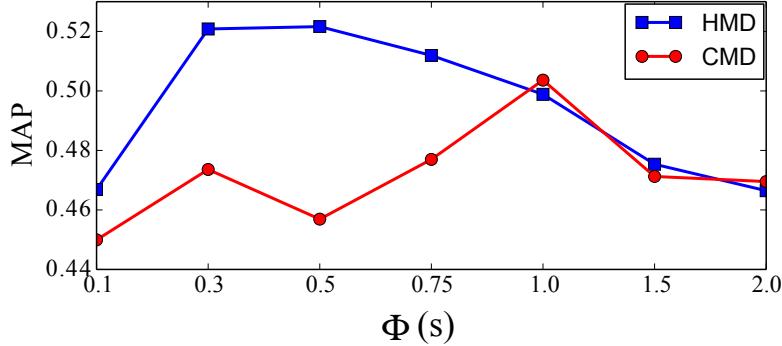


Figure 5.10: MAP scores for different duration truncation values (Φ) for the MSD_{CM}^{hmd} and the MSD_{CM}^{cmd} . Only the superscript in the name of the dataset is used in the figure legend.

each annotated melodic phrase as a query and perform a search across all the annotated phrases in the dataset (referred to as target search space). In addition to the annotated phrases, we add randomly sampled melodic segments (referred to as noise candidates) in the target space to simulate a real world scenario. We generate the starting time stamps of the noise candidates by randomly sampling a uniform distribution. The length of the noise candidates are generated by sampling the distribution of the duration values of the annotated phrases. The number of noise candidates added are 100 times the number of total annotations in the entire music collection. For every query we consider the top 1000 nearest neighbors in the search results ordered by the similarity value. A retrieved melodic phrase is considered as a true hit only if it belongs to the same phrase category as the query. As a baseline in this study we consider the same method as described in this section but without applying the *svara* duration truncation and complexity weighting procedure. We denote this baseline method by M_B .

To assess the performance of our approach and the baseline method we use mean average precision (MAP), a common measure in information retrieval (Manning et al., 2008). To assess if the difference in the performance of any two methods is statistically significant we use the Wilcoxon signed rank-test (Wilcoxon, 1945) with $p < 0.01$. To compensate for multiple comparisons, we apply the Holm-Bonferroni method (Holm, 1979).

5.3.3 Results and Discussion

In Table 5.3, we summarize the MAP scores and the standard deviation of the average precision values obtained using the baseline method (M_B), the method that uses duration truncation (M_{DT}) and the ones using the complexity weighting (M_{CW1} , M_{CW2}), for both the MSD_{CM}^{cmd} and the MSD_{CM}^{hmd} . Note that M_{CW1} and M_{CW2} are only applicable to the MSD_{CM}^{cmd} (Section 5.3.1.6).

	$\text{MSD}_{\text{CM}}^{\text{hmd}}$			
Norm	\mathbf{M}_B	\mathbf{M}_{DT}	\mathbf{M}_{CW1}	\mathbf{M}_{CW2}
Z_{tonic}	0.45 (0.25)	0.52 (0.24)	-	-
Z_{mean}	0.25 (0.20)	0.31 (0.23)	-	-
Z_{tetra}	0.40 (0.23)	0.47 (0.23)	-	-

	$\text{MSD}_{\text{CM}}^{\text{cmd}}$			
Norm	\mathbf{M}_B	\mathbf{M}_{DT}	\mathbf{M}_{CW1}	\mathbf{M}_{CW2}
Z_{tonic}	0.39 (0.29)	0.42 (0.29)	0.41 (0.28)	0.41 (0.29)
Z_{mean}	0.39 (0.26)	0.45 (0.28)	0.43 (0.27)	0.45 (0.27)
Z_{tetra}	0.45 (0.26)	0.50 (0.27)	0.49 (0.28)	0.51 (0.27)

Table 5.3: MAP scores for the two datasets $\text{MSD}_{\text{CM}}^{\text{hmd}}$ and $\text{MSD}_{\text{CM}}^{\text{cmd}}$ for the four method variants \mathbf{M}_B , \mathbf{M}_{DT} , \mathbf{M}_{CW1} and \mathbf{M}_{CW2} and for different normalization techniques. Standard deviation of average precision is reported within round brackets.

We first analyse the results for the $\text{MSD}_{\text{CM}}^{\text{hmd}}$ dataset. From Table 5.3 (upper half), we see that the proposed method variant that applies a duration truncation performs better than the baseline method for all the normalization techniques. Moreover, this difference is found to be statistically significant in each case. The results for the $\text{MSD}_{\text{CM}}^{\text{hmd}}$ in this table correspond to $\Phi = 500$ ms, for which we obtain the highest accuracy compared to the other Φ values as shown in Figure 5.10. Furthermore, we see that Z_{tonic} results in the best accuracy for the $\text{MSD}_{\text{CM}}^{\text{hmd}}$ for all the method variants and the difference is found to be statistically significant in each case. From Table 5.3, we notice a high standard deviation of the average precision values. This is because some occurrences of melodic phrases possess a large amount of melodic variation (acting as outliers), and therefore, the average precision value of the retrieved results using them as a query is much smaller compared to the other occurrences. In Figure 5.11, we show a boxplot of average precision values for each phrase category and for both \mathbf{M}_B and \mathbf{M}_{DT} to get a better understanding of the results. We observe that with an exception of the phrase category H_2 , \mathbf{M}_{DT} consistently performs better than \mathbf{M}_B for all the other phrase categories. A close examination of this exception reveals that the error often is in the segmentation of the steady *svara* regions of the melodic phrases corresponding to H_2 . This can be attributed to a specific subtle melodic movement in H_2 that is confused by the segmentation method as a melodic ornament instead of a *svara* transition, leading to a segmentation error.

We now analyse the results for the $\text{MSD}_{\text{CM}}^{\text{cmd}}$ dataset. From Table 5.3 (lower half), we see that using the method variants \mathbf{M}_{DT} , \mathbf{M}_{CW1} and \mathbf{M}_{CW2} we obtain reasonably higher

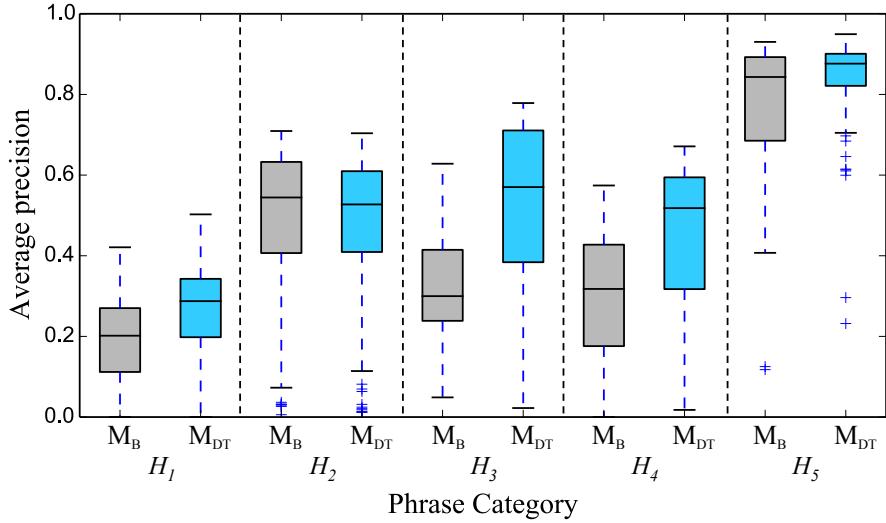


Figure 5.11: Boxplot of the average precision values obtained using M_B and M_{DT} for each melodic phrase category for the MSD_{CM}^{hmd} . These values correspond to Z_{tonic} .

MAP scores compared to the baseline method M_B and the difference is found to be statistically significant for each method variant across all normalization techniques. This MAP score for M_{DT} corresponds to $\Phi = 1$ s, which is considerably higher than the MAP scores for other Φ values as shown in Figure 5.10. We also see that M_{CW2} performs slightly better than M_{CW1} and the difference is found to be statistically significant only in the case of Z_{tetra} . We do not find any statistically significant difference in the performance of methods M_{DT} and M_{CW2} . Unlike in the case of the MSD_{CM}^{hmd} dataset, for the MSD_{CM}^{cmd} dataset Z_{tetra} results in the best performance with a statistically significant difference compared to the other normalization techniques across all method variants. We now analyse the average precision values for every phrase category for M_B , M_{DT} and M_{CW2} . Since M_{CW2} performs slightly better than M_{CW1} we only consider M_{CW2} for this analysis. In Figure 5.12, we see that M_{DT} performs better than M_B for all phrase categories. We also observe that M_{CW2} consistently performs better than M_B with the sole exception of C_2 . This exception occurs because M_{CW2} presumes a consistency in terms of the number of saddle points across the occurrences of a melodic phrase, which does not hold true for C_2 . This is because phrases corresponding to C_2 are rendered very fast and the subtle pitch movements are not the characteristic aspect of such melodic phrases. Hence, the artists often take the liberty of changing the number of saddle points.

Overall, we see that duration truncation of steady melodic regions improves the performance in both MSD_{CM}^{hmd} and MSD_{CM}^{cmd} datasets. This reinforces our hypothesis that elongation of steady svara regions (up to a permissible limit) in the melodies of IAM in the context of the characteristic melodic phrase does not change the musical iden-

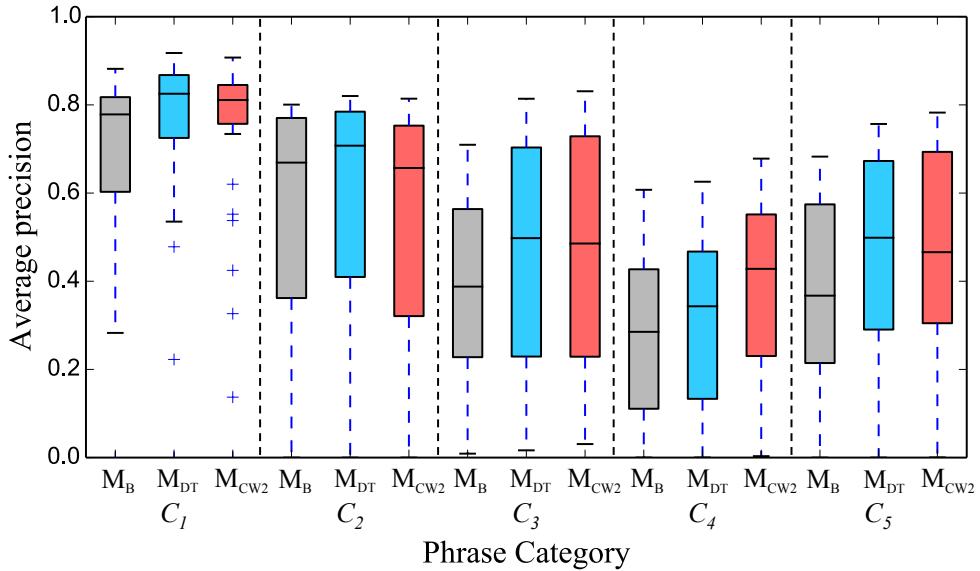


Figure 5.12: Boxplot of the average precision values obtained using methods M_B, M_{DT} and M_{CW2} for each melodic phrase category for the MSD_{CM}^{cmd}. These values correspond to Z_{tetra}.

tity of the phrase. This correlates with the concept of *nyās svara* (Section 2.3.2), where the artist has the flexibility to stay and elongate a single *svara*. A similar observation was reported in Rao et al. (2014), where the authors proposed to learn the optimal global DTW constraints a priori for each pattern category. However, as they report, their proposed solution could not improve the performance. Further comparing the results for the MSD_{CM}^{hmd} and MSD_{CM}^{cmd} datasets we notice that Z_{tonic} results in the best performance for the MSD_{CM}^{hmd} and Z_{tetra} for the MSD_{CM}^{cmd}. This can be attributed to the fact that the number of the pitch-transposed occurrences of a melodic phrase is significantly higher in the MSD_{CM}^{cmd} compared to the MSD_{CM}^{hmd} (Section 5.2.3). Also, since the non-linear timing variability in the MSD_{CM}^{hmd} is very high, any normalization (Z_{mean} or Z_{tetra}) that involves a decision based on the mean frequency of the phrase is likely to fail.

5.4 Melodic Pattern Discovery

As argued in Section 5.1, the potential of a pattern-based melodic analysis in characterization of *rāgas*, compositions and artists in IAM gets severely restricted in a supervised experimental setup. As described before, this is mainly due to the factors such as limited dataset size, knowledge bias and human errors in the melodic pattern corpus provided by domain experts. Therefore, to overcome these issues we follow an unsupervised methodology to discover melodic patterns in sizable audio music

collections of **IAM**. Since a quantitative evaluation of an unsupervised method for sizable datasets is difficult and rather ill-defined, improving such methods and learning optimal values of the system parameters becomes a challenging task. We therefore studied the computation of melodic similarity, a crucial component in a melodic pattern discovery method in a supervised manner in Section 5.2 and Section 5.3. The learnings from these supervised studies aid our unsupervised melodic pattern discovery method presented in this section.

From our literature review presented in Section 2.5 and Section 2.4 we see that several approaches have been proposed for extracting different kinds of repeating structures in music, including long-duration repetitions such as different sections of a music piece (Serrà et al., 2012; Goto, 2006; Paulus et al., 2010), relatively small-duration repetitions being themes, riffs (Hsu et al., 2001), and melodic motifs (Meredith et al., 2002; Collins, 2011; Janssen et al., 2013). While there exists a number of approaches for motivic discovery in sheet music (Meredith et al., 2002; Cambouropoulos, 2006; Conklin & Anagnostopoulou, 2001; Lartillot, 2005b), there are fewer approaches that work on audio music recordings (Dannenberg & Hu, 2003). This can be attributed to the audio-symbolic gap (Collins et al., 2014), which is argued to be bridged by a reliable automatic transcription system to abstract the audio music content into musically meaningful discrete symbols. Although, for several music traditions such as **IAM** melodic transcription still remains a challenging and a rather ill-defined task. A detailed account of the shortcomings in the existing pattern discovery methods in the context of their applicability to melodies in **IAM** is presented in our literature review in Section 2.4.2 and Section 2.5.2.3. We see that there exists a wide scope for developing methodologies for the discovery and analysis of short duration melodic patterns (or motifs) in large audio music collections. Approaches for motif discovery can benefit from the literature in the domain of time series analysis such as time series representation (Lin et al., 2003), core pattern discovery methods (Mueen et al., 2009), and search and indexing techniques (Rakthanmanon et al., 2013). We now proceed to describe our methodology for melodic pattern discovery in sizable audio music collections of **IAM**.

5.4.1 Method

Our approach consists of four main blocks as shown in Figure 5.13. The data processing block generates pitch subsequences from every audio recording in the music collection, which are potential pattern candidates. The intra-recording pattern discovery block performs an exact pattern discovery by detecting the closest subsequence pairs within an audio recording (referred to as seed patterns). The inter-recording pattern detection block considers each seed pattern as a query and searches for its occurrences in the entire music collection. The rank refinement block reorders a ranked list of search results by recomputing melodic similarity using a more sophisticated similarity measure. The following description is based on our work presented in Gulyati et al. (2014c).

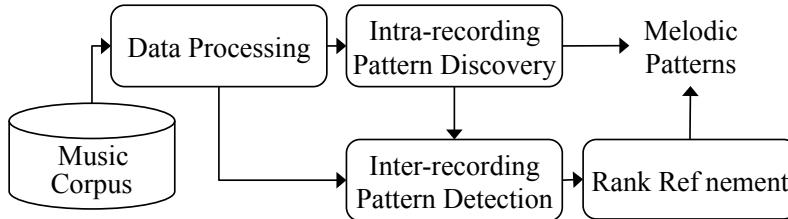


Figure 5.13: Block diagram of the proposed approach for melodic pattern discovery in large audio collections of IAM.

We choose to perform first an intra-recording pattern discovery because melodic patterns are repeated within a music piece in IAM. Moreover, the scalability of the computational approaches considered here for discovering patterns at the level of the entire music collection is questionable. To confirm this hypothesis, we conducted an experiment using a state-of-the-art algorithm for time series motif discovery (Mueen et al., 2009), with a trivial modification to extract the top K motifs. Using just 16 hours of audio data (amounting to around 20 million pitch samples), the algorithm could discover only 40 melodic patterns in 24 hours using Euclidean distance. Besides pattern pairs being from the same recording, only a few of the obtained pattern pairs were melodically similar and meaningful. This is expected as we found in Section 5.2.3 that Euclidean distance is not appropriate for handling large-non linear timing variations present across occurrences of melodic patterns. In order to scale pattern discovery for hundreds of hours of audio data using a computationally complex DTW-based distance measure we choose to first perform pattern discovery within an audio recording.

Before we proceed to describe our method it should be noted that during the course of this dissertation several processing blocks and system parameters presented in the subsequent sections have evolved. The methodology presented in this section is based on our work reported in Gulati et al. (2014c). However, after that study, based on our findings in Gulati et al. (2015b) and in Gulati et al. (2015b) we have modified the procedure followed in several processing blocks and system parameters. In order to facilitate reproducibility of our experiments and research outcomes, along with the original description of the method (Gulati et al., 2014c) we also present the modifications done for the new (the most recent) variant of the method. Wherever applicable the modifications are presented during the description of the processing block. Since the evaluation methodology we followed involve cumbersome listening tests, the results presented in this section are only corresponding to the initial variant of the method presented in Gulati et al. (2014c).

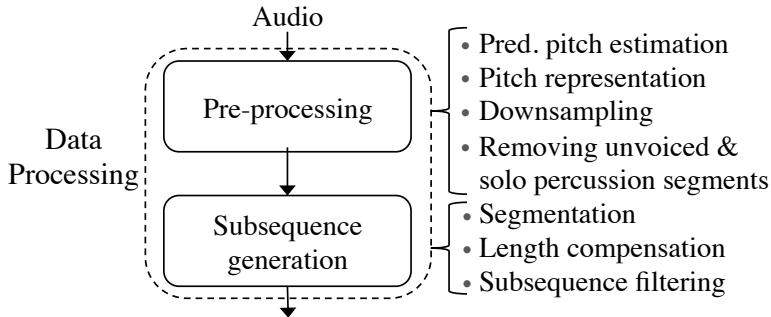


Figure 5.14: Block diagram of the data processing block in melodic pattern discovery task.

5.4.1.1 Pre-processing

The steps involved in the pre-processing block are shown in Fig. 5.14. A description of each of these steps is given below:

a) Predominant Pitch Estimation and Representation We consider melody as the predominant pitch in the audio signal. For estimating predominant pitch we follow the procedure we used in Section 5.2, which is described in detail in Section 4.3.1. We use a frame size of 46 ms and a hop size of 4.44 ms. Noticeably, the predominant pitch estimation method that we use also performs voicing detection, which is used in the later part of our data processing methodology to filter unvoiced segments (Figure 5.14). We do not perform any post-processing on the estimated pitch contours.

For the melody representation to be musically relevant, the pitch values are converted from Hertz to Cents (Section 4.3.3.1). In order to compare melodies across different artists and recordings we additionally consider the tonic pitch of the lead artist in the recording as the reference frequency during this conversion. The tonic of the lead artist for each recording in the collection is identified automatically using MJS method, which is found to be the best performing method for this task in our comparative evaluation (Section 4.2).

In the new variant of our method we post-process the predominant pitch contours as described in Section 4.3.2. We perform median and Gaussian filtering to remove spurious pitch jumps lasting over a few samples and to smooth the pitch contours. In addition, we also interpolate short non-voiced segments that usually correspond to intra-phrase breath pauses or to consonants in the lyrics (Section 4.3.2.3).

b) Downsampling As mentioned above, we estimate predominant pitch at a sampling rate of around 225 Hz. However, in order to reduce the computational cost, we downsample the predominant pitch sequence (Figure 5.14). We could not find any study

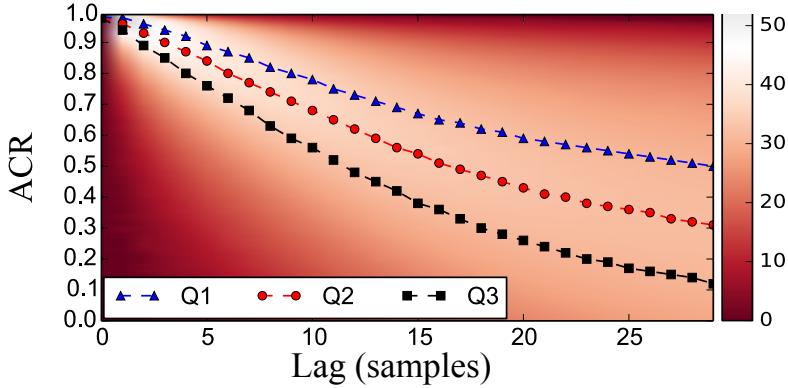


Figure 5.15: Histograms of autocorrelation (ACR) values (histogram value is indicated by the colormap on the right: for ease of visualization, we compress the range of the histogram values by taking its fourth root). Q1, Q2 and Q3 denote the three quartile boundaries of the histogram.

that systematically reports the effect of sampling rate on melodic similarity in IAM⁴⁹. In such a case, we derive an optimal sampling rate by analyzing the ACR of short-time pitch segments generated using a sliding window of 2 s. We compute the ACR of all possible pitch segments in the entire dataset for different lags l , $l \in \{0, 1, \dots, 30\}$, and examine the histogram of normalized ACR values at each lag (Figure 5.15). We select the lag at which the third quartile Q3 has an ACR value of 0.8, which corresponds to a sampling rate of 22.22 ms (or 45 Hz). We informally found that this sampling rate generally preserves melodic nuances and rapid pitch movements in Carnatic music while reducing the computational requirements of the task. Note that in the new variant of our method, we use the optimal sampling rate derived from our comprehensive quantitative evaluations in Gulati et al. (2015b) (see Section 5.2).

c) Solo Percussion Removal As described in Section 4.4, a concert of Carnatic music typically contains a solo percussion section, referred to as a *tani* section, which can last up to 2–25 minutes. We find that the predominant pitch estimation method employed in this study often tracks pitch corresponding to the *mṛdaṅgam* strokes instead of detecting *tani* sections as non-voiced segments. This poses a challenge for melodic pattern discovery as there are several repeating percussion patterns in *tani* sections, which are often discovered as the closest melodic pattern pairs. In Section 4.4, we explain this issue in detail and describe a classification-based approach to detect *tani* sections in audio recordings of Carnatic music. Once detected, we can simply discard the pitch samples corresponding to these sections and overcome the challenge. In addition to avoiding the unwanted melodic patterns being present in the output,

⁴⁹Note that the current study was performed before our supervised studies presented in Section 5.2 and Section 5.3, where we analyse the influence of sampling rate on melodic similarity.

by discarding the **tani** sections we also reduce the computational complexity of our method.

5.4.1.2 Subsequence Generation

In this processing block we generate melodic pattern candidates from the resultant pitch representation (Figure 5.14). The steps involved in generating candidate subsequences are described below.

a) Segmentation As seen in Section 5.2.3, an accurate segmentation of melodic patterns has a big impact on the computation of melodic similarity, and eventually on the retrieval accuracy of melodic patterns. In the literature (Section 2.5.2.1) there are several models studied for melodic segmentation (Cambouropoulos, 2006; Müller et al., 2009; Cambouropoulos, 2001a). Pearce et al. (2008) and Rodríguez L. et al. (2014) provide a comparison of a number of such models. However, none of these models is directly applicable on the continuous melody representation we use for **IAM**. Due to the lack of such studies and reliable models for segmentation of melodic patterns in **IAM**, we use a brute-force approach for generating pattern candidates. We generate candidate pitch subsequences by using a sliding window of length W with a hop size of one sample. Given no quantitative studies investigating the length of the melodic patterns in **IAM**, we make a choice of $W = 2$ s based on recommendations from a few professional musicians.

Since unvoiced segments are removed from the pitch sequence at the pre-processing step (Figure 5.14), a pattern candidate can include pitch samples separated by more than W seconds. To handle these cases, we use the time stamps of the first sample (T_1) and the last sample (T_2) in the subsequence. We filter out all subsequences for which $T_2 - T_1 > W + \phi$. We select $\phi = 0.5$ s to account for the short pauses during a phrase rendition. This value was empirically set to differentiate between inter- and intra-phrase pauses.

In the new variant of our method, the processing step described in the previous paragraph becomes redundant, and is therefore not applied. Since in this variant the short-duration unvoiced regions in the predominant pitch contours are interpolated, there will not be any situation where $T_2 - T_1 > W + \phi$.

b) Subsequence Filtering One of the challenges in melodic pattern discovery is the presence of combinatorial redundancy in the form of musically trivial patterns in the output (Lartillot, 2005b). One such redundancy in our case is that of a melodic pattern comprising a single **svara**. Instead of removing such musically uninteresting patterns after they are discovered, we detect and discard such subsequences (or pattern candidates) during the pre-processing step (Figure 5.14). The criterion for discarding

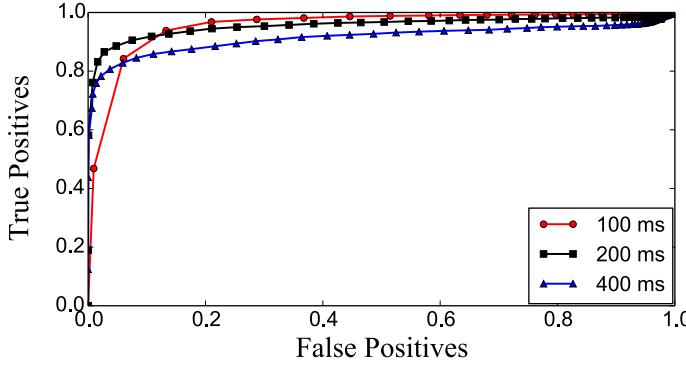


Figure 5.16: ROC curves for ‘flat’ and ‘non-flat’ region classification for different values of window length (W_σ) used for selecting an optimal value of the standard deviation.

such subsequences is summarized below:

$$v = \sum_{i=0}^{\hat{W}} \Theta(\sigma_i \geq \tilde{\sigma}_v), \quad (5.3)$$

where v is the flatness measure of a subsequence, \hat{W} denotes its number of samples, $\Theta(z)$ is a Heaviside step function yielding $\Theta(\text{true}) = 1$ and $\Theta(\text{false}) = 0$, and σ_i is the standard deviation at the i -th sample of a subsequence, computed using a window of length W_σ centered at sample i . The threshold $\tilde{\sigma}_v$ determines if a sample belongs to a flat region or not. In order to determine the optimal values of W_σ and $\tilde{\sigma}_v$, we manually labeled a number of regions in pitch contour as ‘flat’ and ‘non-flat’ for 4 excerpts in our dataset. We iterated over different parameter values and analysed the resultant ROC curve shown in Figure 5.16. Doing so, we found that $W_\sigma = 200$ ms resulted in the best performance and that the knee of the curve corresponded to $\tilde{\sigma}_v = 45$ Cents. Having a value of v for each subsequence, we finally filter out the ones for which $v \leq \tilde{v}\hat{W}$, using $\tilde{v} = 0.8$. The latter was set by visual inspection.

In the new variant of our pattern discovery method, we use the segmentation approach described in Section 4.5.1.2 to detect the stable **svara** regions in melodies. The pitch samples in a subsequence that correspond to these stable **svara** regions are regarded as ‘flat’, which is analogous to $\sigma_i \geq \tilde{\sigma}_v$ being ‘True’ for those samples. All other parameters remain the same as described above. This change is done because the segmentation approach produces more reliable estimates of the stable melodic regions compared to the simple measure using local standard deviation of the pitch samples.

5.4.1.3 Intra-recording Pattern Discovery

In this step our aim is to discover melodic patterns within each recording in the audio collection. We perform an exact pattern discovery by computing the similarity

between every possible subsequence pair obtained within an audio recording. Thus, the computational complexity of this task is quadratic ($\mathcal{O}(n^2)$) over the number of subsequences. We regard the top $K_{\text{intra}} = 25$ closest subsequence pairs in each recording as seed patterns. We omit overlapping subsequences in order to avoid trivial matches and additionally constrain the top K_{intra} seed pattern pairs to be mutually non-overlapping. Due to this constraint for some recordings we obtain less than 25 pattern pairs.

Melodic Similarity We compute melodic similarity between two subsequences using a DTW-based distance measure (Section 2.6.2). This choice is supported by our findings in Section 5.2.3, wherein we showed that a DTW-based distance measure outperforms Euclidean distance measure in melodic similarity computation in IAM. We use a step condition of $\{(1,0), (1,1), (0,1)\}$ and the squared Euclidean distance as the cost function. The accumulated cost matrix for this step condition is computed as described in Eq. 2.9. We do not use any penalty for insertion and deletion. In addition, we apply the Sakoe-Chiba global constraint with the band width set to 10% of the pattern length. This constraint was found to be sufficiently large for accounting time warpings in melodic repetitions in Carnatic music (see Table 5.1). For the case of Hindustani music, an unconstrained DTW variant is shown to perform the best. However, to make the system computationally tractable we finally choose 10% band width for both the music traditions. Notice that these parameter values are close to the optimal settings but not the most optimal settings for a DTW variant that we obtained in Table 5.1. We make these compromises in order to avail the lower bounding techniques as explained in the subsequent sections.

Lower Bounding DTW The computational complexity of a brute-force pattern discovery system using a DTW distance measure is quadratic ($\mathcal{O}(n^2)$) over both the number of subsequences and the length of a subsequence. For a dataset as big as ours that contains millions of subsequences (pattern candidates), where the length of each subsequence is around 100 samples, the system becomes extremely demanding in terms of the CPU time. Even after reducing the number of pattern candidates by filtering out subsequences in the pre-processing step and reducing the length of the subsequences by downsampling the pitch representation, it is not practically feasible to perform the task. One of the ways to address this problem is to use indexing or lower bounding techniques with which we can prune subsequence pairs that cannot possibly be the best matches. Pruning of the subsequence pairs reduces the number of times the DTW computation is done, and thus make DTW distance computations tractable for datasets with large number of subsequences.

In literature, there are several methods proposed for indexing the DTW distance (Keogh & Ratanamahatana, 2004; Vlachos et al., 2003; Kim et al., 2001). These methods differ in terms of their computational complexities and the type of indexing techniques they employ (approximate or exact). In this study to speed up DTW compu-

tations we use the exact DTW indexing technique (Keogh & Ratanamahatana, 2004) and apply cascaded lower bounds as explained in Rakthanmanon et al. (2013). This method is parameter free, it does not require any pre-processing of the data and there is no constraint on the minimum and maximum query length. In particular, we use LB_KIM_FL (first-last) lower bound (Kim et al., 2001) and LB_Keogh bound (Keogh & Ratanamahatana, 2004) for both query to reference and reference to query matching (denoted by LB_Keogh_EQ and LB_Keogh_EC, respectively). In LB_Keogh_EQ lower-bounding envelopes are computed for the query pattern, and in LB_Keogh_EC, they are computed for the reference. Besides, we apply early abandoning, both during the computation of lower bounds as well as during the DTW distance computation. These lower bounding techniques are explained in Section 2.6.3, however, for a more detailed explanation we refer to Rakthanmanon et al. (2013).

In Algorithm 3, we show the pattern discovery routine that uses cascaded lower bounds to speed up DTW. This pseudo-code provides a better understanding of the pruning procedure and puts in context the utility of using different lower bounds. Note that, in this routine we show the discovery of only the closest subsequence pair. With a trivial addition of incorporating a priority queue that stores the K most closest subsequence pairs at each step, we extend it to our use-case of discovering the top K closest melodic patterns. We further optimize this routine by pre-computing the subsequence envelopes used in the computation of LB_Keogh bound (Section 2.6.3).

Algorithm 3 Discovering the closest subsequence pair using the DTW distance and cascaded lower bounds.

```

Input: array  $\rho$  containing  $N$  number of subsequences
best_so_far = infinity;
for  $i=0$ ;  $i \leq N - 1$ ;  $i++$  do
    for  $j=0$ ;  $j \leq N - 1$ ;  $j++$  do
        dist_FL = LB_KIM_FL( $\rho_i$ ,  $\rho_j$ )
        if dist_FL < best_so_far then
            dist_EQ = LB_Keogh( $\rho_i$ ,  $\rho_j$ )
            if dist_EQ < best_so_far then
                dist_EC = LB_Keogh( $\rho_j$ ,  $\rho_i$ )
                if dist_EC < best_so_far then
                    true_dist = DTW( $\rho_i$ ,  $\rho_j$ )
                    if true_dist < best_so_far then
                        best_so_far = true_dist
                        closest_pair_index = ( $i, j$ )

```

Pattern Length Compensation Along with the local non-linear time warping, the overall length of a melodic pattern may also vary across repetitions. For example, a melodic pattern of length 2 s might be sung in 2.2 s in a different position in the recording. We handle this by using multiple time scaled versions of a subsequence in

the distance computation. Performing appropriate uniform time-scaling prior to DTW is known to produce tighter lower bounds (Zhu & Shasha, 2003b), which is similar to a technique referred to as local DTW. It should be noted that such timing variations could as well be addressed by using a subsequence variant of DTW. However, the lower bounding techniques we use to speed up the DTW distance computation do not work for the subsequence variant of the DTW.

For every subsequence, we generate five subsequences by uniformly time scaling it by a factor of $\Omega \in \{0.9, 0.95, 1, 1.05, 1.1\}$, such that the length of the resulting subsequences is W . We use cubic interpolation for uniformly time scaling a subsequence. Creating five uniform time scaled subsequences for each subsequence increases the computational cost by a factor of 25. We observe that the distance between a subsequence pair $X_{1.0}$ and $Y_{1.05}$ is very close to the distance between the pair $X_{1.05}$ and $Y_{1.1}$ (the sub-index denotes the interpolation factor Ω). Thus by following this rationale and using this approximation, we can avoid the distance computation between 16 of the 25 combinations without a significant compromise on accuracy. We only retain the combinations for which the difference between the interpolation factors of subsequence pairs are unique.

5.4.1.4 Inter-recording Pattern Detection

After discovering melodic patterns within each audio recording, we now proceed to detect their occurrences in all the recordings in the collection. For this, we consider every seed pattern as a query and perform an exhaustive search over all the subsequences obtained from the entire audio music collection. For every seed pattern we store top $K_{\text{inter}} = 200$ closest matches (referred to as search patterns). To avoid redundancy in the search results, we constrain search patterns for every seed pattern to be mutually non-overlapping. Similar to the intra-recording pattern discovery step, here also for every subsequence we consider 5 uniformly time scaled subsequences in the distance computation. Furthermore, for detecting occurrences of seed patterns in other recordings we use the same similarity measure and lower bounding techniques as used in Intra-recording pattern discovery. The pattern detection routine is a small modification of the routine shown in Algorithm 3. Instead of a single subsequence array, we would have two arrays: one of which comprises the seed patterns (queries) and the other comprises subsequences obtained from all the recordings (target candidates).

5.4.1.5 Rank Refinement

As mentioned before, the lower bounds used for speeding up the distance computations are not valid for any variant of the DTW distance. This constraint governed the choices made for the DTW variant and the parameter settings for computing melodic similarity in both the intra and the inter-recording pattern processing blocks. However, once the top matches are found, nothing prevents us from reordering the ranked

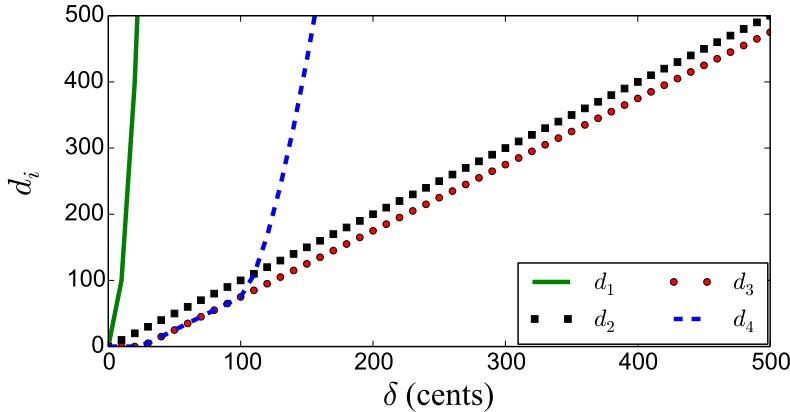


Figure 5.17: Output of the four different distance measures ($d_i, i \in \{1 \dots 4\}$) as a function of cityblock distance δ .

list using any variant of the DTW distance. This is because the number of top matches we consider ($K_{\text{inter}} = 200$) per query is orders of magnitude smaller than the total number of subsequences obtained from the entire audio collection. For every query pattern, we now recompute the melodic similarity with its top K_{inter} search patterns using a more robust and a better performing variant of the **DTW** distance.

During rank refinement, we select a **DTW** step condition of $\{(1,2), (1,1), (2,1)\}$ to avoid some pathological warping of the path. This step condition, which also acts as a local constraint was shown to better model the melodic similarity in Section 5.2.3. Furthermore, we investigate four different distance measures $d_i, i = 1, \dots, 4$, used in the computation of the **DTW** cost matrix. These distance measures are described below in Eq. 5.4. For a better understanding of the distance measures we also show them in Figure 5.17.

$$\begin{aligned}
 d_1 &= \delta \\
 d_2 &= \delta^2 \\
 d_3 &= \begin{cases} \delta - 25, & \text{if } \delta > 25 \\ 0, & \text{otherwise} \end{cases} \\
 d_4 &= \begin{cases} (\delta - \varphi_1)^{1.5} + \varphi_2, & \text{if } \delta > 100 \\ d_3, & \text{otherwise} \end{cases}
 \end{aligned} \tag{5.4}$$

where $\delta = |\hat{p}_1 - \hat{p}_2|$ is the city block distance between two pitch values and all numeric values are in Cent scale. We set $\varphi_1 = 99.55$ and $\varphi_2 = 74.7$ to maintain point and slope continuity of the function. The formulation for these different distance measures (d_i) is inspired by our own experience and some of the approaches we find in the literature (Ishwar et al., 2013; Rao et al., 2014). We denote the four variants of the rank refinement method by V_i , where $i \in \{1 \dots 4\}$.

5.4.2 Evaluation

5.4.2.1 Music Collection

The music collection used for studying the task of pattern discovery in this section comprises 365 hours Carnatic music. It contains 1764 audio recordings, which are carefully compiled as a part of the CompMusic Carnatic music corpus (Section 3.2.2). As explained in Section 3.2.2, these audio recordings are ripped from commercially released music CDs. The selected music material is diverse in terms of the number of artists, gender of lead artists, number of different *rāgas*, year of release and various forms within Carnatic music.

5.4.2.2 Evaluation Methodology

One of the challenges in an unsupervised data-driven approach such as the one we use for discovering melodic patterns is its evaluation. We here perform a quantitative evaluation based on expert feedback. For the entire dataset we obtain over 15 million search patterns for each of the rank refinement methods. We divide seed patterns into three categories based on the distance between the seed pairs, which we denote by Δ . The distribution of the distance Δ between the seed pattern pairs is shown in Figure 5.18. Then, to have an equal representation from the range of values of Δ , 200 seed pairs equally distributed among these categories are randomly selected for evaluation. Seed category boundaries are $\mu \pm 1.5\sigma$, where μ and σ are the mean and the standard deviation of the distribution of Δ . For every selected seed pattern we consider the first 10 search patterns for each of the four rank refinement methods for evaluation. Thus, in total, we obtain 200 seed pairs and 8,000 search patterns for expert evaluation.

Expert evaluation is performed by a professional Carnatic musician who has received over 20 years of music education. For examining similarity between two melodic patterns, the musician listened to the audio fragments corresponding to these patterns and scored a 0 for melodically dissimilar and a 1 for melodically similar pattern. The musician annotated melodic similarity for each seed pair and between the seed and its search patterns for every rank refinement method.

To quantify the musician’s assessment of the similarity between the melodic patterns we use **MAP**, a typical evaluation measure in information retrieval (Manning et al., 2008), which is also very common in **MIR**. This way, we have a single number to evaluate the performance of the four different rank refinement methods. Since we do not have ground-truth annotations of melodic patterns for our dataset, total number of occurrences of different patterns is unknown. Therefore, while computing the **MAP** scores we consider the total number of relevant patterns as the number of relevant patterns retrieved in the top 10 search results. For assessing statistical significance we use the Mann-Whitney U test (Mann & Whitney, 1947) with $p < 0.05$. To compensate for multiple comparisons, we apply the Holm-Bonferroni method (Holm, 1979). Thus,

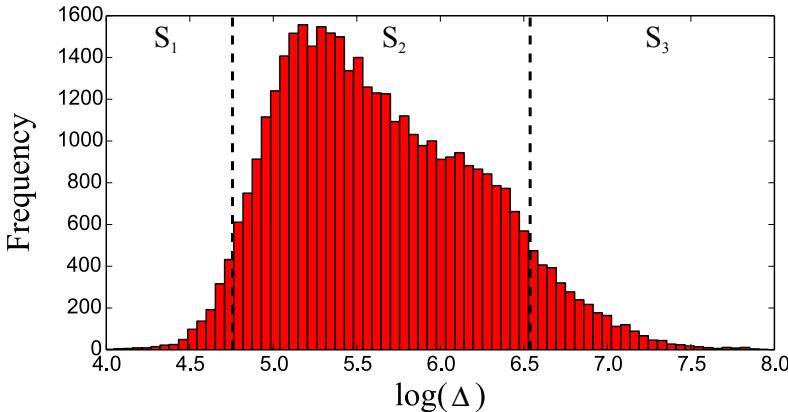


Figure 5.18: Distance distribution of seed patterns. Three seed pattern categories are marked by S_1 , S_2 and S_3 .

eventually we use a much more stringent criterion than $p < 0.05$ for measuring statistical significance. We also use ROC curves to analyse the separation between the distance distribution of melodically similar and dissimilar subsequences (Manning et al., 2008).

5.4.3 Results and Discussion

Before presenting the results of the evaluation of the discovered patterns, we provide details in terms of the number of patterns and DTW computations done at each step. Our dataset comprising 365 hours of audio data contains nearly 300 million pitch samples (considering 225 Hz as the sampling rate of the pitch contours). In a brute force segmentation scenario that would amount to roughly the same number of pattern candidates. However, since we downsample the pitch contours and filter out musically trivial patterns, we retain around 17.5 million pattern candidates after data pre-processing step. In the intra-recording pattern discovery block, for all the recordings, nearly 1.41 trillion distance computations are done to obtain 79,172 seed patterns. In the inter-recording pattern detection block, nearly 12.42 trillion distance computations are done to obtain 15 million search patterns for each variant of the rank-refinement method. These numbers give us an idea about the computational complexity of the task and shows the scale at which this study is performed.

We now analyse the contribution of different lower bounds in pruning the search space. In Table 5.4 we show in percentage the number of times the program counter exits after a lower bound computation with respect to the total number of distance computations. As mentioned before, the total number of distance computations are 1.41 trillion for intra-recording pattern discovery and 12.42 trillion for inter-recording pattern detection. From Table 5.4 we see that the DTW computation is avoided in 76% and 99% of the distance computations done in the intra-recording pattern discovery

Lower bound	Intra-rec.(%)	Inter-rec.(%)
LB_KIM_FL	52	45
LB_Keogh_EQ	23	51
LB_Keogh_EC	1	3

Table 5.4: Percentage of exits after a lower bound computation with respect to the total number of distance computations.

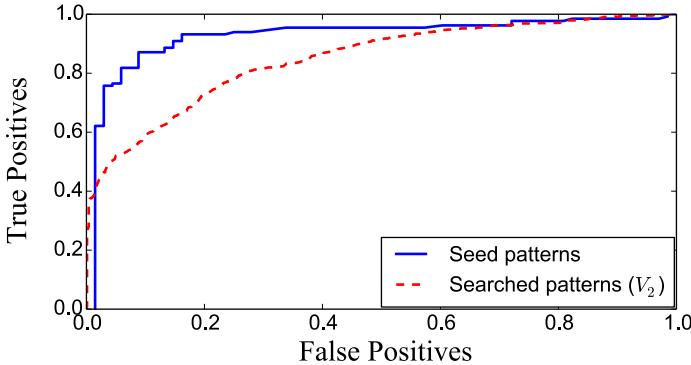


Figure 5.19: ROC curve for seed pairs and search patterns (using V_2) in the evaluation set.

and the inter-recording pattern detection task, respectively. It is evident that the lower bounding methods are more effective in the latter case compared to the former. This is expected as different songs may correspond to different *rāgas* and hence use different set of musical notes, which eventually produces tighter lower bounds. An interesting observation here is that **LB_KIM_FL** lower bound whose computational complexity is $\mathcal{O}(1)$ prunes nearly 50% of the total numbers of possible subsequence pairs.

We now present our formal evaluations. We first evaluate the performance of the intra-recording pattern discovery task. We find that the fraction of melodically similar seed pairs within each seed category S_1 , S_2 and S_3 consistently decreases: 0.98, 0.67 and 0.31, respectively. This is expected as the seed categories are created based on the distance (Δ) between the seed pairs. These numbers indicate that the computed distance strongly correlates to the melodic similarity between the patterns. However, from these numbers we do not get much information about the amount of separation between the distance distributions of melodically similar and dissimilar seed pattern pairs. To examine the separation, we compute the ROC curve as shown in Figure 5.19 (solid blue line). The knee of the curve corresponds to a precision of approximately 80% for 10% of false positive cases. This indicates that the chosen DTW-based distance measure is a sufficiently good candidate for computing melodic similarity for the case of intra-recording seed pattern discovery.

Seed Category	V_1	V_2	V_3	V_4
S_1	0.92	0.92	0.91	0.89
S_2	0.68	0.73	0.73	0.66
S_3	0.35	0.34	0.35	0.35

Table 5.5: MAP scores for four variants of rank refinement method (V_i) for each seed category (S_1 , S_2 and S_3).

Next, we evaluate the performance of inter-recording pattern detection task and assess the effect of the four DTW cost variants explained in Section 5.4.3 (denoted by $V_1 \dots V_4$). To investigate the dependence of the performance on the category of the seed pair, we perform the evaluation within each seed category. In Table 5.5 we show the MAP scores obtained for the different variants of the rank refinement method (V_1, V_2, V_3 and V_4) for each seed category (S_1, S_2 and S_3). In addition, we also present a box plot of corresponding average precision values in Figure 5.20. In general, we observe that every method performs well for category S_1 , with a MAP score around 0.9 and no statistically significant difference between each other. For category S_2 , V_2 and V_3 perform better than the rest and the difference is found to be statistically significant. The performance is poor for the third category S_3 for every variant. The difference in performance between any two methods across seed categories is statistically significant. We observe that MAP scores across different seed categories correlate strongly with the fraction of melodically similar seed pairs in that category (discussed above). This means that the seed pattern pairs that have high distance Δ between them obtain low average precision when used as query patterns and vice-versa. This might be because across repetitions of such patterns there could be a high degree of melodic variation, to model which the current similarity measure appears inadequate. In addition, closely repeating seed pattern pairs (i.e. with low distance Δ between them) might have more number of repetitions with low degree of melodic variations, for which the current similarity measure performs well.

Finally, we analyse the distance distributions of melodically similar and dissimilar search patterns. For this we use the best performing rank-refinement variant V_2 . To examine the separation in the distance distribution, we compute the ROC curve as shown in 5.19 (dashed red line). We observe that the separability between melodically similar and dissimilar subsequences in this case is poorer than the one obtained for the seed pairs (solid blue line). This suggests that it is much harder to differentiate melodically similar from dissimilar patterns when the search is performed across recordings. This can be attributed to the total number of melodically dissimilar subsequences (irrelevant documents) for every query subsequence, which is orders of magnitude higher in this task compared to the task of pattern discovery within a recording. In addition, one of the reasons can also be that the melodic phrases of two allied rāgas (Section 2.3.2) are differentiated based on subtle melodic nuances (Viswanathan &

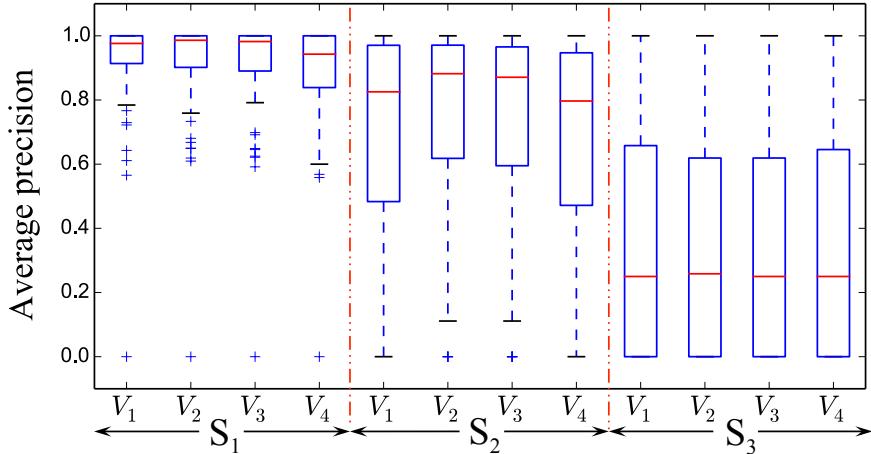


Figure 5.20: Boxplot of average precision values for the different variants of the rank refinement method (V_i) for each seed category.

Allen, 2004). Hence, one faces a much more difficult task, which requires a superior melodic similarity measure.

To demonstrate the capabilities of the approach, we show a few examples of the discovered melodic patterns in Figure 5.21. Our approach robustly extracts patterns in different scenarios such as large local time warpings (b), uniform scaling (c), patterns with silence regions (d) and across different tonic pitches (e and f). It is worth mentioning that, during the process of annotation, the musician found several musically interesting results. For example, striking similarity between phrases of two different *rāgas*, between phrases in sung melodies and the melodies played on instruments (Violin or *Vīṇa*), and phrases sung by different artists. Many of the discovered patterns are the characteristic melodic phrases of the *rāgas*, which are the primary cues for *rāga* recognition. All the discovered melodic patterns in this study can be browsed and listened to through a web interface (Appendix B). Overall, we find that the obtained results are musically relevant and can be used to establish meaningful relationships between audio recordings. We explore the usability of these discovered patterns in the task of automatic *rāga* recognition in Chapter 6.

5.5 Characterization of Melodic Patterns

In the previous section we described our approach to discover melodic patterns in sizable audio music collections of **IAM**. Our primary goal was to extract as many different types of repeating melodic patterns and as many occurrences of them as possible, irrespective of their musical relevance⁵⁰. A well known problem with the compu-

⁵⁰Note that in our method in Section 5.4, we only removed melodically trivial patterns that comprise primarily single svara.

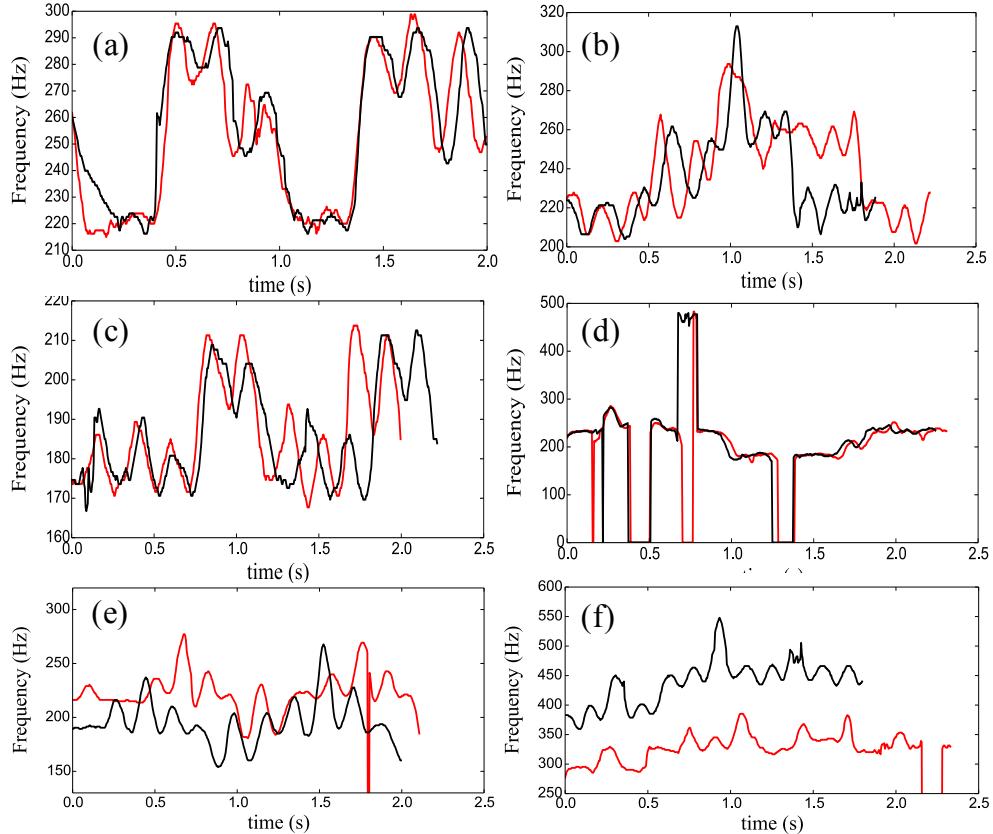


Figure 5.21: Examples of the discovered melodic patterns.

tational approaches for musical pattern discovery is the large volume of discovered patterns, wherein a large fraction of them often tends to be musically uninteresting and irrelevant (Section 2.5.2). This aspect of automated methods for motivic analysis is studied by Marsden (2012a), who says:

...computational approaches find many more motives and many more relationships between fragments than in traditional motivic analysis...these are then filtered by some mechanism which selects motives and relationships with privileged positions within the network of relationships.

Thus, one of the main challenges of pattern discovery methods is to identify the musically meaningful patterns in the output. In the case of melodies in IAM as well recurring patterns differ drastically in terms of their musical relevance (Section 2.3.2). Their functional role in melodies varies from being a melodic ornamentation to being a characteristic melodic phrase of a *rāga*. Thus, in order to effectively utilize the discovered melodic patterns in different melodic analyses and applications in IAM,

characterization of these patterns in terms of their musical relevance and functional roles is crucial. In this section, we address this issue and characterize discovered melodic patterns in order to identify one of the most interesting types of melodic patterns in **IAM**, the *rāga* motifs.

From the literature we see that most of the motivic analysis approaches employ a filtering step to select the musically significant patterns (Section 2.5.2.1). A common approach is to prefer long duration patterns (Cambouropoulos, 2006; Karydis et al., 2006). Another type of approaches prefer patterns that occur more frequently than others (Cambouropoulos, 2006; Meredith et al., 2002). The methods proposed by Conklin (2010); Conklin & Anagnostopoulou (2011) are also based on patterns' frequency of occurrence, but inversely weighted by its frequency of occurrence in an anti-corpus. Collins et al. (2011) evaluate several strategies for assigning importance to discovered melodic patterns. A more detailed description of these filtering strategies is provided in Section 2.5.2.1. We note that most of these filtering strategies are not directly applicable in our context. For example, since our discovery method operates with a limitation of a fixed length melodic pattern, the strategy of maximal length patterns is not applicable. The other type of commonly used strategy in which the most frequently occurring patterns are selected also appears futile. This is because in **IAM**, gamakas and several other melodic ornaments are often the most frequently occurring patterns, which in our task are relatively less musically relevant compared to the *rāga* motifs. Thus, there is a need for a novel filtering strategy that can characterize discovered melodic patterns in **IAM** and can identify the musically interesting ones such as *rāga* motifs.

There is another challenge in following an in-exact or approximate melodic pattern matching methodology, which is that of determining a meaningful similarity threshold. It becomes even a bigger challenge in an unsupervised analysis, such as the case in our pattern discovery method. The discovery approach described in Section 5.4 works with a fixed number of closest pattern matches irrespective of the absolute value of the melodic similarity. As a result of which the output of the pattern discovery approach may contain a large amount of music irrelevant or noisy matches. Although, as seen in Section 5.4.3, the separation between the distance distributions of melodically similar and dissimilar patterns appears favorable (Figure 5.19), indicating that an optimal melodic similarity threshold can potentially remove the noisy matches with a reasonable accuracy. However, determining such a melodic similarity threshold is non-trivial.

In this section, we address both the issues, determining a meaningful melodic similarity threshold, and characterizing the discovered melodic patterns by performing a network analysis. We exploit the topological properties of the network to determine a similarity threshold. For characterizing patterns we first detect non-overlapping communities in the network and then characterize the communities by utilizing the related editorial metadata. As described in Section 2.3.2, repeating patterns in melodies of **IAM** can belong to varied categories. In order to reduce the complexity involved

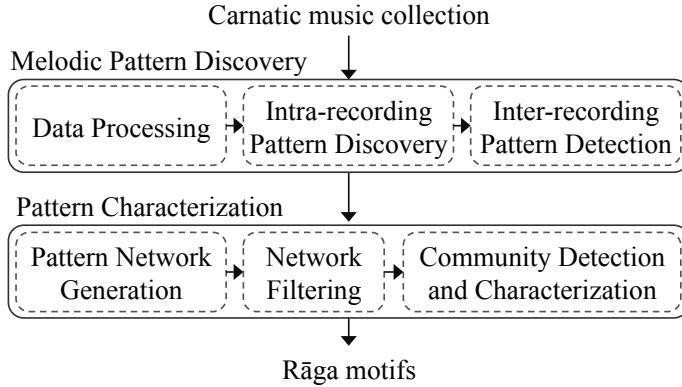


Figure 5.22: Block diagram of the proposed approach for characterizing melodic patterns.

in identification and evaluation of different types of melodic patterns, we focus on *rāga* motifs, which is arguably the most advantageous pattern category for computational melodic analyses of **IAM**. *Rāga* motifs are learned explicitly through years of musical training, and they provide a base for artists' to improvise. Due to these reasons *rāga* motifs are distinctly recognized by performing musicians. In addition, such melodic patterns are crucial for *rāga* based music retrieval systems, automatic *rāga* recognition, studying similarities between artists and recordings, and in developing pedagogical tools for **IAM**. Due to these factors we selected the *rāga* motif category of patterns to study the task of characterization of melodic patterns. In summary, our objective is to discover *rāga* motifs in audio collections of **IAM**.

5.5.1 Method

The block diagram of the proposed approach is shown in Figure 5.22. There are two main processing blocks: (a) melodic pattern discovery and (b) pattern characterization. Both these blocks are described at length in the subsequent sections.

5.5.1.1 Melodic Pattern Discovery

As mentioned, discovering melodic patterns in a sizable audio collection of **IAM** is a challenging task. To get a reliable input for our approach, we employ the pattern discovery approach we describe in Section 5.4. This is one of the few unsupervised systems, we are aware of, that can discover meaningful melodic patterns in large-scale collections of **IAM**. Recall that the output of our pattern discovery approach comprises seed patterns, which are discovered within the recordings, and also, their nearest neighbors in the entire audio collection. In this study, we extract the top 25 closest seed pattern pairs from every recording and consider the top 20 closest patterns for each seed pattern across the recordings. We use the same parameter settings and the implementation of the method as described in Section 5.4.

Notice that the output of our melodic pattern discovery system may contain a high degree of redundancy in terms of overlapping patterns. This is despite the constraints to only select mutually non-overlapping patterns during the discovery and search phase. This redundancy arises primarily because we perform the inter-recording pattern search for each pattern in the seed pair. Since seed pair patterns are (often) close repetitions of each other, we end up retrieving nearly the same set of patterns as nearest neighbors for both of them. Thus, there exists a large number of overlapping patterns in the output of our pattern discovery block. Despite the known redundancy, we chose to perform inter-recording pattern search for each pattern in the pair separately as it exploits intra-class variability present in the patterns, which might provide better retrieval results.

We reduce the above mentioned redundancy in the output of pattern discovery by following a simple procedure. For every discovered melodic pattern we search for its closest pattern across all the recordings using the same similarity measure as used in the intra-recording pattern discovery block. For each recording we parse a list of patterns in that recording sorted in the increasing order of their distances from their closest patterns. While parsing the list we remove every pattern for which there exists an overlap with another pattern placed higher (lower index) in the sorted list. Thus, at the end of this process we retain only non-overlapping patterns in every recording.

5.5.1.2 Melodic Pattern Characterization

Before we formally describe this processing step, we provide the underlying intuition behind it. As explained earlier, a repeating melodic pattern in **IAM** can correspond to either a *rāga* motif, a composition-specific motif or to a gamaka pattern (Section 2.3.2). Our objective is to characterize the discovered patterns in order to identify the ones that are *rāga* motifs. The information available to accomplish this task is: a bunch of melodic patterns, pitch sequences corresponding to the patterns, editorial metadata of the recordings to which the patterns belong and the location of the patterns in the recordings. If we take a single melodic pattern, the only possible indicator of its category could be the characteristics of its pitch sequence. However, to the best of our knowledge (acquired from published studies and discussions with musicians), *rāga* motifs do not possess any distinctive pitch characteristics. Therefore, characterization of the discovered melodic patterns considering one pattern at a time under unsupervised scenario appears to be unfeasible. Such melodic motifs are explicitly learned for each *rāga* though years of musical training. Despite the importance, a comprehensive published collection of *rāga* motifs (audio or pitch sequence) is unavailable.

Instead of analyzing melodic patterns individually, if we analyse them in clusters, where a cluster comprises melodically similar melodic patterns, we can infer to an extent the categories of these melodic patterns. Consider that we have a cluster of melodic patterns and each pattern comes from a different recording in different *rāga*.

It is highly likely that the patterns in the cluster belong to the **gamaka** category, since those patterns occur across **rāgas** and across recordings. On the other hand, if patterns within a cluster belongs to only one **rāga** and different recordings, it is highly likely that the patterns correspond to a **rāga** motif. Thus, by analyzing the properties of a cluster in terms of its relation with different musical attributes and editorial metadata, we can characterize the patterns as belonging to **rāga** motifs or not. Notice that, we eventually exploit the functional roles of different kinds of patterns in melodies of **IAM** in order to identify them in a pool of discovered patterns.

We described above our intuition behind analyzing melodic patterns in units of clusters in order to characterize them. However, clustering melodic patterns further involves many challenges. During clustering we seek to group melodic patterns such that a cluster contains different occurrences of only one melodic phrase. To achieve this, our system should be able to differentiate between melodically similar and dissimilar patterns, for which we need a meaningful similarity threshold. Recall that the extraction of melodic patterns from audio recordings does not involve any similarity thresholding (Section 5.4), we select the top 25 and 20 nearest neighbors in discovery and search phase. Determining a musically meaningful similarity threshold in an unsupervised setup is a challenging task, which we address using the concepts of complex networks.

We now formally describe the processing block. As explained, in this block, we aim to first cluster the discovered patterns and then characterize the clusters in order to identify the ones that represent different **rāga** motifs. For this, we perform a network analysis in which nodes represent the discovered melodic patterns and edges represent the melodic similarities between these patterns. We described below the processes involved in this task (Figure 5.22).

Pattern Network Generation We start by building an undirected weighted network using the discovered melodic patterns from the previous step. The patterns are considered as the nodes of the network and the edge between any two patterns (i, j) is weighted based on the distance Δ_{ij} between the patterns. Noticeably, Δ_{ij} is computed using the same distance measure as used in the intra-recording pattern discovery block in Section 5.4.1.3. The weight of the edge \mathbf{w}_{ij} between the nodes i and j is given by (5.5).

$$\mathbf{w}_{ij} = e^{-\Delta_{ij}/\bar{\Delta}}, \quad (5.5)$$

where, $\bar{\Delta}$ is the mean of Δ_{ij} over every combination of i and j .

Network Filtering The main objective of this processing block is to filter the network in order to retain only the musically meaningful connections between the nodes. Since the edge weights between the pairs of melodically similar and dissimilar nodes may vary by orders of magnitude, we first consider to exploit this heterogeneity to extract the network's backbone. We therefore apply disparity filtering (Serrano et al.,

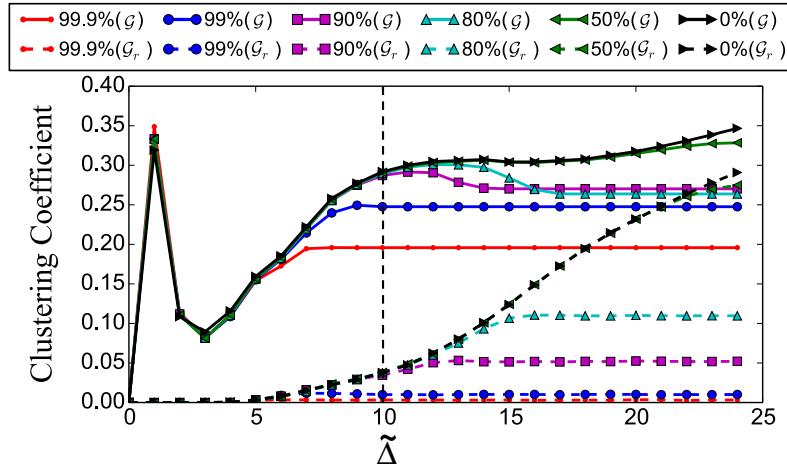


Figure 5.23: Evolution of the clustering coefficient of \mathcal{G} and \mathcal{G}_r over different thresholds and for different statistical confidence values used for disparity filtering (see legend).

2009) to preserve only the edges that represent statistically significant deviations with respect to a null model of edge weight assignment for every node. The only parameter used in the disparity filtering is the statistical confidence value. We iterate over 5 different confidence values {99.99, 99, 90, 80, 50}. However, as we will show, the application of disparity filtering is found to be quite irrelevant for the present case.

We next proceed to filter edges in the network based on a melodic similarity threshold $\tilde{\Delta}$. As mentioned, determining a meaningful similarity threshold in an unsupervised setup is a challenging task. We propose to estimate $\tilde{\Delta}$ based on the topological properties of the network. For this, we analyse the evolution of the clustering coefficient of both the obtained network \mathcal{G} and the corresponding randomized network \mathcal{G}_r over a range of similarity thresholds. Clustering coefficient measures the extent to which the nodes in a network tend to cluster together (Newman, 2003). The randomized network \mathcal{G}_r is obtained by swapping the edges between randomly selected node pairs such that the degree of each node is preserved (Maslov & Sneppen, 2002). This way, \mathcal{G}_r can be considered as the maximally random network with that particular degree distribution. In Figure 5.23, we show the evolution of the clustering coefficient of \mathcal{G} and \mathcal{G}_r over different similarity thresholds (indicated by exponentially spaced bins). In addition, we can also see the clustering coefficient curves for different statistical confidence values used for disparity filtering. The evolution of the clustering coefficients is used for obtaining a similarity threshold as explained below.

We hypothesize that the more musically meaningful $\tilde{\Delta}$ is, the higher is the difference between the clustering coefficients of \mathcal{G} and \mathcal{G}_r . We therefore select $\tilde{\Delta} = 10$. Note that even though the similarity threshold corresponding to $\tilde{\Delta} = 1$ results in a higher value of the clustering coefficient, we reject it because the filtered network consists only of

a small number of nodes. These nodes correspond to near-exact pattern repetitions discovered within the same recording. Such patterns typically represent composition-specific motifs, and hence are irrelevant in our context. In Figure 5.23, we also observe that the disparity filtering using a confidence value higher than 80% significantly lowers the clustering coefficient, which can be attributed to the removal of musically meaningful edges in the network. On the other hand, given $\tilde{\Delta} = 10$, the disparity filtering with a confidence value lower than 80% does not significantly affect the clustering coefficient. We can thus conclude that, in the given scenario, disparity filtering does not bring in any clear advantage. Finally, after applying $\tilde{\Delta}$, we transform \mathcal{G} to an unweighted network.

Community Detection and Characterization We next take the unweighted undirected network that results from the previous step, and perform a non-overlapping community detection using the method proposed in Blondel et al. (2008). This method is based on modularity optimization and is parameter-free from the point of view of the user. It has been extensively used in various applications (Fortunato, 2010) and can deal with very large networks (Blondel et al., 2008). We use the implementation available in networkX (Hagberg et al., 2008), a Python language package for exploration and analysis of networks and network algorithms. Using this method for our entire dataset, we obtain around 1800 communities of melodic patterns.

To get a better understanding of the network obtained after filtering and the detected communities, we show a graphical representation of the network in Figure 5.24. We find that many communities comprising large number of nodes in the network correspond to **kampitam** melodic pattern, which is a kind of **gamaka** that occur in several **rāgas** across compositions and hence have large number of occurrences. We also find that the communities with a relatively smaller number of nodes, which in Figure 5.24 appear as isolated communities around the periphery often correspond to the **rāga** motifs. A few examples of such communities are shown in Figure 5.24 on the right.

We here define the notation used in the subsequent paragraphs for characterizing pattern communities. A community \mathcal{C}_q is comprised of N nodes, and the node count over different **rāgas** is given by the ordered list $\mathcal{A}_q = (\mathcal{A}_{q,1}, \mathcal{A}_{q,2}, \dots, \mathcal{A}_{q,L_a})$ such that $\mathcal{A}_{q,i} \geq \mathcal{A}_{q,j}, \forall i < j$, where each element in \mathcal{A}_q denotes the number of nodes in a particular **rāga** and L_a is the total number of unique **rāgas** comprising the community. Similarly, the node count over the audio recordings is given by the ordered list $\mathcal{B}_q = (\mathcal{B}_{q,1}, \mathcal{B}_{q,2}, \dots, \mathcal{B}_{q,L_b})$ such that $\mathcal{B}_{q,l} \geq \mathcal{B}_{q,m}, \forall l < m$, where each element in \mathcal{B}_q denotes the number of nodes belonging a particular audio recording and L_b is the total number of recordings comprising the community. For both these cases, $\sum_{i=1}^{L_a} \mathcal{A}_{q,i} = \sum_{l=1}^{L_b} \mathcal{B}_{q,l} = N$.

We now proceed to characterize the detected communities in order to identify the ones that represent **rāga** motifs. For that we first categorize a community C_q as belonging to the **rāga** a'_q corresponding to the maximum number of nodes $\mathcal{A}_{q,1}$ in that community. Subsequently, for each **rāga**, we rank all the communities belonging to that **rāga**. To

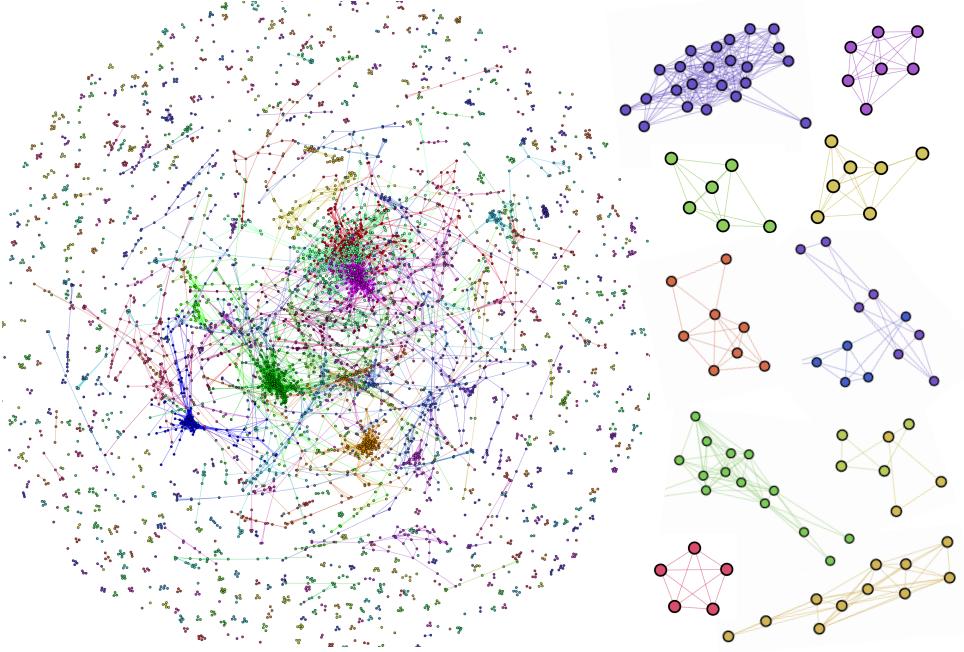


Figure 5.24: Graphical representation of the melodic pattern network after filtering by threshold $\tilde{\Delta}$. The detected communities in the network are indicated by different colors. Few examples of these communities are shown on the right.

rank the communities we empirically devise a goodness measure \mathbf{G} , which denotes the likelihood that a community C_q represents a *rāga* motif. We propose to use

$$\mathbf{G} = N \mathcal{L}^4 \mathbf{c}, \quad (5.6)$$

where \mathcal{L} is an estimate of the likelihood of *rāga* a'_q in C_q ,

$$\mathcal{L} = \frac{\mathcal{A}_{q,1}}{N}, \quad (5.7)$$

and \mathbf{c} indicates how uniformly the nodes of the community are distributed over audio recordings,

$$\mathbf{c} = \frac{\sum_{l=1}^{L_b} l \cdot \mathcal{B}_{q,l}}{N}. \quad (5.8)$$

Higher \mathbf{c} implies a more uniform distribution. Since a community that represents a *rāga* motif is expected to contain nodes from a single *rāga* (high value of \mathcal{L}) and the nodes belong to many different recordings (high value of \mathbf{c}), the goodness measure \mathbf{G} is high for such a community. In general we prefer large communities, but, to avoid

detecting large communities (high value of N) corresponding to gamaka motifs (low value of \mathcal{L}) we use a fourth power on \mathcal{L} . Composition-specific motifs are expected to have a low c , as they are not repeated across multiple recordings, assuming that different recordings correspond to different compositions.

Note that it might be possible that a music collection contains multiple recordings of the same composition. In such cases, differentiating composition-specific motifs and *rāga* motif becomes difficult using **G** measure. This issue can be overcome with only a trivial modification in our approach, i.e., by considering the node distribution \mathcal{B}_q over compositions instead of recordings. In our current system this could not be implemented, because for a number of recordings their composition information was unavailable.

5.5.2 Evaluation

5.5.2.1 Music Collection

The music collection used in this study is a subset of the CompMusic Carnatic music corpus (Section 3.2.2). The collection comprises 44 hours of polyphonic audio music recordings of Carnatic music across 10 different *rāgas*. For each *rāga* we select 16 music pieces, which amounts to a total of 160 recordings. There are 139 vocal music recordings and 21 instrumental recordings comprising violin, *vīṇa* and bamboo flute. In Table 5.6, we summarize the relevant details of the dataset. We see that it is diverse in terms of the number of unique compositions and number of lead artists. Furthermore, it includes different forms of compositions (*kīrtana*, varnam and viruttam) and recordings containing varied improvised sections such as *ālāpna*, nereval and *kalpanā-svaras*. The chosen *rāgas* contain diverse set of *svaras* (notes) both in terms of the number of *svaras* and their pitch classes (svarasthānās). From Table 5.6, we also notice that several *rāgas* such as *Kalyāṇi*, *Kāmbhōji* and *Bēgadā* have a large fraction of *svaras* in common. We refer to them as allied *rāgas* (Section 2.3.2). This further increases the complexity of the task at hand, since the discrimination between the phrases of allied *rāgas* may be based on subtle melodic nuances.

5.5.2.2 Setup and Evaluation Measures

Given the unsupervised nature of this study, we perform a listening test to formally evaluate the extent to which the selected melodic patterns correspond to *rāga* motifs. For each of the 10 *rāgas* in the dataset, we select the top 10 communities based on the goodness measure **G** (Eq. 5.6). From each of these communities, we select their representative melodic pattern based on the betweenness centrality of the nodes (Newman, 2003), i.e., the node with the highest betweenness centrality is considered as the representative melodic pattern of that community. In case of a tie, we select the one with the highest node degree (Newman, 2003). Finally, we arrive at a set of 100 melodic patterns, which are then used to perform the listening test. These audio ex-

Rāga	Dur	#Com	#Art
Hamsadhvāni	2.46	12	14
Kāmavardani	3.94	13	16
Darbār	2.59	8	13
Kalyāni	6.94	9	16
Kāmbhōji	6.91	12	13
Bēgada	3.41	9	16
Kāpi	2.24	12	16
Bhairavi	5.33	7	16
Behāg	1.51	12	16
Tōdī	8.75	12	16
Total	44.08	106	57

Table 5.6: Details of the dataset in terms of the duration (Dur) in hours, number of unique compositions (#Com) and unique lead artists (#Art) for each rāga. Svaras constituting these rāgas are listed in Table C.4.

amples are also made available online (see the companion page of the corresponding study in Appendix A).

For the listening test we select 10 professional Carnatic musicians with over 15 years of formal music training. Each musician is presented with the audio fragments corresponding to the selected melodic patterns in a random order. They are also presented with the rāgas corresponding to the melodic patterns. The musicians are asked to rate each melodic pattern based on whether it is a characteristic phrase of that rāga. We use binary ratings ('Yes' or 'No').

The audio fragments were segmented with a one second buffer on either side of the pattern to offer some context and reduce the effect of abrupt boundaries. In order to quantify the musicians' assessment, we use mean ratings for each pattern ϕ , μ_ϕ , considering 'Yes' as 1 and 'No' as 0. For analyzing the ratings per rāga, we study the mean and standard deviation of all μ_ϕ for patterns in every rāga, which we denote by μ_r and σ_r , respectively.

5.5.3 Results and Discussion

We first analyse the musicians' ratings at the level of melodic patterns. In Figure 5.25, we show μ_ϕ for the 100 selected melodic patterns, where the grouping is based on their corresponding rāgas. We find that the mean and the standard deviation of μ_ϕ

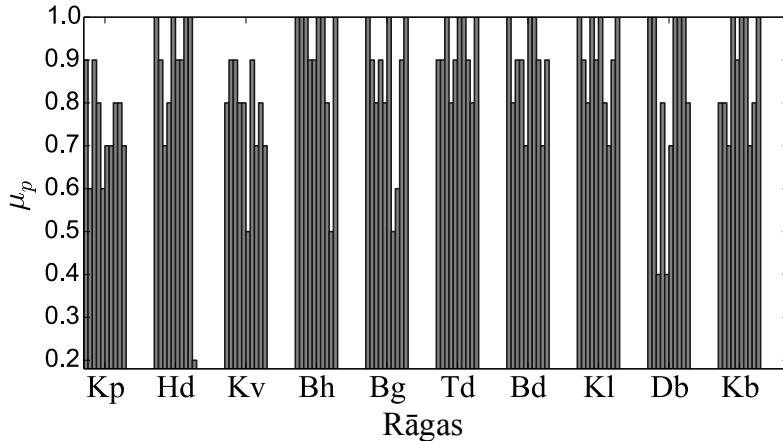


Figure 5.25: Mean musician rating per melodic pattern for each rāga: Kāpi (Kp), Hamsadhvāni (Hd), Kāmavardani (Kv), Bhairavi (Bh), Behāg (Bg), Tōdī (Td), Bēgaḍa (Bd), Kalyāni (Kl), Darbār (Db), Kāmbhōji (Kb).

for the melodic patterns is 0.85 and 0.16, respectively. For a better understanding of μ_ϕ across patterns and the overall musicians' agreement, we show the histogram of μ_ϕ in Figure 5.26. We see that 33 melodic patterns are rated as rāga motifs by all 10 musicians and 25 patterns are rated as rāga motifs by 9 out of 10 musicians. Similarly, the musicians' agreement can be inferred for the rest of the patterns from this histogram. We observe that 91% of the patterns are always marked as rāga motifs by at least 7 out of 10 musicians.

We now proceed to analyse the results for different rāgas. In Table 5.7, we summarize mean μ_r and standard deviation σ_r of μ_ϕ for each rāga. We observe that there is a considerable amount of variation in μ_r across the rāgas. It ranges from 0.75 for rāga Kāpi to 0.92 in the case of rāga Tōdī. An interesting observation here is that the phrase-based rāgas⁵¹ are the top performing rāgas with the exception of rāga Darbār. From Table 5.7 and Table 5.6, we notice a strong correlation between μ_r and the total duration of the audio recordings across rāgas. This suggests that longer music pieces are likely to facilitate the discovery of rāga motifs owing to more number of occurrences of such melodic patterns.

We examine the melodic patterns with low scores. An investigation of 9 out of 100 patterns that obtain $\mu_\phi \leq 0.6$ reveals that many of these patterns correspond to the composition-specific phrases that do not characterize the rāga. The method wrongly identifies them as rāga motifs because their associated communities have a high **G** score owing to a high **c** value. This can be attributed to the fact that these patterns are discovered from multiple recordings, since their corresponding compositions have

⁵¹Rāgas whose identity is primarily derived based on phraseology than the svaras (Krishna & Ishwar, 2012)

Rāga	μ_r	σ_r	Rāga	μ_r	σ_r
Hamsadhvāni	0.84	0.23	Bēgada	0.88	0.11
Kāmavardani	0.78	0.17	Kāpi	0.75	0.10
Darbār	0.81	0.23	Bhairavi	0.91	0.15
Kalyāṇi	0.90	0.10	Behāg	0.84	0.16
Kāmbhōji	0.87	0.12	Tōḍī	0.92	0.07

Table 5.7: Mean (μ_r) and standard deviation (σ_r) of μ_ϕ for each rāga. Rāgas with $\mu_r \geq 0.85$ are highlighted.

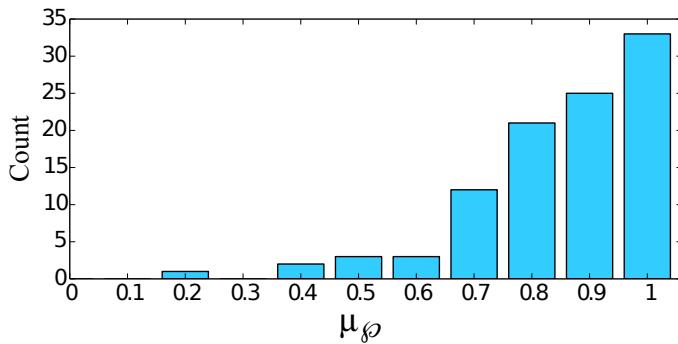


Figure 5.26: Histogram of μ_ϕ for all of the 100 melodic patterns selected for the listening test.

several recordings in the dataset. As described in Section 5.5.1.2, in such a scenario, the goodness measure \mathbf{G} can be made more robust to such cases by computing \mathbf{c} using the distribution of nodes \mathcal{B}_q over unique compositions rather than over audio recordings.

The results show that the proposed method successfully discovers rāga motifs with a high accuracy. We see that, even for the allied rāgas present in the dataset such as Kāmbhōji and Bēgada (Section 5.5.2.1), the method is able to discover distinct characteristic rāga motifs. As mentioned, allied rāgas are challenging because they have a substantial overlap in the set of svaras that they comprise (see also Table 5.6). Finally, on a more informal side, it is worth mentioning that musicians were impressed when, after the listening test, they came to know that the melodic patterns were discovered by a machine following an unsupervised approach.

We consider this work as a preliminary study with a lot of scope for improvements. In particular, the approach can be extended to identify patterns belonging to other categories (gamaka or composition-specific patterns), definition of the goodness measure \mathbf{G} can be improved as suggested in Section 5.5.3, listening test could be done more

rigorously by including negative examples (gamaka patterns), and the resultant patterns of the approach can be quantitatively evaluated by using them in tasks such as composition identification. On these lines, in Section 6.2 we present an approach that uses clustered melodic patterns to perform the task of automatic *rāga* recognition. That way, in addition to the qualitative assessment as done in this work, we can also quantitatively assess the musical relevance of the extracted and characterized melodic patterns.

5.6 Summary and Conclusions

In this chapter, we presented our methodology for discovering musically significant melodic patterns in sizable audio music collections of **IAM**. Our methods utilize the melodic representations and descriptors obtained in Chapter 4, and combine concepts from music signal processing, time-series analysis, information retrieval, and complex networks to successfully extract musically meaningful melodic patterns in audio collections. We studied three main tasks involved in this process: melodic similarity, pattern discovery, and characterization of the discovered melodic patterns.

We first carried out an in-depth supervised analysis of melodic similarity, which is a crucial component in pattern discovery. We evaluated 560 different combinations of various computational procedures and parameters values that are often used for this task. We showed that melodic similarity computation is very sensitive to the choice of parameters and processing steps. A higher sampling rate of the melody representation and mean normalization is desired for computing a reliable melodic similarity in Carnatic music. For Hindustani music, on the other hand, sampling rate (within the considered range) has no significant affect on the performance and tonic normalization of the melody results in a better performance. In general, DTW-based distance measure performs better than the Euclidean distance, and the usage of local constraint in DTW enhances the performance. The DTW variant without any global constraint is preferred (specially for Hindustani music), which suggests that there are large non-linear timing variations across occurrences of the melodic pattern. We observed that an accurate segmentation of the melodic patterns has a huge positive impact on the computation of melodic similarity. The best methodology variant thus identified for melodic similarity is further improved by addressing two specific challenges that arise due to large non-linear timing variations and rapid melodic movements in melodic patterns. The solution we proposed exploits specific melodic characteristics that are particular to **IAM**. We showed that duration truncation of the steady *svara* regions in melodic phrases results in a statistically significant improvement in the computation of melodic similarity. Furthermore, we showed that the complexity of a melodic pattern in Carnatic music is a distinguishing aspect of the pattern, and can be successfully utilized to improve melodic similarity.

Subsequently, we presented our data-driven unsupervised approach for discovering repeated melodic patterns in large audio collections of **IAM**. We first discovered seed

patterns within a recording, and later, used those as queries to detect similar occurrences in the entire music collection. We used DTW-based distance measures to compute melodic similarity and compared four different rank refinement variants. Discovering 25 closest seed pattern pairs within each recording and retrieving their 200 closest patterns in an audio collection of 365 hours of Carnatic music comprising 1764 recordings results in around 12 trillion distance computations. We showed that using cascaded lower bounding techniques borrowed from time-series analysis we save nearly 76% of the total computations for the intra-recording pattern discovery task and around 99% of the total computations for the inter-recording pattern detection task. Thus, such indexing techniques can scale DTW on hundreds of hours of music collections. We evaluated a randomly sampled subset of the extracted melodic patterns comprising 8000 patterns by performing listening tests by a professional Carnatic musician. Our quantitative evaluation based on the expert feedback indicated that a DTW-based distance measure performs well for intra-recording discovery. However, the performance for the inter-recording pattern detection task is inferior suggesting that we require better melodic similarity measures for searching occurrences across recordings. Qualitative feedback from the musician suggests that the extracted melodic patterns are interesting and contain several instances of musically significant patterns such as *rāga* motifs.

The output of the pattern discovery method contains different types of repeating patterns. We presented an approach to identify musically significant patterns, the *rāga* motifs, and specifically, to distinguish them from *gamaka* and composition-specific patterns (Section 2.3.2). We employed a network analysis and use a non-overlapping community detection algorithm to cluster melodic patterns. Using the topological properties of the network, we determined a musically meaningful similarity threshold. We devised a goodness measure for characterizing the detected communities. In a listening test with 10 professional Carnatic musicians we showed that the proposed method successfully discovers *rāga* motifs with accuracy, even in the presence of allied *rāgas* in the dataset. Thus, we demonstrated that the methodology we use to determine melodic similarity threshold and to define the goodness measure successfully identifies musically significant patterns. Our results show that the functional roles of different melodic phrases in IAM can be effectively exploited to identify them in an unsupervised manner. To the best of our knowledge, utilization of the functional roles of different melodic patterns in IAM in order to describe and characterize them is done for the first time.

Overall, we showed that our unsupervised methodology that does not require any exemplar patterns can successfully discover musically significant melodic patterns (*rāga* motifs) in sizable music collections of IAM. These patterns can be used in a number of MIR tasks and applications for IAM. Due to their importance in characterizing *rāgas* they can immediately be utilized for automatic *rāga* recognition, a topic that we study in Chapter 6. These patterns can be used to interlink large volumes of audio recordings in order to define novel music similarity measures, and to perform musico-

logically relevant studies such as characterization of rāgas, artists, and compositions in IAM. Furthermore, there can be enhanced music listening applications, and pedagogical tools that can potentially utilize these melodic patterns. Concrete examples of such applications are provided in Chapter 7.

Chapter 6

Automatic Rāga Recognition

6.1 Introduction

In this chapter, we address the task of automatically recognizing rāgas in audio recordings of IAM. We describe two novel approaches for rāga recognition that jointly capture the tonal and the temporal aspects of melody. The contents of this chapter are largely based on our published work in Gulati et al. (2016a,b).

Rāga is a core musical concept used in compositions, performances, music organization, and pedagogy of IAM. Even beyond the art music, numerous compositions in Indian folk and film music are also based on rāgas (Ganti, 2013). Rāga is therefore one of the most desired melodic descriptions of a recorded performance of IAM, and an important criterion used by listeners to browse its audio music collections. Despite its significance, there exists a large volume of audio content whose rāga is incorrectly labeled or, simply, unlabeled. A computational approach to rāga recognition will allow us to automatically annotate large collections of audio music recordings. It will enable rāga-based music retrieval in large audio archives, semantically-meaningful music discovery and musicologically-informed navigation. Furthermore, a deeper understanding of the rāga framework from a computational perspective will pave the way for building applications for music pedagogy in IAM.

Rāga recognition is the most studied research topic in MIR of IAM. There exist a considerable number of approaches utilizing different characteristic aspects of rāgas such as svara set, svara salience and ārōhana-avrōhana. A critical in-depth review of the existing approaches for rāga recognition is presented in Section 2.4.3, wherein we identify several shortcomings in these approaches and possible avenues for scientific contribution to take this task to the next level. Here we provide a short summary of this analysis:

- Nearly half of the number of the existing approaches for rāga recognition do not utilize the temporal aspects of melody at all (Table 2.3), which are crucial

in characterizing rāgas. Approaches that do utilize the temporal aspects, invariably consider a discrete representation of melody in the analysis. Thus, they do not capture the characteristics of the continuous melodic transitions across the *svaras*, which may be relevant for rāga recognition. Therefore, approaches that can work with a continuous melody representation and still can capture effectively the temporal aspects of melody are worth exploring for this task.

- The musical complexity in recognizing rāgas depends on both the number and the specific set of rāgas to be distinguished. A large number of the existing approaches for this task are evaluated using different datasets, which typically comprise a small number of rāgas with only a handful of recordings per rāga. Therefore, a reliable assessment and comparison of the existing approaches becomes a challenging task. Thus, a sizable representative dataset of Hindustani and Carnatic music that can be openly shared and used by the community will be instrumental in systematically improving the state of the art in rāga recognition.
- A comparative evaluation of the existing methods is missing in the literature. Most of the studies only evaluate their proposed method and do not perform any comparison with the existing studies. A comprehensive comparison of different approaches on the same dataset and under the same experimental setup is needed in order to identify the strengths and weaknesses of these methods.

In this chapter, we describe two novel methods for rāga recognition M_{VSM} and M_{TDMS} that overcome a number of shortcomings in the existing approaches as enumerated above and in our literature review (Section 2.4.3). In addition, we compile and curate sizable datasets of Hindustani and Carnatic music for evaluating our methods. To the best of our knowledge these are the largest datasets ever used for this task. We also make these datasets publicly available online (Appendix B). We now proceed to describe our methods in detail.

6.2 Pattern-Based Rāga Recognition

In this section, we describe our pattern-based approach (M_{VSM}) to rāga recognition. M_{VSM} utilizes melodic patterns automatically discovered from audio recordings using our approach described in Chapter 5, and employs the vector space modeling concept to build a rāga model from these melodic patterns. This section is based on our published work presented in Gulati et al. (2016b).

As mentioned before, every rāga has a set of characteristic melodic patterns that capture the essence of the rāga. These melodic patterns are one of the most prominent cues for rāga identification, used by a performer, as well as by a listener (Section 2.3.2). They act as a building block to construct melodies both in musical compositions and improvisations. However, despite the importance of melodic patterns in

characterizing rāgas, they have not been fully exploited by the computational methods for rāga recognition. There exist only a handful of methods that utilize melodic patterns for this task (Section 2.4.3). These methods work with a discrete representation of melody and pre-defined dictionaries of melodic patterns, which severely limits the potential of a pattern-based approach for rāga recognition. To the best of our knowledge, there exists only one method that uses automatically discovered melodic patterns for this task (Dutta et al., 2015). However, the melodic patterns are extracted from specific short duration regions (*pallavi* lines) of recordings in Carnatic music, and therefore, the scalability of this method on large audio collections is questionable. Furthermore, the authors address the task of rāga verification, which is less challenging compared to rāga recognition. A detailed discussion on the existing methods and their shortcomings is provided in Section 2.4.3.

Before formally describing our method, we first present the intuition and motivation behind the approach. A number of similarities can be seen between a rāga rendition and a textual description of a topic. Like an author describes a topic by using different words relevant to the topic, an artist renders a rāga by using appropriate melodic phrases that suit the context. There are words that are quite specific to a topic, which are analogous to the characteristic melodic phrases of a rāga. Stop words, which are not specific to any topic or to a document can be seen as generic *gamaka* type melodic patterns, which are not specific to a rāga or to a recording (Section 2.3.2.2). Words that are specific to a document are analogous to composition specific patterns. This analogy drives our method and motivates us to employ concepts of vector space modeling (VSM) to perform rāga recognition using melodic patterns. We now proceed to describe M_{VSM} in detail.

6.2.1 Vector Space Modeling of Melodic Patterns

The block diagram of M_{VSM} is shown in Figure 6.1. There are three main processing blocks: melodic pattern discovery, pattern clustering and term frequency inverse document frequency (TF-IDF)-based feature extraction. We describe each of these blocks in the following sections.

6.2.1.1 Melodic Pattern Discovery

In this block, we extract repeating melodic patterns from the collection of audio recordings. For this, we employ the pattern discovery method described in Section 5.4. There are three main processing modules in our pattern discovery method: data-processing, intra-recording pattern discovery and inter-recording pattern search, which are already described at length in Section 5.4.1.

The results of the experiments presented in this section are provided for the following parameter settings used in the pattern discovery block. These parameter values are set based on our learnings in different experiments described in Chapter 5. We use the predominant pitch sampled at 45 Hz for Hindustani music and 55 Hz for Carnatic

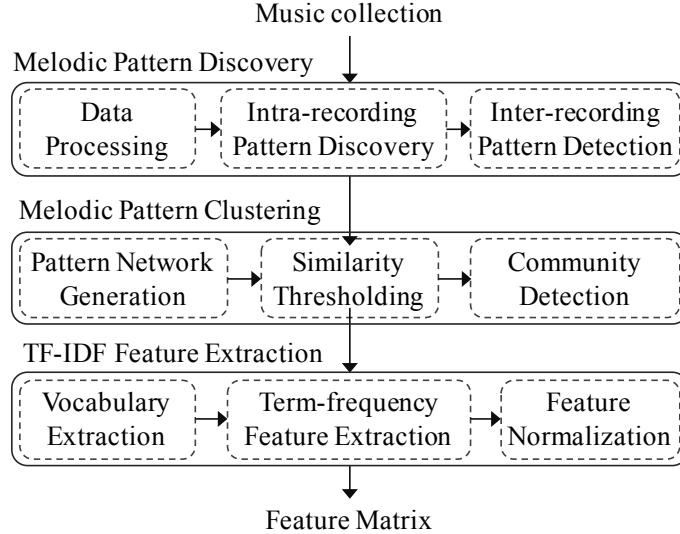


Figure 6.1: Block diagram of the proposed phrase-based approach to rāga recognition.

music. We apply all the post-processing procedures described in Section 4.3.2 with the same set of parameter values. We extract melodic patterns of length 2 s in both the music traditions. In addition to normalizing the predominant pitch by the tonic of the recording, we also perform an octave normalization while comparing patterns (Section 5.3.1.2). We extract 30 closest pattern pairs within each recording, and for each of them their 20 closest patterns across the recordings. We use a global DTW band constraint with 10% bandwidth. The local cost in DTW distance is computed using the squared Euclidean function. For each melodic pattern candidate its five time-scaled versions are considered in both intra and inter-recording pattern search as described in Section 5.2.1.3. We also perform duration truncation of the steady *svaras* in the melodies (Section 5.3.1.4). We truncate *svaras* to a maximum of 500 ms and 300 ms for Hindustani and Carnatic music melodies, respectively. These values are derived from our results presented in Section 5.3.3. More detailed information about the system parameters and configurations can be obtained from the companion web page of the relevant article (Gulati et al., 2016b)⁵².

Note that the output of the pattern discovery method contains different types of repeated melodic patterns with varied degree of their musical relevance (Section 5.5). In this step we do not filter any melodic pattern based on their relevance with respect to rāgas. A soft selection of the relevant patterns is implicitly done in our methodology described in the subsequent sections.

⁵²<http://compmusic.upf.edu/node/278>

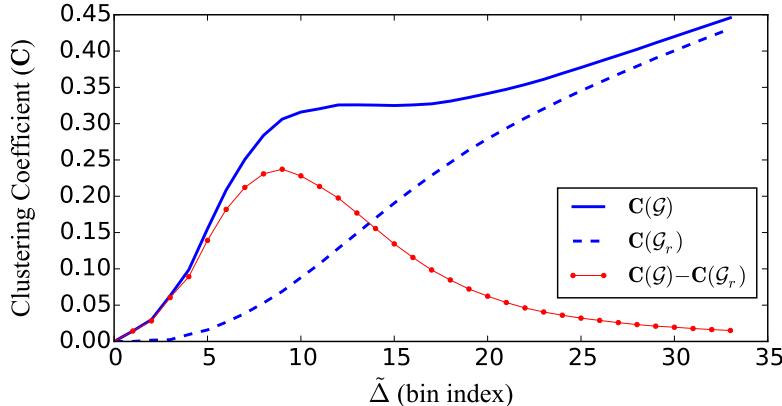


Figure 6.2: Evolution of the clustering coefficients of \mathcal{G} and \mathcal{G}_r , and their difference for different similarity thresholds ($\tilde{\Delta}$).

6.2.1.2 Melodic Pattern Clustering

In order to effectively utilize the discovered melodic patterns for rāga recognition, it is important to cluster together all the patterns that are different occurrences of the same underlying melodic phrase. For this, we propose to perform a network analysis, wherein the clustering is performed using a non-overlapping community detection method. The network analysis and clustering process used here is the same as described in Section 5.5.1.2, but with a different end goal of characterizing melodic patterns. For the sake of completeness we here provide a brief description of the process we follow.

We start by building an undirected network \mathcal{G} using the discovered patterns as the nodes of the network. We connect any two nodes only if the distance between them is below a similarity threshold $\tilde{\Delta}$. Noticeably, the distance between two melodic patterns is computed using the same measure as used in the intra-recording pattern discovery block (Section 5.4.1.3). The weight of the edge, when it exists, is set to 1, and all non-connected nodes are removed from the network.

As discussed in Section 5.5 determining a meaningful similarity threshold in an unsupervised manner is a challenging task. For estimating an optimal value of $\tilde{\Delta}$ we follow the approach as described in Section 5.5.1.2. We compare the evolution of the clustering coefficient C of the obtained network \mathcal{G} with the clustering coefficient of a randomized network \mathcal{G}_r over different distance thresholds $\tilde{\Delta}$. The randomized network \mathcal{G}_r is obtained by swapping the edges between randomly selected pairs of nodes such that the degree of each node is preserved (Maslov & Sneppen, 2002). In Figure 6.2, we show $C(\mathcal{G})$, $C(\mathcal{G}_r)$ and $C(\mathcal{G}) - C(\mathcal{G}_r)$ for different values of $\tilde{\Delta}$. The optimal threshold $\tilde{\Delta}^*$ is taken as the distance that maximizes the difference between the two clustering coefficients.

In the next step of our method we group together similar melodic patterns (Figure 6.1). To do so, we detect non-overlapping communities in the network of melodic patterns using the method proposed by Blondel et al. (2008). This community detection method is based on optimizing the modularity of the network and is parameter-free from the user's point of view. This method is capable of handling very large networks and has been extensively used in various applications (Fortunato, 2010). We use its implementation available in networkX (Hagberg et al., 2008), a Python language package for exploration and analysis of networks and network algorithms. Note that, from now on, the melodic patterns grouped within a community are regarded as the occurrences of a single melodic phrase. Thus, a community essentially represents a melodic phrase or motif.

6.2.1.3 TF-IDF Feature Extraction

As mentioned above, we draw an analogy between *rāga* rendition and textual description of a topic. Using this analogy we represent each audio recording using a vector space model, wherein melodic patterns are considered as words (or terms). This process is divided into three blocks (Figure 6.1).

We start by building our vocabulary \mathcal{V} , which translates to selecting relevant pattern communities for characterizing *rāgas*. For this, we include all the detected communities except the ones that comprise patterns extracted from only a single audio recording. Such communities are analogous to the words that only occur within a document and, hence, are irrelevant for modeling a topic.

We experiment with three different sets of features f_1 , f_2 and f_3 , which are similar to the **TF-IDF** features typically used in text information retrieval. We denote our corpus by \mathcal{R} comprising $N_{\mathcal{R}} = |\mathcal{R}|$ number of recordings. A melodic phrase and a recording is denoted by \wp and r , respectively

$$f_1(\wp, r) = \begin{cases} 1, & \text{if } f(\wp, r) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (6.1)$$

where, $f(\wp, r)$ denotes the raw frequency of occurrence of pattern \wp in recording r . f_1 only considers the presence or absence of a pattern in a recording. In order to investigate if the frequency of occurrence of melodic patterns is relevant for characterizing *rāgas*, we take $f_2(\wp, r) = f(\wp, r)$. As mentioned, the melodic patterns that occur across different *rāgas* and in several recordings do not aid *rāga* recognition. Therefore, to reduce their effect in the feature vector we employ a weighting scheme, similar to the inverse document frequency (idf) weighting in text retrieval.

$$f_3(\wp, r) = f(\wp, r)\mathcal{I}(\wp, \mathcal{R}) \quad (6.2)$$

$$\mathcal{I}(\wp, \mathcal{R}) = \log \left(\frac{N_{\mathcal{R}}}{|\{r \in \mathcal{R} : \wp \in r\}|} \right) \quad (6.3)$$

where, $|\{r \in \mathcal{R} : \phi \in r\}|$ is the number of recordings where the melodic pattern ϕ is present, that is $f(\phi, r) \neq 0$ for these recordings.

6.2.2 Evaluation

6.2.2.1 Music Collection

The music collection used for evaluation in this study is a subset of the CompMusic Carnatic music corpus (Section 3.2.2). Due to the differences in the melodic characteristics of Carnatic and Hindustani music, to evaluate our method we compile and curate two datasets, RRD_{CMD} and RRD_{HMD} , one for each music tradition (Section 3.3.4). A separate evaluation on both the music traditions allows a better analysis and interpretation of the results.

RRD_{CMD} and RRD_{HMD} comprise 124 and 116 hours of commercially available audio recordings, respectively. All the editorial metadata for each audio recording is publicly available in MusicBrainz⁵³, an open-source metadata repository. RRD_{CMD} contains full-length recordings of 480 performances belonging to 40 rāgas with 12 music pieces per rāga. RRD_{HMD} contains full-length recordings of 300 performances belonging to 30 rāgas with 10 music pieces per rāga. The selected music material is diverse in terms of the number of artists, the number of forms, and the number of compositions, and thus, it is representative of these music traditions. The chosen rāgas contain diverse sets of svaras, both in terms of the number of svaras and their pitch-classes (svarasthānās).

To the best of our knowledge, these are the largest and the most comprehensive (in terms of the available metadata) datasets ever used for studying the task of automatic rāga recognition. To facilitate reproducible research and comparative studies we also make these datasets publicly available online (Appendix B). A further detailed description of these datasets is provided in Section 3.3.4.

6.2.2.2 Classification and Experimental Setup

The features obtained above are used to train a classifier. In order to assess the relevance of these features for rāga recognition, we experiment with different algorithms exploiting diverse classification strategies (Hastie et al., 2009a): multinomial naive Bayes (NBM), Gaussian naive Bayes (NBG), and Bernoulli naive Bayes (NBB), support vector machines with a linear and a radial basis function kernel, and with a stochastic gradient descent learning (SVML, SVMR and SGD, respectively), logistic regression (LR) and random forest (RF). We use the implementation of these classifiers available in scikit-learn toolkit (Pedregosa et al., 2011), version 0.15.1. Since in this study, our focus is to extract a musically relevant set of features based on melodic patterns, we use the default parameter settings for the classifiers available in scikit-learn.

⁵³<https://musicbrainz.org/>

We use leave-one-out cross validation methodology for evaluations (Mitchell, 1997), in which one recording in the evaluation dataset forms the testing set and the remaining ones become the training set. We use the mean classification accuracy across recordings as the evaluation measure. To assess if the difference in the performance between any two methods is statistically significant, we use McNemar's test (McNemar, 1947) with $p < 0.01$. In addition, to compensate for multiple comparisons, we apply the Holm-Bonferroni method (Holm, 1979).

Note that in our published work in Gulati et al. (2016b) we use a different evaluation strategy (12-fold cross-validation methodology). Therefore the results provided in the article differ slightly from the ones presented in this thesis. In addition, the method for assessing statistical significance is also different (Mann-Whitney U test), which is due to the difference in the evaluation strategy. The evaluation strategy used here (leave-one-out cross validation) does not involve any random sampling (or split) of the evaluation set, which makes our experimental setup more definite.

6.2.2.3 Comparison with the State of the art

We compare our results with two state of the art methods proposed in (Chordia & Şentürk, 2013) and (Koduri et al., 2014). As an input to these methods, we use the same features, predominant pitch and tonic, as used in our method. The only difference is that the pitch contours fed to these methods are not post-processed (Section 4.3.2). This is because these methods primarily exploit the intonation aspect of the svaras, and therefore, performing a smoothening operation on pitch contours might degrade their performance. The method in Chordia & Şentürk (2013) uses smoothed pitch-class distribution (PCD) as the tonal feature and employs 1-nearest neighbor (1-NN) using Bhattacharyya distance for predicting rāga labels. We denote this method by M_{PC} . The authors in Chordia & Şentürk (2013) report a window size of 120 s as an optimal duration for computing PCDs (denoted here by PCD_{120}). However, in our experiments we find that PCDs computed over the entire audio recording (denoted here by PCD_{full}) result in a significant improvement (Gulati et al., 2016b). We therefore use PCD_{full} for comparison. Note that in Chordia & Şentürk (2013) the authors do not experiment with a window size larger than 120 s.

The method in Koduri et al. (2014) is proposed primarily for Carnatic music. This method also uses features based on pitch distribution. However, unlike in M_{PC} , the authors use parameterized pitch distributions as features. We denote this method by M_{GK} . We consider two variants of this method. One in which the parameterization is done using a single pitch histogram that includes distribution of all the svaras in the recording (denoted here by PD_{param}), and the other, wherein a separate histogram is constructed for each individual svara in the recording (denoted here by $PD_{context}$). In the second variant the mapping of a pitch sample to a svara is based on its local melodic context as explained in Koduri et al. (2014). Note that M_{GK} utilizes specific intonation aspects of svaras in Carnatic music, and is not devised and tested for the

Method	Feature	SVML	SGD	NBM	NBG	RF	LR	1-NN
M_{VSM}	f_1	51.04	55	37.5	54.37	25.41	55.83	-
	f_2	45.83	50.41	35.62	47.5	26.87	51.87	-
	f_3	45.83	51.66	67.29	44.79	23.75	51.87	-
M_{PC}	PCD _{full}	-	-	-	-	-	-	73.12
M_{GK}	PD _{param}	30.41	22.29	27.29	28.12	42.91	30.83	25.62
	PD _{context}	54.16	43.75	5.2	33.12	49.37	54.79	26.25

Table 6.1: Accuracy (%) of rāga recognition on RRD_{CMD} dataset by M_{VSM} and other methods methods using different features and classifiers. Bold text signifies the best accuracy by a method among all its variants

Hindustani music recordings. We therefore do not consider this method for comparing results on Hindustani music.

The authors of M_{PC} courteously ran the experiments on our dataset using the original implementations of the method. For M_{GK} , the authors kindly extracted the features (PD_{param} and PD_{context}) using the original implementation of their method and the experiments using different classification strategies were done by us.

6.2.3 Results and Discussion

Before we proceed to present our results, we notify readers that the accuracies reported in this section for different methods vary slightly from the ones reported in Gulati et al. (2016b). As mentioned, the experimental setup used here is different compared to the paper (Section 6.2.2.2). In addition, the parameters used for discovering melodic patterns are also different (Section 6.2.1.1). Furthermore, we do not consider the dataset that comprise only 10 rāga as done in Gulati et al. (2016b), for which our method was shown to outperform the state of the art.

In Table 6.1 and Table 6.2, we present the results of our proposed method M_{VSM} and the two state of the art methods M_{PC} and M_{GK} for the two datasets RRD_{CMD} and RRD_{HMD}, respectively. The highest accuracy for every method is highlighted in bold for both the datasets. Due to the poor performance of SVML and NBB classifiers in the experiments we do not consider them in our further analysis.

We start by analyzing the results of the variants of M_{VSM} for both the datasets. From Table 6.1 and Table 6.2, we see that the highest accuracy obtained by M_{VSM} for RRD_{CMD} is 67.29%, and for RRD_{HMD} is 82.66%. The difference in the performance across the two datasets can be attributed to several factors. One of them being the difference in the number of rāgas across the two datasets: 40 in RRD_{CMD} and 30 RRD_{HMD}. The performance difference can also be because of the difference in the

Method	Feature	SVML	SGD	NBM	NBG	RF	LR	1-NN
M_{VSM}	f_1	71	72.33	69.33	79.33	38.66	74.33	-
	f_2	65.33	64.33	67.66	72.66	40.33	68	-
	f_3	65.33	62.66	82.66	72	41.33	67.66	-
M_{PC}	PCDfull	-	-	-	-	-	-	91.66

Table 6.2: Accuracy (%) of rāga recognition on RRD_{HMD} dataset by M_{VSM} and other methods methods using different features and classifiers. Bold text signifies the best accuracy by a method among all its variants.

overall length of the audio recordings. Recordings of the music pieces considered in our datasets are significantly longer in Hindustani music compared to Carnatic music (Section 3.3.4). However, since the melodic characteristics and complexity varies considerably across the two music traditions, a definite reason can only be identified by performing the experiments with equal number of rāgas in the dataset and equal duration of the music pieces (Section 6.4).

From Table 6.1 and Table 6.2, we see that for both the datasets, the accuracy obtained by M_{VSM} across the feature sets differs substantially. We also see that their performance is very sensitive to the choice of the classifier. With the exceptions of the NBM and LR classifier, feature f_1 in general performs better than the other two features and the difference is found to be statistically significant in each case. This suggests that, considering just the presence or the absence of a melodic pattern, irrespective of its frequency of occurrence, might be sufficient for rāga recognition. Interestingly, this finding is consistent with the fact that characteristic melodic patterns are unique to a rāga and a single occurrence of such patterns is sufficient to identify the rāga (Krishna & Ishwar, 2012). However, the best performance is obtained by the combination of f_3 and NBM classifier, and the difference in its performance compared to any other variant is statistically significant. The NBM classifier outperforming other classifiers using appropriate features is also well recognized in the text classification community (McCallum & Nigam, 1998). We, therefore, only consider the combination of f_3 and the NBM classifier for comparing M_{VSM} with the other methods. Overall, the accuracies across different features indicate that normalizing the frequency of occurrence of melodic patterns (as done for f_1 and f_3) is beneficial for rāga recognition. A plausible reason can be that a normalization procedure reduces the weight of the type of melodic patterns that occur more frequently, such as the gamakas type patterns (Section 2.3.2), and are not the characteristic patterns of any particular rāga. It is worth noting that the feature weights assigned by a classifier can be used to identify the relevant melodic patterns for rāga recognition. These patterns can serve as a dictionary of semantically-meaningful melodic units for many computational tasks in IAM.

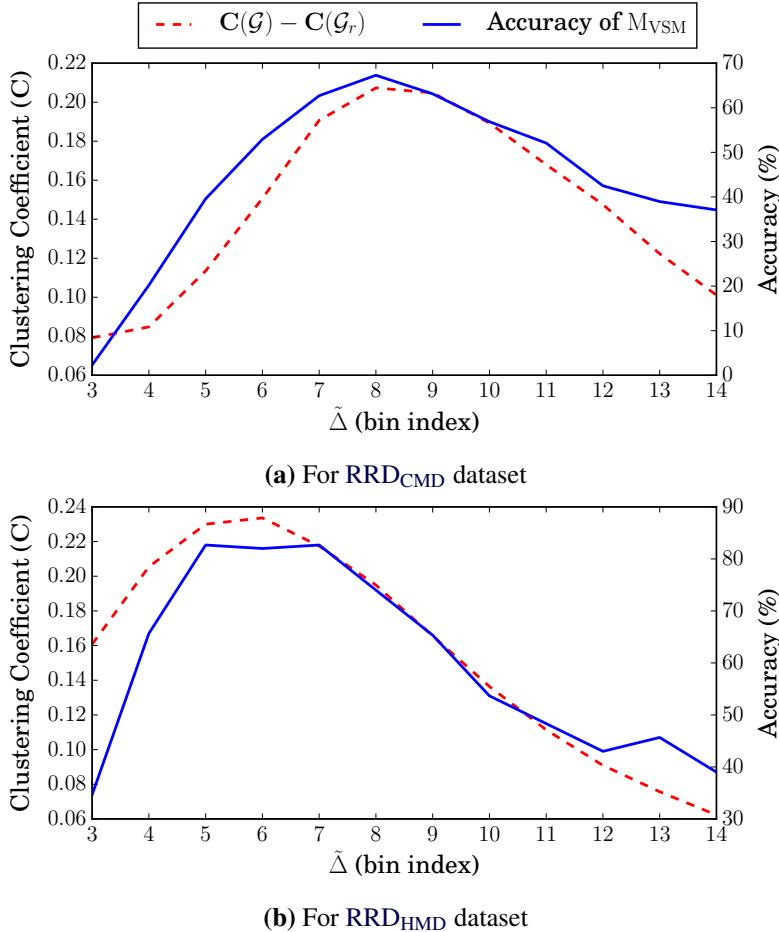


Figure 6.3: Accuracy of M_{VSM} and $C(\mathcal{G}) - C(\mathcal{G}_r)$ for different similarity thresholds ($\tilde{\Delta}$) for both the datasets.

In addition to analyzing the final rāga recognition accuracies, we also verify our approach to obtain the optimal similarity threshold $\tilde{\Delta}^*$ (Section 6.2.1.2). For this, we perform rāga recognition using different similarity thresholds $\tilde{\Delta}$. In Figure 6.3, we show the accuracy obtained by M_{VSM} , and $C(\mathcal{G}) - C(\mathcal{G}_r)$ as a function of similarity threshold $\tilde{\Delta}$ for both the datasets. We see that these curves are highly correlated for both the datasets. Thus, we see that our strategy to obtain the optimal threshold $\tilde{\Delta}^*$, which we defined in Section 6.2.1.2 as the distance ($\tilde{\Delta}$) that maximizes the difference $C(\mathcal{G}) - C(\mathcal{G}_r)$, is successful and, it results in the best rāga recognition accuracy.

We now analyze the confusion matrix to understand the type of classification errors made by M_{VSM} . In Figure 6.4 we show the confusion matrix of the rāga predictions on RRD_{CMD} dataset. We observe that our method achieves near-perfect accuracy for several rāgas including **Rītigauṇa**, **Behāg**, **Mukhāri**, **Māyāmālāvagauṇa**, **Śankarābharaṇam**, **Nāṭa**, **Tōḍī**, **Kalyāṇi**, **Pūrvikalyāṇi** and **Varāli**. This is consistent

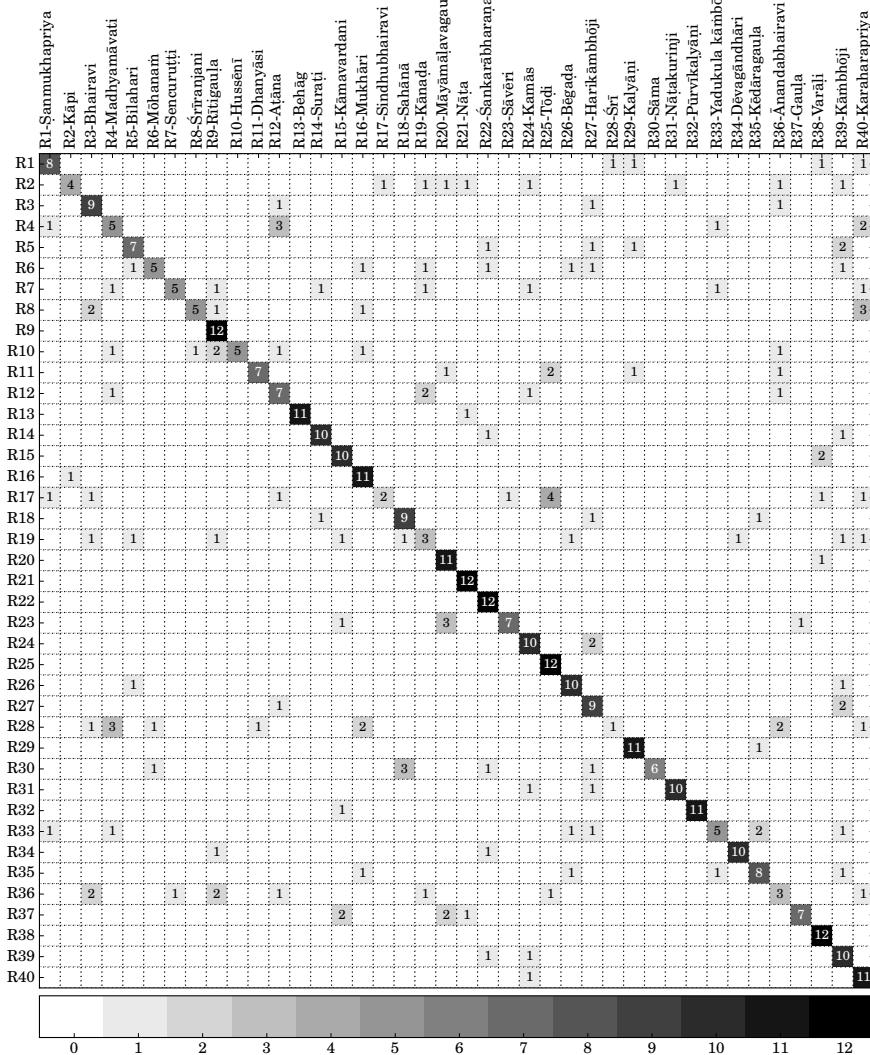


Figure 6.4: Confusion matrix of the `rāga` predictions by `MvSM` on `RRDCMD` dataset. The different shades of grey are mapped to different number of audio recordings.

with the fact that these are considered to be phrase-based *rāgas*, that is, their identity is predominantly derived from melodic phraseology (Krishna & Ishwar, 2012). From Figure 6.4 we notice that the frequent confusions are between *rāga* Sindhuhair-avi and *Tōdī*, *Sāvēri* and *Māyāmālava*, *Śrī* and *Madhyamāvati*, *Aṭāna* and *Madhyamāvati*, *Śrīranjani* and *Karaharapriya* and *Sāma* and *Sahānā*. On investigating it further we find that in almost every case of confusion the pair comprises allied *rāgas* (Section 2.3.2.1), i.e. *rāgas* that have a common set of svaras and similar melodic movements. Distinguishing between allied *rāgas* is a challenging task, since it is

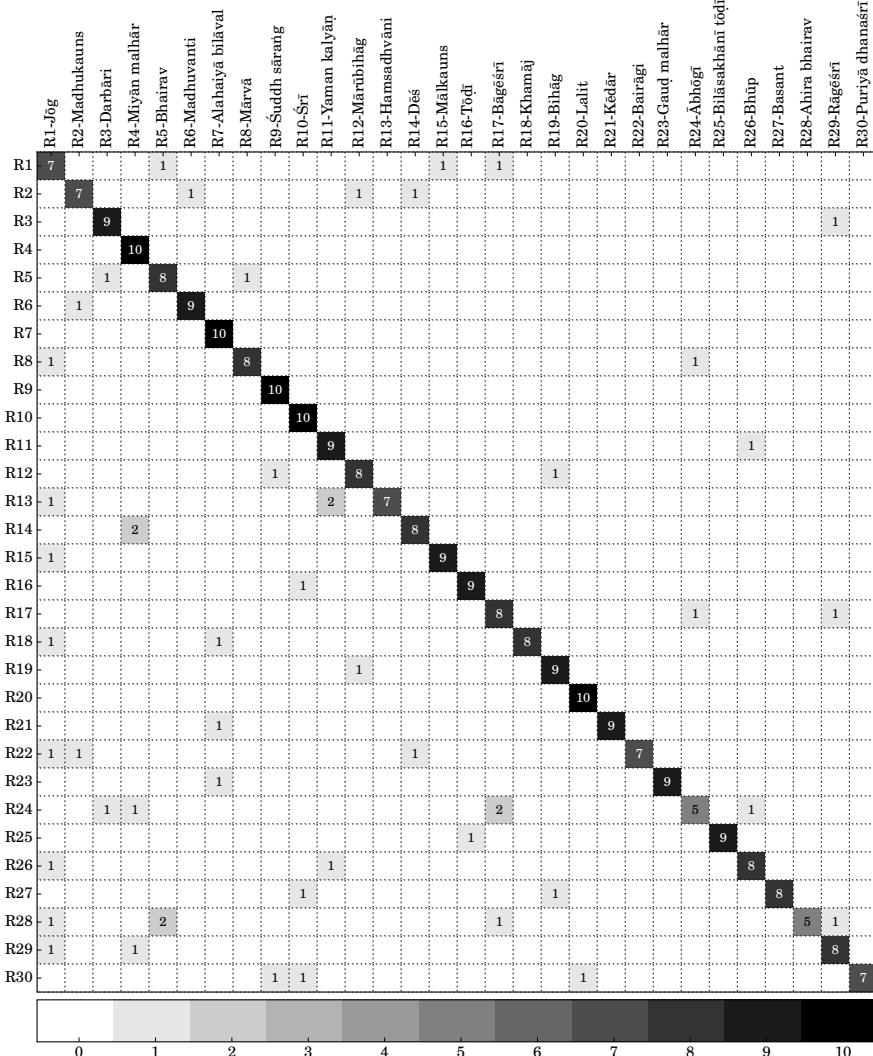


Figure 6.5: Confusion matrix of the rāga predictions by MvSM on RRD_{HMD} dataset. The different shades of grey are mapped to different number of audio recordings.

based on subtle melodic nuances. We also note that, some cases where the accuracy is relatively low correspond to scale-based rāgas. This is in line with Krishna & Ishwar (2012), where the authors remark that the identification of such rāgas is not based on melodic phraseology.

We now analyze the confusion matrix of the rāga predictions on RRD_{HMD} dataset (Figure 6.5). We observe that our method achieves good accuracy for rāga Miyāñ malhār, Sūddh sāraṅg, Alahaiyā bilāval, Śrī, Lalit, Yaman kalyāñ, Kēdār, Gaud malhār and Bilāsakhāñi tōdī. Similar to the case of Carnatic music, these rāgas are phrase-

based rāgas of Hindustani music. For common confusions as well we find the same reason that most of them occur between allied rāgas. For example, we notice that *Ahira bhairav* is confused with *Bhairav*. This is explicable since the former is derived from the latter rāga and they both share the same lower tetrachord structure. We see that rāga *Dēś* is confused with *Miyān malhār*. Both these rāgas come within the malhār group of rāgas and bear commonalities. We find a similar explanation for most of the confusions in the predictions. Going ahead, we notice from Figure 6.5 that the performance of our method is poor for rāgas including *Ābhōgī*, *Jōg* and *Madhukauns*. An interesting common factor across these rāgas is that they are pentatonic rāgas comprising five svaras (*Jōg* is pentatonic in ascending progression). This suggests that rāgas with less number of comprising svaras are more prone to get confused with other rāgas. However, further instigation of this phenomenon is left for the future work. Overall, the analysis of the classification errors for both Carnatic and Hindustani music datasets suggests that our proposed method is more suitable for recognizing the phrase-based rāgas compared to the scale-based rāgas. In addition, we see that the common mistakes done by the method are explicable from a musicological perspective.

Finally, we compare M_{VSM} with M_{PC} and M_{GK} methods. From Table 6.1, we see that M_{VSM} outperforms M_{GK} and the difference is found to be statistically significant. When compared with M_{PC} , the performance of M_{VSM} is inferior on both the datasets, wherein it is worse on RRD_{HMD} dataset. In both the cases the difference in the performance is statistically significant across the methods. One of the plausible reasons for this difference can be that M_{PC} utilizes entire full length recordings for this task, whereas, M_{VSM} uses only short-time melodic patterns which sum up to only a fraction of the total duration of the performances. Since the recordings are considerably longer in RRD_{HMD} dataset, the difference in the accuracies is even more prominent. Note that for comparison in our study we use PCD_{full} variant of M_{PC} , which performs significantly better than the original variant proposed by the authors, PCD_{120} (Gulati et al., 2016b). Interestingly, a further comparison of the results of M_{VSM} and M_{PC} for each rāga reveals that their performance is complementary. M_{VSM} successfully recognizes several rāgas with high accuracy for which M_{PC} performs poorly, and vice-versa. This becomes evident by comparing the confusion matrix of the rāga predictions by M_{VSM} (Figure 6.4 and Figure 6.5) and M_{PC} (Figure C.2 and Figure C.3). For example, for the case of RRD_{CMD} dataset, M_{VSM} performs better for rāgas *Madhyamāvati*, *Mukhāri* and *Harikāmbhōji* as compared to M_{PC} . And, M_{PC} performs better for rāgas *Kāpi*, *Sindhuhairavi* and *Ānandabhairavi* as compared to M_{VSM} . This suggests that the proposed pattern-based method can be combined with the PCD -based methods to achieve a higher rāga recognition accuracy.

Overall, our results indicate that the proposed phrase-based approach (M_{VSM}) that uses melodic patterns discovered in an unsupervised setup is a successful strategy for rāga recognition. However, as we have seen in comparison with the other methods there is still scope for improvement. One of the advantages of this approach is the

interpretability of the results. Since the classification is based on melodic phrases, the user can better understand why the classifier is assigning a certain *rāga* label. In addition, based on the weight that a classifier assigns to a particular feature, the user can also infer the importance of that pattern in characterizing the *rāga*. Furthermore, since this method does not require a pitch distribution of an entire music piece, and is solely based on melodic patterns, it can be used in a real-time *rāga* recognition setup. Finally, the results of **M_{VSM}** also suggest that the discovered melodic patterns are musically relevant and can be used to perform higher level melodic analyses.

6.3 Time Delayed Melodic Surface for Rāga Recognition

The method described in the previous section (**M_{VSM}**) uses melodic patterns for recognizing *rāgas*. Using automatically discovered short-duration melodic patterns that constitute only a fraction of the total duration of the audio recordings, **M_{VSM}** shows promising results by achieving an accuracy comparable to the state of the art method. As discussed before, **M_{VSM}** has several advantages such as musically meaningful interpretation of the classification results. While we further refine our methodology for discovering melodic patterns and in turn improve *rāga* recognition, we also propose another approach (**M_{TDMS}**) for this task.

Similar to **M_{VSM}**, **M_{TDMS}** aims to capture the tonal and the temporal characteristics of melody by using its continuous representation. However, instead of using short-duration patterns extracted directly from the surface representation of melody as done in **M_{VSM}**, in **M_{TDMS}** we seek to abstract the melody representation. An abstraction of melody that captures both the tonal and the temporal aspects relates to the concept of *chalan* in **IAM** (Section 2.3.2). *Chalan* (literally meaning gait or movement) of a *rāga* defines its melodic outline in terms of how a melodic transition is to be made from one *svara* to another, the precise intonation to be followed, and the proportion of time spent on each *svara*. *Chalan* can be considered as an abstraction of *rāga* motifs, and is a characterizing feature of *rāgas*. **M_{TDMS}** utilizes *chalan* aspect of melodies in **IAM** to perform *rāga* recognition.

In order to abstract the continuous melody representation and incorporate *chalan* aspects, **M_{TDMS}** uses a novel feature, the *time delayed melodic surface* (TDMS). TDMS captures tonal and temporal melodic aspects that are useful in characterizing and distinguishing *rāgas*. TDMSs alleviate several of the shortcomings in the existing approaches (Section 2.4.3) and improves the accuracy of *rāga* recognition by large margins. TDMS is inspired by the concept of delay coordinates (Takens, 1981). The main strengths of TDMS are:

- It is a compact representation that describes both the tonal and the temporal characteristics of a melody
- It simultaneously captures the melodic characteristics at different time-scales,

the overall usage of the pitch-classes in the entire recording, and the short-time temporal relation between individual pitches.

- It is robust to pitch octave errors.
- It does not require a svara-level transcription of the melody nor its discrete representation.
- It has few parameters that are easy to tune (when set within reasonable bounds).
- It is easy to implement, fast to compute, and has a musically meaningful interpretation.
- As it will be shown, it obtains unprecedented accuracies in the rāga recognition task, outperforming the state of the art by a large margin, without the use of any elaborated classification schema.

We now proceed to describe M_{TDMS} , which is based on the TDMS features. The computation of TDMS is described in the subsequent section (Section 6.3.1). The classification strategy and the distance measure used on top of the TDMS features for recognizing rāgas are described in Section 6.3.2.2. In Section 6.3.2, we present the experimental setup used to evaluate this method, and finally, discuss the obtained results (Section 6.3.3).

6.3.1 Time Delayed Melodic Surface

The computation of a TDMS involves three steps as shown in Figure 6.6: pre-processing, surface generation, and post-processing. In the pre-processing step, we obtain a low-level representation of the melody from an audio recording, which is then put to a meaningful tonal context by normalizing it with respect to the tonic pitch of the recording. In the surface generation step, we compute a two dimensional surface based on the concept of delay coordinates. Finally, in the post-processing step, we apply power compression and Gaussian smoothing to the computed surface. All these processing steps are described in the subsequent sections.

6.3.1.1 Pre-processing

Predominant Pitch Estimation: In this step we process an audio recording to obtain a low-level representation of the melody, which is subsequently used in the computation of the TDMS. For this, we consider the predominant pitch in the audio signal as the melody representation. For predominant pitch estimation we follow the method described in Section 4.3.1. In addition, we also post-process the estimated pitch to smoothen it and remove spurious pitch jumps (Section 4.3.2).

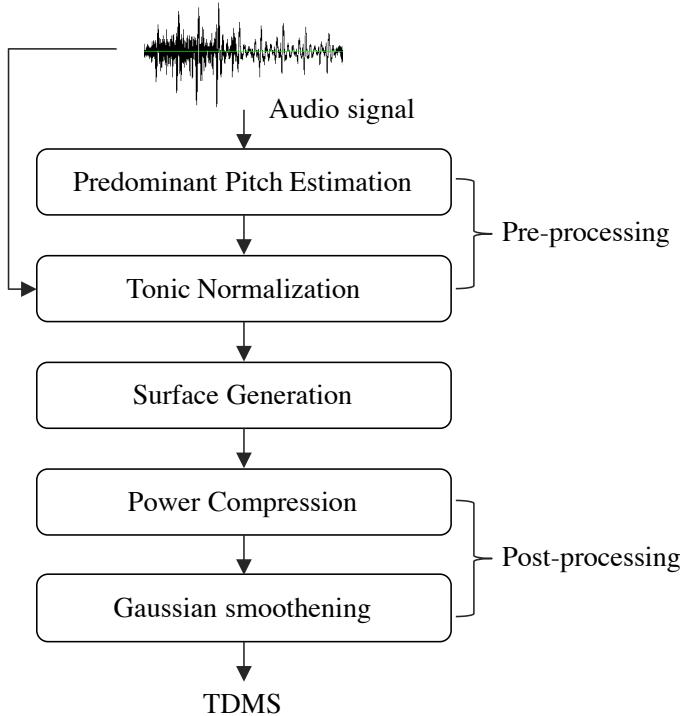


Figure 6.6: Block diagram for TDMS computation.

Tonic Normalization: The base frequency chosen in a performance of IAM is the tonic pitch of the lead artist, to which all other accompanying instruments are tuned (Section 2.3.2). Tonic pitch varies across artists and their recordings, and therefore, a meaningful feature for rāga recognition should be normalized with respect to the tonic pitch. To achieve this, we normalize the predominant pitch of every recording by considering its tonic pitch as the reference frequency during the Hertz-to-Cent-scale conversion (Section 4.3.3). The tonic pitch of the lead artist for every recording is automatically identified using MJS, the method that outperformed all other existing methods for this task as shown in our comparative evaluation (Section 4.2.3).

6.3.1.2 Surface Generation

The next step is to construct a two-dimensional surface based on the concept of delay coordinates (also termed phase space embedding) (Takens, 1981; Kantz & Schreiber, 2004). In fact, such two-dimensional surface can be seen as a discretized histogram of the elements in a two-dimensional Poicaré map (Kantz & Schreiber, 2004). For a

given recording, we generate a surface $\check{\mathbf{S}}$ of size $\eta \times \eta$ by computing

$$\check{s}_{ij} = \sum_{t=\tau}^{N-1} I(\mathbf{B}(\hat{p}_t), i) I(\mathbf{B}(\hat{p}_{t-\tau}), j) \quad (6.4)$$

for $0 \leq i, j < \eta$, where \check{s}_{ij} is the $(i, j)^{\text{th}}$ element of the two-dimensional matrix $\check{\mathbf{S}}$, \hat{p}_t is the t^{th} sample (in Cent-scale) of the pitch sequence of length N , I is an indicator function such that

$$I(x, y) = \begin{cases} 1, & \text{iff } x = y \\ 0, & \text{otherwise} \end{cases} \quad (6.5)$$

\mathbf{B} is an octave-wrapping integer binning operator defined by

$$\mathbf{B}(x) = \left\lfloor \left(\frac{\eta x}{1200} \right) \bmod \eta \right\rfloor, \quad (6.6)$$

and τ is a time delay index (in frames) that is left as a parameter. Note that, the frames where a predominant pitch could not be obtained (unvoiced segments) are excluded from any calculation. For the size of $\check{\mathbf{S}}$ we use $\eta = 120$. This value corresponds to 10 Cents per bin, an optimal pitch resolution reported by Chordia & Şentürk (2013). In that study, the authors show that rāga recognition using PCDs with a bin width of 10 Cents outperforms the PCDs with a bin width of 100 Cents, and obtains a comparable results to PCDs with a bin width of 5 Cents.

An example of the generated surface $\check{\mathbf{S}}$ for a music piece⁵⁴ in rāga Yaman is shown in Figure 6.7(a). We see that the prominent peaks in the surface correspond to the svaras of rāga Yaman. We notice that these peaks are steep and the dynamic range of the surface is high. This can be attributed to the nature of the melodies in these music traditions, particularly in Hindustani music, where the melodies often contain long held svaras. In addition, the dynamic range is high also because the pitches in the stable svara regions lie within a small frequency range around the mean svara frequency compared to the pitches in the transitory melodic regions. Because of this, most of the pitch values are mapped to a small number of bins, making the prominent peaks more steep.

6.3.1.3 Post-processing

Power Compression: In order to accentuate the values corresponding to the transitory regions in the melody and reduce the dynamic range of the surface, we apply an element-wise power compression

$$\bar{\mathbf{S}} = \check{\mathbf{S}}^\alpha, \quad (6.7)$$

where α is an exponent that is left as a parameter. Once a more compact (in terms of the dynamic range) surface is obtained, we apply Gaussian smoothing. With that, we attempt to attenuate the subtle differences in $\bar{\mathbf{S}}$ corresponding to the different melodies within the same rāga, while retaining the attributes that characterize that rāga.

⁵⁴<http://musicbrainz.org/recording/e59642ca-72bc-466b-bf4b-d82bfbc7b4af>

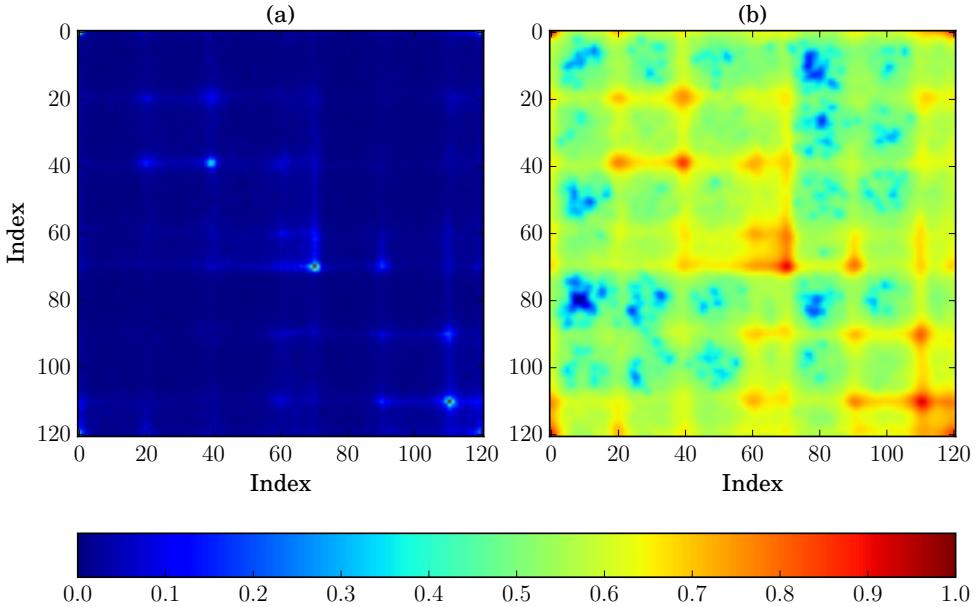


Figure 6.7: Generated surface for a music piece before (a) and after (b) applying post-processing ($\bar{\mathbf{S}}$ and $\hat{\mathbf{S}}$, respectively). For ease of visualization, both matrices are normalized here between 0 and 1.

Gaussian Smoothening: We perform Gaussian smoothing by circularly convolving $\bar{\mathbf{S}}$ with a two-dimensional Gaussian kernel. We choose a circular convolution because of the cyclic (or octave-folded) nature of the TDMS (Eq. 6.6), which mimics the cyclic nature of pitch-classes. The standard deviation of this kernel is σ_g bins (samples). The length of the kernel is truncated to $8\sigma_g + 1$ bins in each dimension, after which the values are negligible (below 0.01% of the kernel’s maximum amplitude). We experiment with different values of σ_g , and also with a method variant excluding the Gaussian smoothing (loosely denoted by $\sigma_g = -1$), so that we can quantify its influence on the accuracy of the system.

Once we have the smoothed surface $\hat{\mathbf{S}}$, there is only one step remaining to obtain the final TDMS. Since the overall duration of the recordings and of the voiced regions within them is different, the computed surface $\hat{\mathbf{S}}$ needs to be normalized. To do so, we divide $\hat{\mathbf{S}}$ by its L1-norm:

$$\mathbf{S} = \hat{\mathbf{S}} / \|\hat{\mathbf{S}}\|_1. \quad (6.8)$$

This also yields values of \mathbf{S} , the final TDMS, that are interpretable in terms of discrete probabilities.

The result after post-processing the surface in Figure 6.7 (a) with power compression and Gaussian smoothing is shown in Figure 6.7 (b). We see that the values corresponding to the non-diagonal elements are accentuated. A visual inspection of Fig-

ure 6.7 (b) provides several musical insights to the melodic aspects of the recording. For instance, the high salience indices along the diagonal, (0,0), (20,20), (40,40), (60,60), (70,70), (90,90), and (110,110), correspond to the 7 svaras used in rāga Yaman. Within which, the highest salience at indices (110,110) correspond to the Nī svara, which is the vādi svara, that is, musically the most salient svara of the rāga, in this case rāga Yaman (Rao et al., 1999). The asymmetry in the matrix with respect to the diagonal indicates the asymmetric nature of the ascending and descending svara progression, ārōhana-avrōhana, of the rāga (compare, for example, the salience at indices (70,90) to indices (90,70), with the former being more salient than the latter). The similarity of the matrix between indices (20,20) and (70,70) with respect to the matrix between indices (70,70) and (120,120) delineates the tetra-chord structure of the rāga. Thus, we see that the TDMS captures several features related with the tonal and the temporal aspects of melody at different time-scales. Finally, it should be noted that an interesting property of the TDMS feature is that the mean of the sum across its rows and columns yields a PCD representation, widely used in rāga recognition (Section 2.4.3).

6.3.2 Evaluation

6.3.2.1 Music Collection

To evaluate M_{TDMS} method we use the same music collection as used in the evaluation of M_{VSM} (Section 6.2.2.1). It comprises two datasets, RRD_{CMD} and RRD_{HMD} , one for each music tradition, Carnatic and Hindustani music, respectively. As mentioned before, a separate evaluation on both the music traditions allows a better analysis and interpretation of the results.

To reiterate, these are the largest datasets ever used for studying the task of automatic rāga recognition. To facilitate reproducible research and comparative studies we also make these datasets publicly available online (Appendix B).

6.3.2.2 Classification and Distance Measures

In order to demonstrate the ability of the TDMSs in capturing rāga characteristics, we consider the task of classifying audio recordings according to their rāga label. To perform classification, we choose a k -nearest neighbors (k -NN) classifier (Mitchell, 1997). The reasons for our choice are manifold. Firstly, the k -NN classifier is well understood, with well studied relations to other classifiers in terms of both performance and architecture. Secondly, it is fast, with practically no training and with known techniques to speed up testing or retrieval. Thirdly, it has only one parameter, k , which we can just blindly set to a relatively small value or can easily optimize in the training phase. Finally, it is a classifier that is simple to implement and whose results are both interpretable and easily reproducible.

The performance of a k -NN classifier highly depends on the distance measure used to retrieve the k neighbors. We consider three different measures to compute the distance between two recordings n and m with TDMS features $\mathbf{S}^{(n)}$ and $\mathbf{S}^{(m)}$, respectively. We first consider the Frobenius norm of the difference between $\mathbf{S}^{(n)}$ and $\mathbf{S}^{(m)}$,

$$\tilde{\delta}_{\text{F}}^{(n,m)} = \|\mathbf{S}^{(n)} - \mathbf{S}^{(m)}\|_2. \quad (6.9)$$

Next, we consider the symmetric Kullback-Leibler divergence

$$\tilde{\delta}_{\text{KL}}^{(n,m)} = D_{\text{KL}}(\mathbf{S}^{(n)}, \mathbf{S}^{(m)}) + D_{\text{KL}}(\mathbf{S}^{(m)}, \mathbf{S}^{(n)}), \quad (6.10)$$

with

$$D_{\text{KL}}(\mathbf{X}, \mathbf{Y}) = \sum \mathbf{X} \log \left(\frac{\mathbf{X}}{\mathbf{Y}} \right), \quad (6.11)$$

where we perform element-wise operations and sum over all the elements of the resultant matrix. Finally, we consider the Bhattacharyya distance, which is reported to outperform other distance measures with a PCD-based feature for the same task in Chordia & Şentürk (2013),

$$\tilde{\delta}_{\text{B}}^{(n,m)} = -\log \left(\sum \sqrt{\mathbf{S}^{(n)} \cdot \mathbf{S}^{(m)}} \right). \quad (6.12)$$

We again perform element-wise operations and sum over all the elements of the resultant matrix. Variants of our proposed method that use $\tilde{\delta}_{\text{F}}$, $\tilde{\delta}_{\text{KL}}$ and $\tilde{\delta}_{\text{B}}$ are denoted by $M_{\text{TDMS}}^{\text{F}}$, $M_{\text{TDMS}}^{\text{KL}}$, and $M_{\text{TDMS}}^{\text{B}}$, respectively.

6.3.2.3 Comparison with Other Methods

In addition to M_{TDMS} , we also evaluate and compare the method proposed by Chordia & Şentürk (2013) (denoted by M_{PC}) in the same way as explained in Section 6.2.2.3. M_{PC} is regarded as the state of the art in rāga recognition, which uses smoothed pitch-class distribution (PCD) as the tonal feature and employs 1-nearest neighbor (1-NN) using Bhattacharyya distance for predicting rāga labels. We also compare the performance of M_{TDMS} with our melodic pattern-based method, M_{VSM} , and with M_{GK} (Section 6.2).

6.3.2.4 Experimental Setup

To evaluate M_{TDMS} we use the same experimental setup as used in the evaluation of M_{VSM} (Section 6.2.2.2). We perform a leave-one-out cross validation (Mitchell, 1997), in which one recording from the evaluation data set forms the testing set and the remaining ones become the training set. To quantify the performance of the considered methods we use the raw overall accuracy (Mitchell, 1997). Since both RRD_{CMD} and RRD_{HMD} are balanced datasets in the number of instances per class, we do not need to correct such raw accuracies to counteract for possible biases towards

Dataset	M_{TDMS}^F	M_{TDMS}^{KL}	M_{TDMS}^B	M_{VSM}	M_{PC}	M_{GK}
RRD_{HMD}	91.3	97.7	97.7	83.0	91.7	NA
RRD_{CMD}	81.5	86.7	86.7	67.3	73.1	54.8

Table 6.3: Accuracy (%) of the three proposed variants, M_{TDMS}^F , M_{TDMS}^{KL} and M_{TDMS}^B , and other methods M_{VSM} , M_{PC} and M_{GK} . The random baseline (\mathfrak{B}_r) for this task is 3.3% for RRD_{HMD} and 2.5% for RRD_{CMD} . NA stands for not applicable.

the majority class. To assess if the difference in the performance between any two methods is statistically significant, we use McNemar’s test (McNemar, 1947) with $p < 0.01$. To compensate for multiple comparisons, we apply the Holm-Bonferroni method (Holm, 1979). Besides accuracy, and for a more detailed error analysis, we also compute the confusion matrix over the predicted classes.

In the case of M_{TDMS} , a test recording is assigned the majority class of its k -nearest neighbors obtained from the training set and, in case of a tie, one of the majority classes is selected randomly. Because we conjecture that none of the parameters we consider is critical to obtain a good performance, we initially make an educated guess and intuitively set our parameters to a specific combination. We later study the influence of every parameter starting from that combination. We initially use $\tau = 0.3$ s, $\alpha = 0.75$, $\sigma_g = 2$, and $k = 1$, and later consider $\tau \in \{0.2, 0.3, 0.5, 1, 1.5\}$ s, $\alpha \in \{0.1, 0.25, 0.5, 0.75, 1\}$, $\sigma_g \in \{-1, 1, 2, 3\}$, and $k \in \{1, 3, 5\}$ (recall that $\sigma_g = -1$ corresponds to no smoothing Section 6.3.1.3).

6.3.3 Results and Discussion

In Table 6.3, we show the results for all the variants of the proposed method M_{TDMS}^F , M_{TDMS}^{KL} and M_{TDMS}^B , and for M_{VSM} , M_{PC} and M_{GK} , using RRD_{HMD} and RRD_{CMD} datasets. We see that the highest accuracy obtained on RRD_{HMD} is 97.7% by M_{TDMS}^{KL} and M_{TDMS}^B . This accuracy is considerably higher than the 91.7% obtained by M_{PC} , and the difference is found to be statistically significant. A comparison between the methods M_{VSM} , M_{PC} and M_{GK} is already presented in Section 6.2.3. Regarding the proposed variants, we see that, in RRD_{HMD} , M_{TDMS}^{KL} and M_{TDMS}^B perform better than M_{TDMS}^F , with a statistically significant difference.

In Table 6.3, we also see that the trend in the performance for RRD_{CMD} across different methods is similar to that for RRD_{HMD} . The variants M_{TDMS}^{KL} and M_{TDMS}^B achieve the highest accuracy of 86.7%, followed by M_{PC} with 73.1%. The difference in performance between both the methods, M_{TDMS}^{KL} and M_{TDMS}^B , and M_{PC} is found to be statistically significant. For RRD_{CMD} also, M_{TDMS}^{KL} and M_{TDMS}^B perform better than M_{TDMS}^F , with a statistically significant difference.

In general, we notice that, for every method, the accuracy is higher on RRD_{HMD} compared to RRD_{CMD} . This, as expected, can be largely attributed to the difference

in the number of classes in RRD_{HMD} (30 rāgas) and RRD_{CMD} (40 rāgas). A higher number of classes makes the task of rāga recognition more challenging for RRD_{CMD} , compared to RRD_{HMD} . In addition to that, another factor that can cause this difference could be the length of the audio recordings, which for RRD_{HMD} are significantly longer than the ones in RRD_{CMD} (Section 3.3.4).

As mentioned earlier, the system parameters corresponding to the results in Table 6.3 were set intuitively, without any parameter tuning. Since TDMSs are novel features that are being used for the first time, we want to carefully analyze the influence that each of the parameters has on the final rāga recognition accuracy, and ultimately perform a quantitative assessment of their importance. In Figure 6.8, we show the accuracy of $M_{\text{TDMS}}^{\text{KL}}$ for different values of these parameters. In each case, only one parameter is varied and the rest are set to the initial values mentioned above.

In Figure 6.8 (a), we observe that the performance of the method is quite invariant to the choice of τ , except for the extreme delay values of 1 and 1.5 s for RRD_{CMD} . This can be attributed to the melodic characteristics of Carnatic music, which presents a higher degree of oscillatory melody movements and shorter stationary svara regions, as compared to Hindustani music. In Figure 6.8 (b), we see that compression with $\alpha < 1$ slightly improves the performance of the method for both datasets. However, the performance degrades for $\alpha < 0.75$ for RRD_{CMD} and $\alpha < 0.25$ for RRD_{HMD} . This again appears to be correlated with the long steady nature of the svaras in Hindustani music melodies. Because the dynamic range of Š is high, TDMS features require a lower value for the compression factor α to accentuate the surface values corresponding to the transitory regions in the melodies of Hindustani music. In Figure 6.8 (c), we observe that Gaussian smoothing significantly improves the performance of the method, and that such performance is invariant across the chosen values of σ_g . Finally, in Figure 6.8 (d), we notice that the accuracy decreases with increasing k . This is also expected due to the relatively small number of samples per class in our datasets (Mitchell, 1997). The best accuracy obtained is for $k = 1$. A similar observation is also reported in a previous study that uses PCD-based feature for the same task (Chordia & Şentürk, 2013).

Overall, the method appears to be invariant to different parameter values to a large extent, which implies that it is easy to extend and tune it to other datasets. It is important to note that, no matter what parameter configuration we use, the accuracy of $M_{\text{TDMS}}^{\text{KL}}$ never goes below the baselines' accuracies (see Figure 6.8).

From the results reported in Figure 6.8, we see that there exist a number of parameter combinations that could potentially yield a better accuracy than the one reported in Table 6.3. For instance, using $\tau = 0.3$ s, $\alpha = 0.5$, $\sigma_g = 2$, and $k = 1$, we are able to reach 97.0% for $M_{\text{TDMS}}^{\text{F}}$ and 98.0% for both $M_{\text{TDMS}}^{\text{KL}}$ and $M_{\text{TDMS}}^{\text{B}}$ on RRD_{HMD} . These accuracies are ad-hoc, optimizing the parameters on the testing set. However, and doing things more properly, we could learn the optimal parameters in training, through a standard grid search, cross-validated procedure over the training set (Mitchell, 1997;

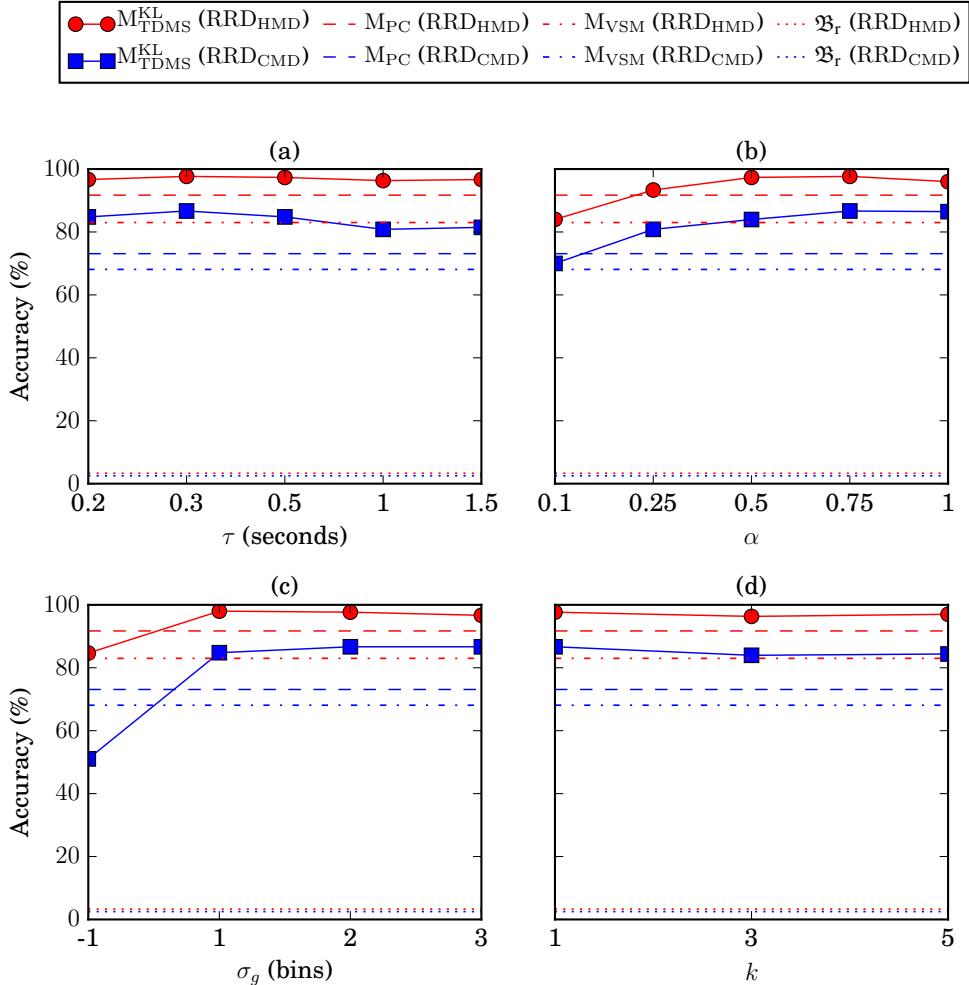


Figure 6.8: Accuracy of M_{TDMS}^{KL} as a function of parameter values. Other methods, M_{PC} , M_{VSM} , and random baseline \mathcal{B}_r are also reported for comparison.

Hastie et al., 2009b). As our primary goal here is not to obtain the best possible results, but to show the usefulness and superiority of TDMSs, we do not perform such an exhaustive parameter tuning and leave it for future research.

We now proceed to analyze the errors made by the best performing variant M_{TDMS}^{KL} . For RRD_{CMD}, we show the confusion matrix of the predicted rāga labels in Figure 6.9. In general, we see that the confusions have a musical explanation. The majority of them are between the rāgas in the sets {Bhairavi, Mukhāri}, {Harikāmbhōji, Kāmbhōji}, {Madhyamāvati, Aṭāna, Śrī}, and {Kāpi, Ānandabhairavi}. Rāgas within each of these sets are allied rāgas (Viswanathan & Allen, 2004) (Section 2.3.2), i.e., they share a common set of svaras and similar phrases. Due to these characteris-

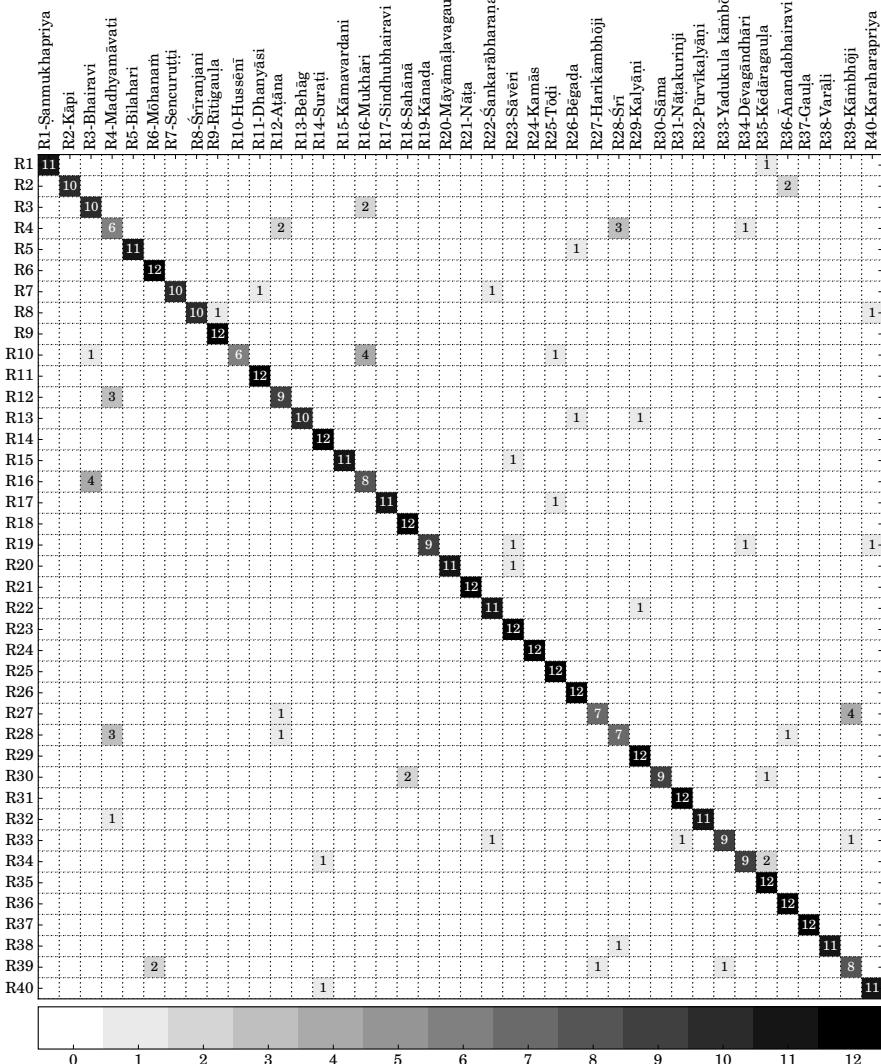


Figure 6.9: Confusion matrix of the predicted rāga labels obtained by $M_{\text{TDMS}}^{\text{KL}}$ on RRD_{CMD}. Shades of grey are mapped to the number of audio recordings.

ics, distinguishing between allied rāgas is a challenging task, which is often based on subtle melodic nuances.

For **RRD_{HMD}**, there are only seven incorrectly classified recordings (Figure C.4). Confusions between **Rāga Alahaiyā bilāval**, **Dēś** and **Khamāj** is explicable as these rāgas share exactly the same set of svaras. Similar is the case between **rāga Dēś** and **Gaud malhār**, wherein both the rāgas belong to the malhār group of rāgas and have similar melodic phrases. For two specific cases of confusions, that of **rāga Khamāj**

with Bāgēśrī, and rāga Darbāri with Bhūp, we find that the error lies in the estimation of the tonic pitch.

6.4 Effect of Dataset on Accuracy

From our review of the current approaches for rāga recognition in Section 2.4.3, we see that their direct comparison in terms of the performance is not meaningful. This is mainly due to the diversity in the datasets using which they are evaluated (Table 2.4). These datasets differ in terms of the number of rāgas, specific set of chosen rāgas, the number of recordings per rāga and the type of audio (monophonic or polyphonic). A comprehensive comparison of the performance of these approaches requires a common evaluation on the same dataset and experimental setup, which is an arduous task. However, to begin with, even a systematic assessment of the influence of a dataset on rāga recognition accuracy for a particular method can provide us several useful insights. Certainly, the trend in the accuracy across different dataset variants will be dependent on the chosen method. However, we hypothesize that using one of the most robust and the best performing methods will minimize this influence. In addition, we believe that it will act as a lower bound on the variations in the accuracies across the dataset variants.

For this experiment we consider M_{TDMS}^{KL} method, which as shown above achieves the highest accuracy on both the datasets, RRD_{CMD} and RRD_{HMD} . We want to assess the influence of the number of rāgas in the dataset and the specific combination of rāgas in a set. To achieve this, we perform evaluations on randomly sampled subsets of both the datasets. We randomly select sets of 5, 10, 20, and 30 rāgas from RRD_{CMD} dataset that comprise 40 rāgas with 12 recordings per rāga. We maintain the same number of recordings (12) per rāga in all the sets. For RRD_{HMD} dataset, the sizes of the subsets are 5, 10, 15, 20 and 25. We repeat this entire process of random selection 50 times and ensure that at each iteration the new subsets are maximally different from the previous ones in terms of the choice of rāga. We now proceed to present the results of our evaluations.

In Figure 6.10, we show a boxplot of the rāga recognition accuracies obtained for different subsets of the datasets over multiple iterations of the experiment. In general, we observe that the accuracy decreases as the number of rāgas in the dataset increase. This is explicable as the task becomes more and more challenging. Notably, the drop in the performance is more significant in the case of the subsets comprising Carnatic music (Figure 6.10(a)) as compared to Hindustani music. In addition, the median accuracies are lower for Carnatic music. Both these factors indicate that the task of rāga recognition is more challenging in the case of audio recordings in Carnatic music as compared to Hindustani music, and that more number of rāgas makes it even more difficult for the former.

Another interesting aspect is to analyze the performance across multiple iterations,

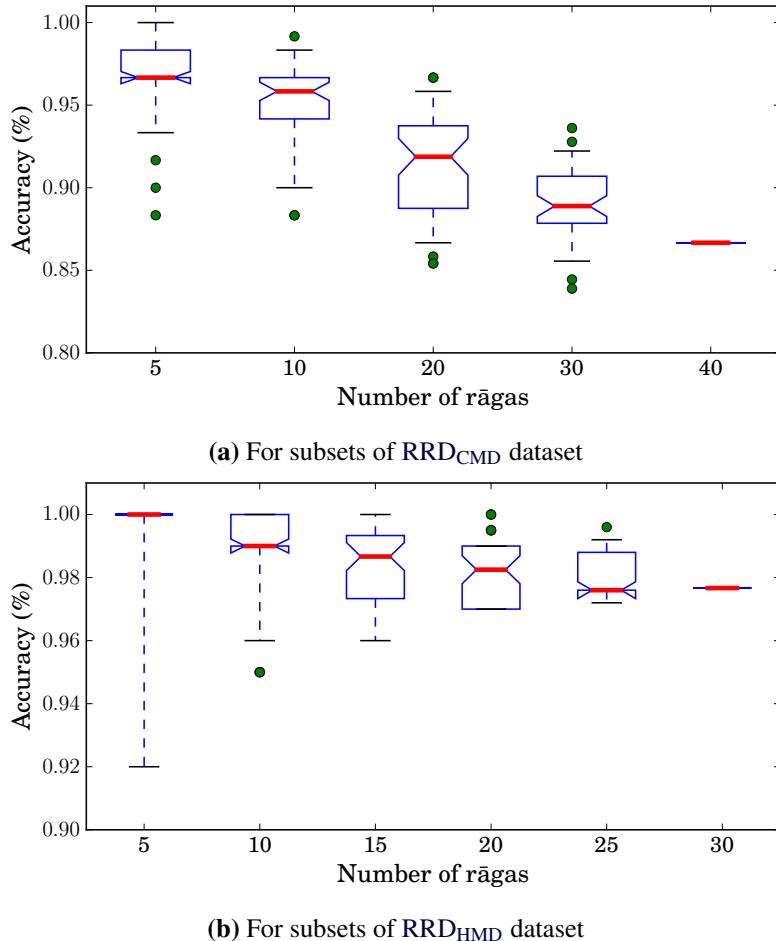


Figure 6.10: Accuracy of M_{VSM} as a function number of rāgas in a subset of RRD_{CMD} and RRD_{HMD} , respectively. For every case 50 randomly sampled selections of rāgas are done from the original datasets.

i.e. across different sets of a fix number of rāgas. We see that the accuracy varies by around 10 % for the case of Carnatic music dataset. This means that the accuracy of a method for the same number of rāgas and the same number of recordings per rāga can vary by 10% just because of a different selection of rāgas. It strongly implies that the complexity of the task of rāga recognition depends on the rāgas being differentiated. Thus, these factors should be taken into account while interpreting the results of a method in rāga recognition task. Note that for Hindustani music this variation is not as significant.

Overall, we see that the accuracy of rāga recognition is sensitive to a dataset, both in terms of the number of rāgas and the specific set of chosen rāgas. We notice that the sensitivity is more for a Carnatic music dataset as compared to a Hindustani music

dataset. An important takeaway of this experiment is the degree to which the accuracy varies across the different dataset variants. This will help us to better interpret and compare the results of the existing studies on rāga recognition.

6.5 Summary and Conclusions

In this chapter, we presented two computational approaches for automatically recognizing rāgas in audio collections of IAM, which address a number of shortcomings in the existing approaches identified in our literature review. They utilize both the tonal and the temporal characteristics of melodies for rāga recognition, and require only an estimate of the predominant pitch and the tonic of the performance from audio recordings.

We first described our pattern-based method for rāga recognition (**M_{VSM}**), which uses melodic patterns discovered in an unsupervised manner by our approach presented in the previous chapter. In order to group the patterns that represent the same melodic phrase we clustered them using a non-overlapping community detection method applied on a network built using the discovered patterns. In this process we also removed connections between the melodically dissimilar patterns by applying a similarity threshold, which we estimated by exploiting the topological properties of the network. We subsequently employed a vector space model to represent audio recordings in terms of these melodic patterns. The TF-IDF-based features thus obtained are then used to train a classifier to predict rāga labels. For evaluating and comparing our method and the state of the art methods we used the sizable Carnatic and Hindustani music collection we compiled and curated for this task. To the best of our knowledge, these are the largest datasets ever used for evaluating rāga recognition methods. We experimented with a number of classification algorithms and found that the multinomial naive Bayes classifier outperforms the rest. We showed that our pattern-based rāga recognition approach that utilize vector space modeling concepts is a successful strategy, yielding comparable accuracies to the state of the art. We see that **M_{VSM}** achieves an accuracy of 67.29% and 82.66% for **RRD_{CMD}** and **RRD_{HMD}** datasets, respectively. The accuracy obtained by the state of the art method is 73.12% and 91.6% for **RRD_{CMD}** and **RRD_{HMD}** datasets, respectively.

An analysis of the classification errors revealed that the majority of the confusions occurred between the allied rāgas, which are differentiated by experts based on subtle melodic nuances. In addition, we found that the errors made by the pattern-based and the PCD-based methods are complementary, and thus, they can be combined to improve the accuracy in rāga recognition. Being a phrase-based approach, there are several advantages of **M_{VSM}**: the features used by this method for classification have musically meaningful interpretations, the feature weight assigned by the classifier can indicate the relative importance of different melodic patterns in distinguishing between the rāgas, and finally, since this approach does not require a pitch distribution, it can be used in a real-time rāga recognition setup. To the best of our knowledge,

no other method to date has employed a fully automated methodology for discovering and selecting musically relevant melodic patterns for *rāga* recognition on this scale. The evaluation of M_{VSM} can also be regarded as an indirect quantitative evaluation of our pattern discovery approach, which based on our results, suggest that the discovered melodic patterns are musically relevant.

We subsequently described our second method (M_{TDMS}) that uses a novel feature, the **TDMS**, which captures both the tonal and the short-time temporal aspects of a melody relevant in *rāga* characterization. This feature is derived directly from the tonic-normalized predominant pitch in the audio. A visual inspection of the **TDMS** revealed interesting musical insights. We demonstrated the capabilities of **TDMSs** in capturing *rāga* characteristics by classifying audio recordings according to their *rāgas* labels. For evaluating this method we use the same music collections as mentioned above. We showed that using a *k*-NN classifier, the proposed feature outperformed the state of the art methods in *rāga* recognition with unprecedented accuracies. We see that M_{TDMS} achieves an accuracy of 86.7% and 97.7% for RRD_{CMD} and RRD_{HMD} datasets, respectively. The accuracy obtained by the state of the art method is 73.1% and 91.7% for RRD_{CMD} and RRD_{HMD} datasets, respectively. M_{TDMS} has several advantages: it is robust to pitch octave errors, it does not require a transcribed melody representation, it involves only a few parameters, it is easy to implement and fast to compute. We also studied the influence of different parameters on the accuracy obtained by **TDMSs**, and found that it is largely invariant to different parameter values. This indicates that it is easy to extend and tune this method to other datasets. An analysis of the classification errors revealed that the confusions occur between musically similar *rāgas* that share a common set of *svaras* and have similar melodic phrases.

We also analyzed the influence of a dataset in terms of the number and selection of *rāgas* on the complexity of the *rāga* recognition task. For this analysis we selected the best performing method (M_{TDMS}). Our results indicated that this task is more challenging for audio recordings in Carnatic music as compared to Hindustani music. We showed that an increase in the number of *rāgas* in the dataset reduces the prediction accuracy of the method more severely in the case of Carnatic music than for Hindustani music. We also showed that for the same number of *rāgas* in a dataset, the selection of *rāgas* influences considerably the complexity of this task. We saw that the accuracy for the Carnatic music dataset can vary up to 10% across different selections of *rāgas* in the dataset. The insights obtained in this analysis will help to better interpret the performance of the existing methods, which are typically evaluated on different datasets. In the future, we would like to investigate the minimum duration of the audio recording required to reliably recognize *rāgas*. Another promising direction for the future work is to explore methodologies that can combine PCD-based, pattern-based, and **TDMS**-based methods to improve *rāga* recognition.

Chapter 7

Applications

7.1 Introduction

The research work presented in this thesis has several applications. A number of these applications have already been described in Section 1.1. While some applications such as rāga-based music retrieval and music discovery are relevant mainly in the context of large audio collections, there are several applications that can be developed on the scale of the music collection compiled in the CompMusic project. In this chapter, we present some concrete examples of such applications that have already incorporated parts of our work. We provide a brief overview of Dunya, a collection of the music corpora and software tools developed during the CompMusic project and, Sarāga and Riyāz, the mobile applications developed within the technology transfer project, Culture Aware MUsic Technologies (CAMUT). In addition, we present three web demos that showcase parts of the outcomes of our computational methods. We also briefly present one of our recent studies that performs musicologically motivated exploration of melodic structures in IAM. It serves as an example of how our methods can facilitate investigations in computational musicology.

7.2 Dunya

Dunya⁵⁵ comprises the music corpora and the software tools that have been developed as part of the CompMusic project. It includes data for five music traditions - Hindustani, Carnatic, Turkish Makam, Jingju and Andalusian music. By providing access to the gathered data (such as audio recordings and metadata), and the generated data (such as audio descriptors) in the project, Dunya aims to facilitate study and exploration of relevant musical aspects of different music repertoires. The CompMusic corpora mainly comprise audio recordings and complementary information that describes those recordings. This complementary information can be in the form of the

⁵⁵<http://dunya.compmusic.upf.edu/>

relevant metadata, and melody, rhythm, and structural descriptors that are computationally extracted from the audio recordings. The metadata for each recording is aggregated from multiple data sources: MusicBrainz, for the editorial metadata and, Wikipedia, for artists' biographies⁵⁶ and related information. For a detailed description of Dunya we refer to Porter et al. (2013).

To extract the music descriptors mentioned above, Dunya uses **Essentia** and a set of software packages developed for specific music traditions. Our work on tonic identification is already integrated in to **Essentia**, and thus, is available in Dunya. The implementation of our other methods for pattern discovery and *rāga* recognition will also be integrated into **Essentia**.

There are two modes in which Dunya provides access to the data mentioned above: a web-based interface, and a **RESTful API**. The web-based GUI⁵⁷ is mainly meant to browse through the music corpora, listen to the music recordings, and visualize the extracted music descriptors that are time-synchronized with the music. In a way, it provides a medium for enhanced music listening. In addition, for every recording, the associated metadata is also shown, which provides the surrounding context to better appreciate the music performance. A screenshot of the recording page for a music piece^{58,59} in Hindustani music is shown in Figure 7.1. We see that there are three main panes, the metadata pane towards the top-left, rhythm pane at the top and melody pane at the bottom. The metadata pane displays the editorial and the automatically generated metadata. In this pane, the main description of the melodic aspects of a performance is given by its tonic and the *rāga* label, indicated by the arrows numbered 1 and 5, respectively. The melody pane at the bottom displays a coarse timbral representation on top of which the predominant pitch contour is shown (indicated by arrow-3). In the same pane, the solid horizontal line (indicated by arrow-2) marks the tonic frequency of the recording. This frequency corresponds to the base svara Sa in the performance, and it acts as a reference to better interpret the frequency intervals in the continuous pitch contour. The second octave of the tonic frequency is also indicated by a dashed horizontal line. Along with the predominant pitch contour its histogram is also shown (arrow-4), which summarizes the pitch content in the entire performance.

While the web interface provides easy and quick access to the data, for a more comprehensive access mainly meant for researchers and developers, Dunya provides a **RESTful API**. Through the **API** it allows access to the audio recordings, gathered metadata and extracted music descriptors. This can be used to access the existing datasets, as well as to create new datasets using the CompMusic corpora. To further facil-

⁵⁶An example of a biography, https://en.wikipedia.org/wiki/T._M._Krishna

⁵⁷<http://dunya.compmusic.upf.edu/>

⁵⁸<http://dunya.compmusic.upf.edu/hindustani/recording/72df913b-ac52-4798-990d-72e04a64bd8c/raga-ragesri>

⁵⁹<http://musicbrainz.org/recording/72df913b-ac52-4798-990d-72e04a64bd8c>

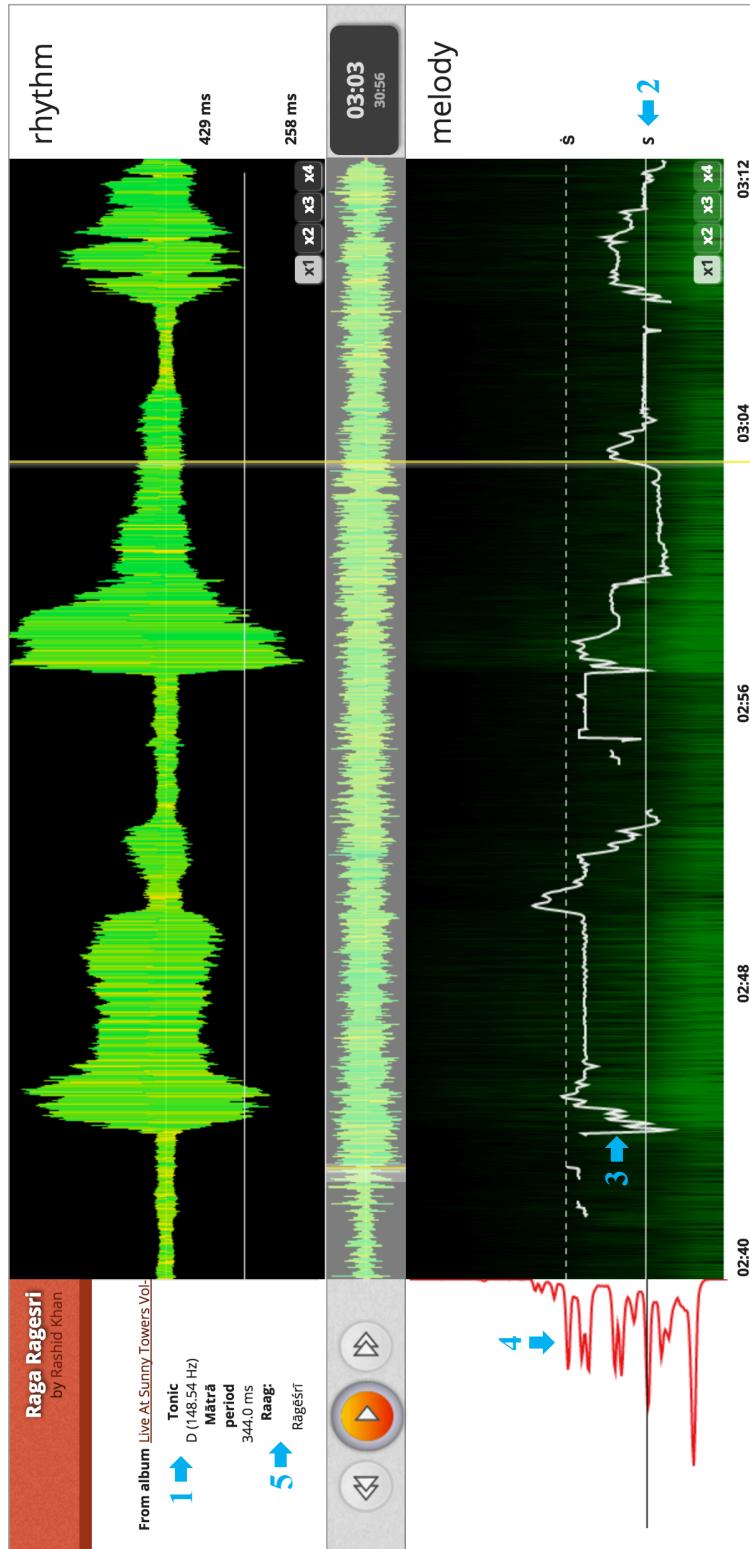


Figure 7.1: Screenshot of the recording page in Dunya showing extracted music descriptors and associated metadata.

itate the usage of this API, a Python⁶⁰ package, PyCompMusic⁶¹, is provided. Using this tool, with just a few lines of code, the entire corpora and associated metadata can be retrieved. An example usage of PyCompMusic is shown below.

```
In [1]: from compmusic import dunya as dn
In [2]: from compmusic.dunya import carnatic as ca
In [3]: dn.set_token("<dunya api token>")
In [4]: concerts = ca.get_concerts()
In [5]: len(concerts)
Out[5]: 328
In [6]: concerts[1]
Out[6]:
{u'mbid': u'c5d9d3bd-bc01-4104-b874-d55219bd0e54',
 u'title': u'Madrasil Margazhi 2006'}
In [7]: recs = ca.get_recordings()
In [8]: len(recs)
Out[8]: 3533
In [9]: recs[1]
Out[9]:
{u'mbid': u'01f863b7-46b4-44f5-b547-fcbaaf66348',
 u'title': u'Vetta Veli'}
```

This is an example of querying the metadata for the Carnatic music collection⁶² in the CompMusic corpora. The first step is to import the relevant modules provided in the PyCompMusic package (In [1] and [2]), followed by a user authentication, which requires a Dunya API token (In [3]). This token can be obtained by registering in to Dunya. We then query for a list of all 328 concerts in the Carnatic music collection (In [4]), and the information about each concert is returned in a dictionary (Out [6]). Subsequently, we also present a query to obtain a list of all 3533 recordings in the collection (In [7]), and show the structure of the response (Out [9]). Using the MBIDs of the concerts and the recordings we can obtain more information about those entities. The Dunya API provides access to the editorial metadata, the audio recordings, and the extracted audio features for all the music collections in the CompMusic corpora.

7.3 Mobile Applications: Sarāga and Riyāz

Sarāga and Riyāz are two mobile applications developed as a part of the CAMUT⁶³ project, which aims to explore the commercial exploitation of the technologies developed in the CompMusic project. These applications aim to foster learning, teaching and appreciation of Indian music forms. Both these applications incorporate parts

⁶⁰<https://www.python.org/>

⁶¹<https://github.com/MTG/pycompmusic>

⁶²<https://musicbrainz.org/collection/55412ad8-1b15-44d5-8dc8-9c3cb0cf9e5d>

⁶³<http://mtg.upf.edu/projects/camut>

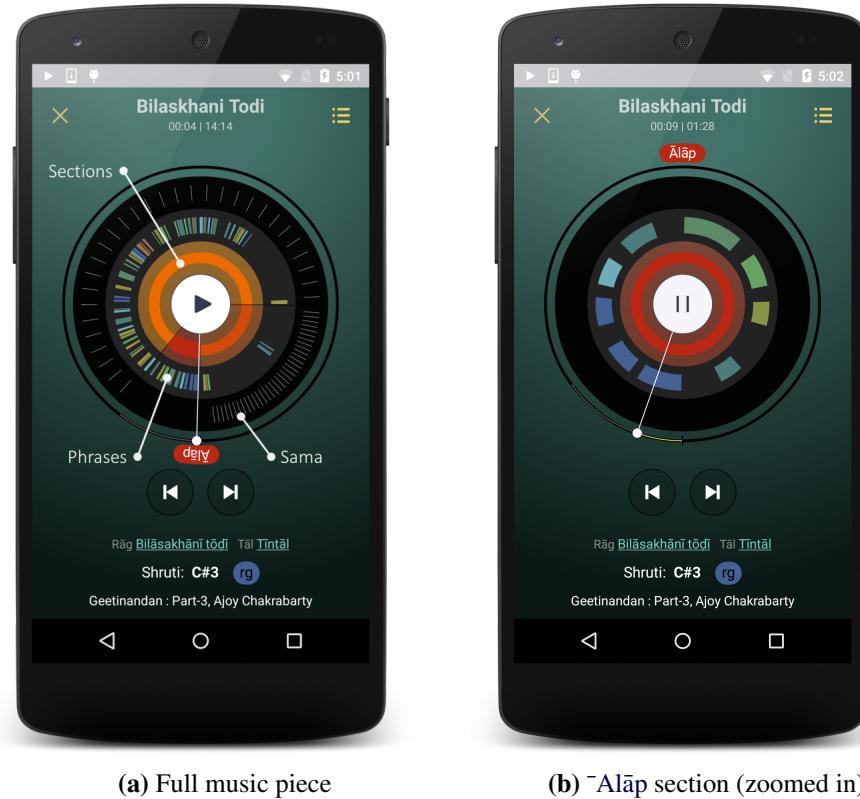


Figure 7.2: Screenshots of the mobile application, Sarāga, showing a music piece of Hindustani music. (a) shows the entire music piece, (b) shows the *Alāp* section in the piece. Melodic phrases are marked by the colored arches. Tonic (*śruti*) of the recording is shown at the bottom along with the current playing melodic phrase.

of the outcome of our research work presented in this thesis. We provide below a brief description of these applications.

7.3.0.1 Sarāga

Sarāga is a mobile application that provides an enriched listening atmosphere over a collection of Carnatic and Hindustani music that is released under the Creative Commons license⁶⁴. It is meant for music connoisseurs and students of these art music traditions to navigate, discover and listen to the music using culturally relevant concepts. Sarāga contains inclusive designing of innovative visualizations and inter and intra-song navigation interfaces that present musically rich information to the user in a compact way. The time synchronized visualizations of musically relevant facets such as melodic patterns, sama locations and sections enable better understanding and appreciation of these music traditions.

⁶⁴<https://creativecommons.org/>

In Figure 7.2, we show a screenshot of the recording screen in Sarāga playing a music piece⁶⁵ in Hindustani music. In Figure 7.2 (a), the entire music piece is visualized. Information regarding the sections, melodic patterns, and the *sama* locations is displayed through the concentric circles as indicated in the screenshot. As the playback advances in time along the circle, these descriptors are highlighted based on the current time. For example, in Figure 7.2 (a), the melodic phrase “rg” is being sung in the piece, which is highlighted at the bottom of the screen. Since music performances in IAM can last long (sometimes up to an hour), Sarāga interface allows to tap and zoom into a particular section. An example of this is shown in Figure 7.2 (b), wherein the ālāp section is selected. Furthermore, a user can also tap on a particular melodic pattern to go to a screen, where all the occurrences of that pattern in the recording are shown together. These functionalities facilitate a user to better understand the structures of different musical facets in a piece of IAM. In addition to these descriptors, there is accompanying data shown at the top and the bottom of the screen, which includes editorial metadata and śruti (tonic) information. The text strings corresponding to the musical concepts (*rāga* and *tāla*) are hyper-linked to their respective pages, where that music concept is described using a set of representative audio examples.

7.3.0.2 Riyāz

Riyāz is a mobile application that aims to facilitate music learning for students of IAM by making their practice sessions more efficient. This is achieved by simulating a student-teacher interaction in which learning happens through imitating the musical exercises built by professional musicians. The application performs singing assessment and provides a detailed feedback, wherein it highlights the mistakes and gives suggestions to improve. In Figure 7.3 (a), we show the main evaluation screen of Riyāz, where a user can visualize, in real-time, the pitch track and the computed svara score. After a practice session, a detailed feedback is provided as shown in Figure 7.3 (b). Note that, Riyāz is a work in progress with a prototype version already available at the time of writing this thesis. Currently there are 1500+ users with 20,000+ user sessions.

7.4 Demos

We now proceed to present two elementary web-based applications that demonstrate the outcome of our pattern discovery approach. In addition, we also present Rāgawise, a prototype web application for real-time *rāga* recognition.

Demo1: Melodic Patten Discovery

One of the motivations behind performing pattern discovery is to explore novel relationships in the data and extract new knowledge. However, it is a challenging task to

⁶⁵<https://musicbrainz.org/recording/3124479b-5118-4cf3-823f-8fefad45e586>



Figure 7.3: Screenshots of the mobile application, Riyāz.

evaluate these aspects in a quantitative evaluation setup, wherein the available ground-truth (typically comprising annotations from an expert) is used as the basis to measure the quality of the output. It becomes even more challenging when these unsupervised analyses are performed on large datasets, such as in our case, in which obtaining the ground-truth becomes unfeasible. In the study presented in Section 5.4, we performed melodic pattern discovery in audio collections of Carnatic music comprising nearly 365 hours of music. Though we performed a quantitative evaluation using a randomly sampled subset of the output, the evaluation numbers do not convey much in terms of the musical novelty of the patterns.

In order to facilitate informal evaluations, take feedback from musicians, identify novel outcomes, and eventually, to improve the quality of the output of our pattern discovery methods we built two web-based applications that allow access to the discovered melodic patterns in intuitive ways.

Our first demo enables a user to browse through all the melodic patterns discovered in the study reported in Section 5.4. Since determining a meaningful similarity threshold

Search results using chosen seed from Nenendu Vedakudura

Similarity	Seed_Id	Start(s)	End(s)	Pair_Id	Start(s)	End(s)	Musicbrainz ID (searched file)	Distance
✓	15757285	33.0	35.0	15757335	37.7	39.9	46848888-11b4-4f27-b6d0-a64ee0odf68e	1330.01
✓	15757285	33.0	35.0	15757339	36.5	38.9	14f9192e-8f5d-4945-9362-05ba1b9fa4b2	1900.42
✓	15757285	33.0	35.0	15757341	897.6	899.9	e3f8bf6d-3cae-4036-a625-d9b8dacef5dc	2010.18
✓	15757285	33.0	35.0	15757337	890.9	893.0	170970da-a19a-4f2d-8dae-4ece61af2780	2041.45
✓	15757285	33.0	35.0	15757338	890.9	893.0	170970da-a19a-4f2d-8dae-4ece61af2780	2041.45
✓	15757285	33.0	35.0	15757336	1089.0	1091.2	60e79cac-afdo-4b3f-ab90-6c925583a60c	2109.13
✓	15757285	33.0	35.0	15757340	576.6	579.0	05ccb6d7-c281-4c16-84f7-1ecede881a38	2158.92
✓	15757285	33.0	35.0	15757343	584.1	586.2	6fb47e57-cdbe-4538-b81c-fd5e43caa64b	2174.02
✓	15757285	33.0	35.0	15757342	380.6	382.9	afa86ff8-0dee-42e9-a5a1-8f62efcd77f	2274.83
✓	15757285	33.0	35.0	15757347	662.7	664.9	170970da-a19a-4f2d-8dae-4ece61af2780	2285.69
✓	15757285	33.0	35.0	15757348	662.7	664.9	170970da-a19a-4f2d-8dae-4ece61af2780	2285.69
✓	15757285	33.0	35.0	15757344	448.0	450.6	41a138f7-59a4-4ef9-ae37-5bf0fd097269c	2298.17
✓	15757285	33.0	35.0	15757345	169.4	171.7	e1a74ccb-655e-4e37-b828-d40f3c1a15ec	2345.79
✓	15757285	33.0	35.0	15757346	516.6	518.7	60e79cac-afdo-4b3f-ab90-6c925583a60c	2350.09
✓	15757285	33.0	35.0	15757356	651.1	653.4	521113d7-9f91-4adc-9445-7de665d03659	2395.79
✓	15757285	33.0	35.0	15757366	419.4	421.8	60e79cac-afdo-4b3f-ab90-6c925583a60c	2395.95

Figure 7.4: Screenshot of a web demo for navigating through the discovered melodic patterns organized by artists, releases and recordings.

is in itself a challenging task, we selected a fixed number of closest pattern matches in the output. To recall, 25 closest seed pattern pairs were selected within a recording, and for each seed pattern its 200 closest neighboring patterns were selected across the entire music collection (Section 5.4.1). This resulted in around 15 million patterns for our audio collection that comprises 1764 recordings. This demo allows us to navigate through all the recordings in the collection, fetch all the seed patterns for a recording, and for each seed pattern, fetch their closest patterns from the entire music collection. In Figure 7.4, we show a screenshot of a page where a list of the closest melodic patterns for a seed pattern is displayed. All the listed melodic patterns can be played and listened to. From the column that specifies the MBIDs of the recordings of the retrieved patterns, we readily notice that these patterns are from different recordings (Figure 7.4). These recordings are from different artists with different tonic pitches, and even across vocal and instrumental music. With such an interface, it becomes easy to assess the quality of the discovered melodic patterns and to identify musically novel patterns.

The demo described above presents the melodic patterns by structuring them in a hierarchical manner (according to the editorial metadata of the collection), which is useful to look for patterns in a particular music piece or by a particular artist. An alternate way to present these patterns and their relationships is through a network visualization. Such a visualization makes it easier to identify interesting and novel relationships between different music pieces, artists, and *rāgas*. In Figure 7.5, we show a screenshot of our second demo that presents a network visualization of the discovered melodic patterns. These melodic patterns are the ones obtained in the study described in Section 5.5. Note that, in this study we employed a network analysis to

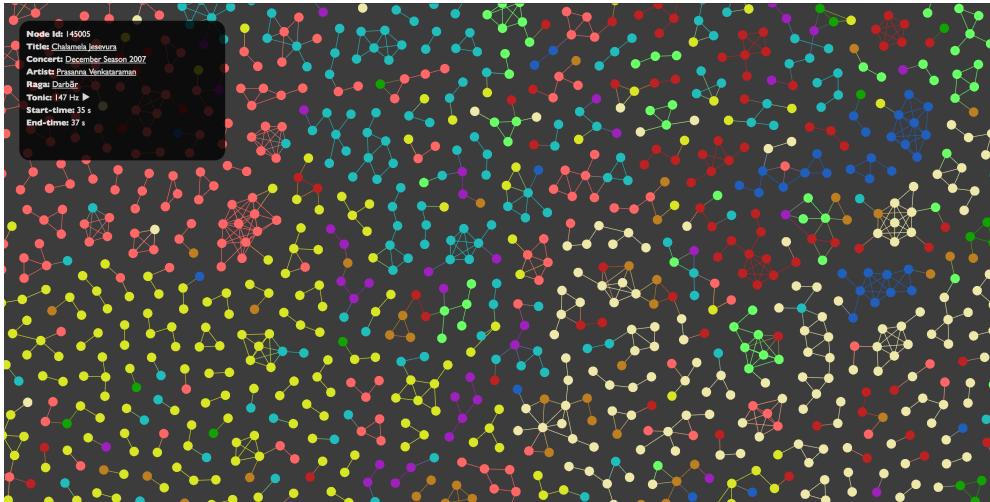


Figure 7.5: Screenshot of a web demo of the network of the discovered melodic patterns. Colors indicate different rāgas.

also determine a melodic similarity threshold, as a result of which we only retain the musically meaningful connections between the melodic patterns. The nodes of the network are the melodic patterns and the edges represent a binary melodic similarity between the nodes. In addition, for each node we also provide the accompanying information of the audio recording from which it is extracted (Figure 7.5, top-left corner). Furthermore, we also provide an option to play a tone that corresponds to the tonic of the recording. This helps to establish the tonal context of the melodic pattern. Both the web demos described above provide useful insights into the outcome of our pattern discovery approach. They are made available online (Appendix B).

Rāgawise: Real-time Raga Recognition

In Chapter 6, we described our computational approaches for rāga recognition. Our objective was to develop novel methods to obtain a rāga label for a recorded music performance. As mentioned earlier, there are several applications of these methods such as automatic rāga annotation of large audio archives, rāga-based music retrieval, establishing meaningful similarity measures across recordings and music pedagogy. In this section we present a prototype web application, **Rāgawise**, which demonstrates the usability of such systems in the context of music pedagogy.

Rāgawise is a real-time rāga recognition prototype system (Gulati et al., 2015a). It uses PCP, pitch transitions and melodic patterns to recognize rāga from an incoming audio stream. For each rāga, it stores a dictionary of the svaras, svara transitions, and typical melodic patterns. It processes the input vocals in real-time to estimate pitch, and subsequently performs melody transcription. The likelihood of each rāga

is updated in real-time based on the identified *rāga* elements in the melody. In order to highlight the melodic events that are characteristic of a *rāga*, a dynamic visualization of the evolution of the likelihood of all the *rāgas* is performed. We present a screenshot of *Rāgawise* in Figure 7.6, where different panels are marked by blue arrows. In the top panel (arrow 1) we display the transcribed melody symbols (*svaras* in this case) using the pitch track obtained in real-time (arrow 2). We continuously process the transcribed melody to detect the presence of different melodic elements, and once detected, we highlight the associated *rāgas* (arrow 3,4,5). We also compute a cumulative salience score of each *rāga* based on the frequency of the detected melodic elements (arrow 6).

Note that, building *Rāgawise* was a team effort done as a part of the Hackday, HAMR, in ISMIR-2015. It was done in collaboration with Kaustuv Kanti Ganguli, Swapnil Gupta and Ajay Srinivasamurthy.

7.5 Computational Musicology

The research work presented in this thesis paves the way for a number of studies in the context of computational musicology in *IAM*, which can be performed at a large scale using the corpora of recorded performances. We provide here an example of our preliminary work in this direction.

As mentioned, *IAM* is primarily an improvisatory music tradition, wherein compositions merely serve as the skeleton of a performance. However, at the same time, melodies are constructed in adherence with the *rāga* grammar, and therefore, they are bound by certain rules (or conventions). This fine distinction between what is ‘fixed’ and what remains ‘free’ is not clearly defined and is implicitly acquired by music students through years of musical training. With this as our motivation, in Ganguli et al. (2016), we propose methods to discover melodic structures in recorded performances of *IAM*. In this study, we utilized several outcomes of the methods described in this thesis such as melody representations, tonic normalization and *nyās* segmentation. We studied five different aspects related with the temporal evolution of melody in a music piece: the overall coarse evolution of melody in time, the transition characteristics of consecutive *nyās svaras*, the relationship between the functional roles of the *svaras* and their duration in a melody, the duration and position of *svaras* in a melody, and the presence of a pulsation in *ālāp* melodies.

One of the interesting results from this study is related to the overall evolution of melody in a music piece. We found that, irrespective of the functional roles of the *svaras*, they are explored in melodies in a linear manner by an artist, starting from the lowest frequency *svara* going all the way to the highest. Interestingly, this trend appears to be largely independent of the duration of the performance. In Figure 7.7, we illustrate this pattern by showing the trajectories of the highest salience bin in a

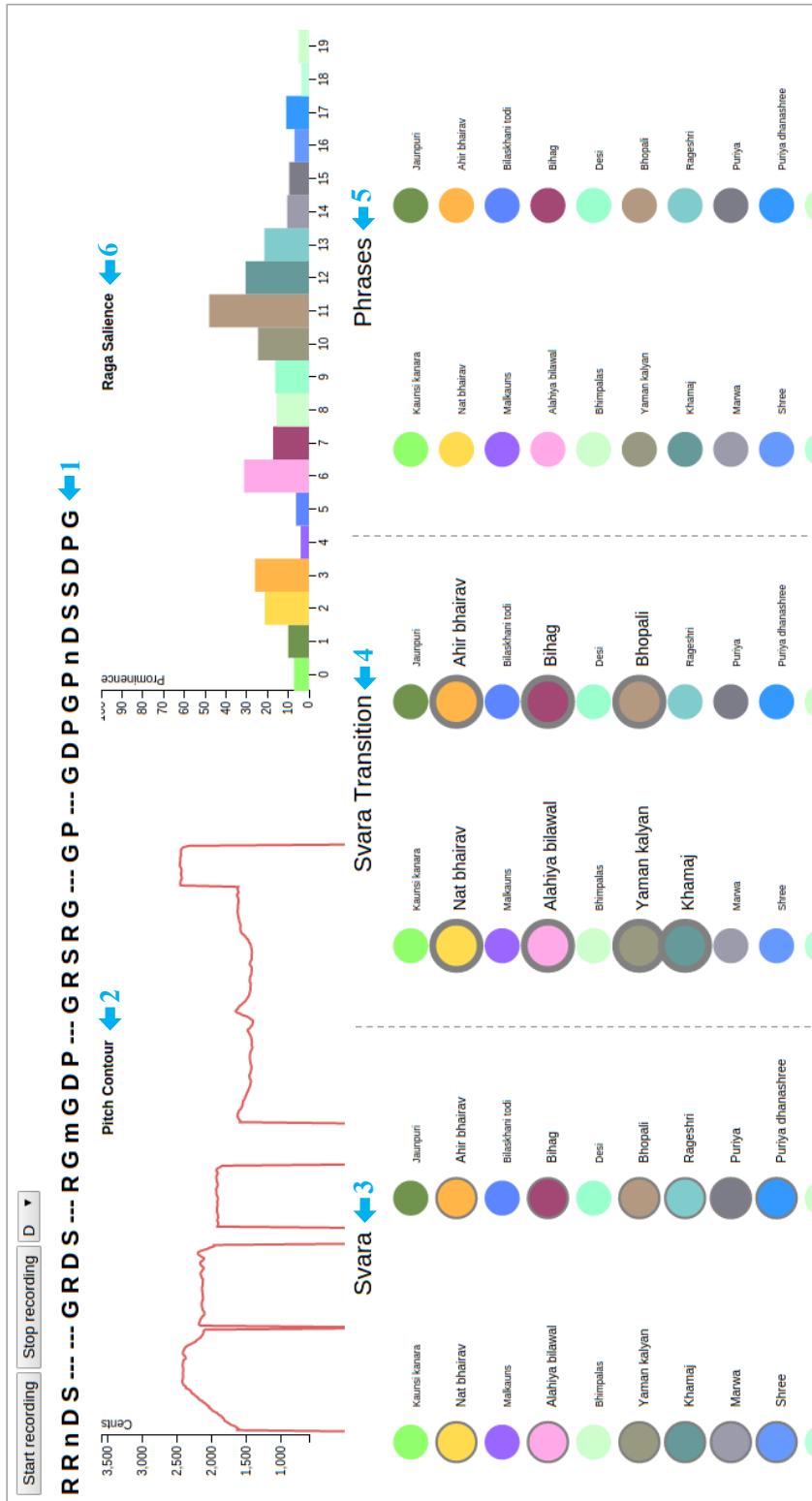


Figure 7.6: Screenshot of Rāgawise, which illustrates real-time pitch tracking, melody transcription and rāga salience evolution. Bottom panel shows a list of ragas for each melodic element (svara, svara transition and melodic patterns). Ragas for which a particular melodic element is detected in the audio stream are highlighted.

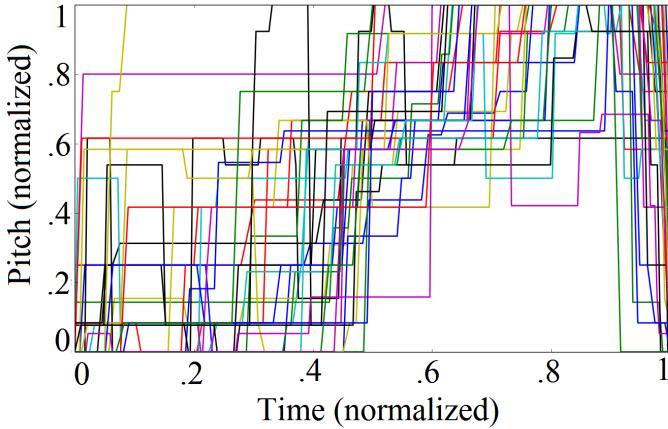


Figure 7.7: Trajectories of the highest salience bin in long-time averaged pitch histograms computed across breath-phrases for 37 performances. The length of the performances are normalized.

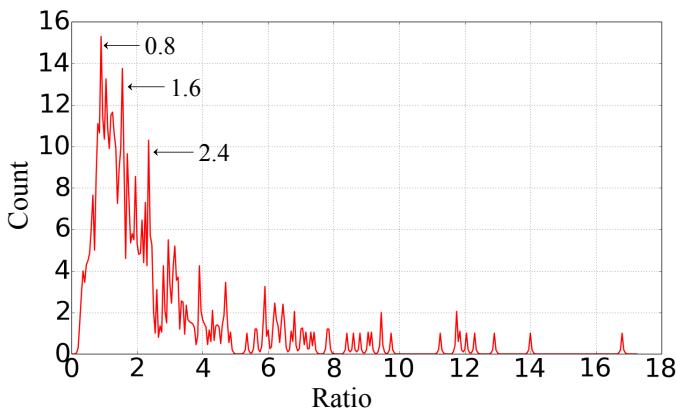


Figure 7.8: Histogram of the ratio of the inter-onset-intervals of salient svaras across breath-phrases.

long-time averaged pitch histogram computed across breath-phrases⁶⁶.

Another interesting result is that although the *ālāp* section is unmetered, a histogram of the ratio of the inter-onset-intervals between the salient svaras across breath-phrases shows strong pulsation (Figure 7.8). This corroborates with the existing musicological discussion that there is indeed a pulsation in *ālāp* melodies. For a detailed description of the methodology used in the analysis and comprehensive results of the study, we refer to Ganguli et al. (2016).

⁶⁶Segment of the melody between two breath pauses (>300 ms) by a singer

7.6 Summary

In this chapter, we presented some concrete examples of different applications of our work in several contexts such as computational musicology, enhanced music listening applications, and pedagogical tools for **IAM**. We presented **Dunya**, a research tool that comprises all the data and software tools developed in the CompMusic project. We briefly described two different ways to access the research corpora compiled in the project. Using this tool and the compiled corpora, a number of datasets can be created to study different aspects of **IAM**. Subsequently, we presented web-based demos that enable a user to browse and listen to the melodic patterns discovered by our approach through an intuitive user interface. Listening to and analyzing the relationships between these melodic patterns provides useful insights, which can be looped back to further improve our system. We also presented a web prototype of **Rāgawise** that demonstrates the utility of an automatic *rāga* recognition system in a real-time setup, which can further be exploited to develop tools to facilitate pedagogy in **IAM**. We presented two mobile applications (**Sarāga** and **Riyāz**) that demonstrate the utility of our work in the context of enhanced music listening experience and music pedagogy. We also presented our preliminary study as an example of how our work can be utilized in computational musicology. Overall, we see that our work has several interesting applications in a wide variety of contexts.

Chapter 8

Summary and Future Perspectives

8.1 Introduction

In this thesis, we have presented a number of computational approaches for analyzing melodic elements at different levels of melodic organization in **IAM**. In tandem, these approaches generate a high-level melodic description of the audio collections in this music tradition. They build upon each other to finally allow us to achieve the goals that we set at the beginning of this thesis, which are:

- To curate and structure representative music corpora of **IAM** that comprise audio recordings and the associated metadata, and use that to compile sizable and well annotated tests datasets for melodic analyses.
- To develop data-driven computational approaches for the discovery and characterization of musically relevant melodic patterns in sizable audio collections of **IAM**
- To devise computational approaches for automatically recognizing *rāgas* in recorded performances of **IAM**.

Based on the results presented in this thesis, we can now say that our goals are successfully met. We started this thesis by presenting our motivation behind the analysis and description of melodies in **IAM**, highlighting the opportunities and challenges that this music tradition offers within the context of music information research and the CompMusic project (Chapter 1). We provided a brief introduction to **IAM** and its melodic organization, and critically reviewed the existing literature on related topics within the context of **MIR** and **IAM** (Chapter 2).

In Chapter 3, we provided a comprehensive overview of the music corpora and datasets that were compiled and structured in our work as a part of the CompMusic project. To the best of our knowledge, these corpora comprise the largest audio collections of **IAM** along with curated metadata and automatically extracted music

descriptors that is available for research. Furthermore, the datasets that we built from these corpora allowed us to successfully evaluate our computational approaches, with some of them being the largest datasets ever used for such evaluations. We described and evaluated the approaches we followed to obtain different melodic descriptors and melody representations with which we perform melodic analysis (Chapter 4).

In Chapter 5, we argued that the potential of a pattern-based melodic analysis in **IAM** can be exploited using an unsupervised approach to extract melodic patterns from audio recordings. We demonstrated that using our novel approach we can successfully discover musically significant melodic patterns in hundreds of hours of audio collections of **IAM**. Importantly, our approach does not require any exemplars of melodic patterns from experts as input. In Chapter 6, we described two novel approaches for **rāga** recognition that jointly utilize the tonal and the temporal characteristics of melodies in **IAM** without discretization of the melody representation. We demonstrated that our approach can effectively utilize the discovered melodic patterns for **rāga** recognition. We also presented our approach that abstracts a continuous melody representation to capture the melodic outline relevant for characterizing **rāgas**. Using this approach, we demonstrated unprecedented accuracies in the task of **rāga** recognition using the largest datasets ever used for this task.

We note that the approaches we propose to perform these tasks are not 100% accurate, and there exists a large scope for improvement. However, we have seen that a majority of the errors made by our approaches can be explained from a musicological and perceptual perspective. The cases where the system fails are often the ones which are challenging even for a human listener.

Note that, here we have only provided an overall summary of the thesis. The key results and conclusions of the work presented in each chapter are summarized at the end of the chapter itself. In the subsequent section, we enumerate our main contributions (Section 8.2), and finally, end the thesis with discussions about the future perspectives (Section 8.3).

8.2 Summary of Contributions

We now present a summary of the main contributions of this thesis.

Contributions to Creating Music Corpora and Datasets

One of the objectives of the CompMusic project was to build a high quality research corpora of **IAM**, with which to study different computational tasks in the context of **MIR**. The task of compiling and curating the research corpora and different test datasets has mainly been a team effort. We describe below some specific contributions from the author.

- The contributions to compiling corpora are mainly in Hindustani music corpus,

starting from the procurement of music CDs, ripping and structuring the audio collection and manually adding all the editorial metadata to MusicBrainz (Section 3.2).

- Compiling and annotating CompMusic Tonic Datasets (TID_{CM1} , TID_{CM2} and TID_{CM3}), which collectively include tonic annotations for 716 audio recordings spanning 168 hours of audio (Section 3.3).
- Compiling and annotating a Nyās Dataset (NDD_{CM}), which includes annotations of nyās segments for 20 audio recordings spanning 1.5 hours of audio, done in collaboration with Kaustuv Kanti Ganguli (Section 3.3.2).
- Improving over the existing Melodic Similarity Dataset (MSD), which originally included 497 annotated instances of 10 different melodic patterns in 33 audio recordings (Ishwar et al., 2013; Ross et al., 2012). In the revised version after a detailed verification, 127 new instances of melodic patterns were added. This is done in collaboration with Kaustuv Kanti Ganguli and Vignesh Ishwar (Section 3.3.3).
- Building Rāga Recognition Datasets for Carnatic music (RRD_{CMD}), and Hindustani music (RRD_{HMD}), which contain rāga labels and the associated metadata. RRD_{CMD} comprises 480 recordings in 40 rāgas spanning 124.5 hours of audio. RRD_{HMD} comprises 300 recordings in 30 rāgas spanning 116.2 hours of audio. To date, these are the largest datasets ever built for studying this task, and we make them publicly available. These datasets are built in collaboration with Vignesh Ishwar and Kaustuv Kanti Ganguli (Section 3.3.4).

Scientific Contributions

- Review of the current approaches for tonic identification, melodic pattern processing and rāga recognition in the context of MIR in IAM, emphasizing their limitations and identifying avenues for scientific contributions (Section 2.4).
- Comprehensive assessment of different tonic identification approaches on a number of sizable datasets, along with a detailed error analysis for different types of music material (Section 4.2).
- Development of a novel nyās landmark-based approach for the segmentation of melodies in Hindustani music. Nyās detection is addressed for the first time from a computational perspective (Section 4.5).
- In depth evaluation of different procedures and parameter settings for computing melodic similarity in the context of short-duration rāga motifs in IAM (Section 5.2).

- A partial transcription and complexity weighting-based approach for improving melodic similarity that exploits the presence of long held *svaras* and *gamakas* in melodies of Hindustani and Carnatic music, respectively (Section 5.3).
- Demonstration of the utility of an unsupervised approach for discovering repeated melodic patterns in sizable audio collections of **IAM** (Section 5.4).
- Characterization of the discovered melodic patterns by employing network analysis tools to identify musically significant patterns, the *rāga* motifs (Section 5.5).
- Development of a novel pattern-based approach for *rāga* recognition, which employs vector space modeling concepts to exploit the discovered melodic patterns for this task (Section 6.2).
- Development of a novel feature, the time delayed melodic surface (TDMS), which captures both the tonal and the short-time temporal characteristics of a melody. This feature together with a simple 1-nearest neighbor classification strategy is shown to outperform the state of the art in *rāga* recognition using the largest datasets ever used for this task (Section 6.3).

Technical Contributions

- Building a web-based demo for navigating through the melodic patterns discovered by our approach organized by artists, releases and recordings (see Figure 7.4).
- Building a web-based demo for visualizing relationships between different audio recordings based on the similarity between the constituent melodic patterns. The relationships are represented in the form of a network of melodic patterns (see Figure 7.5).
- Building a web prototype of a real-time *rāga* recognition system, *Rāgawise*. This work is done in collaboration with Kaustuv K. Ganguli, Swapnil Gupta and Ajay Srinivasamurthy (Section 7.4).
- Developing two mobile applications: *Sarāga*, which provides enhanced listening experience, and *Riyāz*, which facilitates pedagogy in the context of **IAM**. It is a team effort, with author's main contributions in the conceptualization and design of the applications, as well as in the implementation of some of the components (Section 7.3).
- Implementation of our tonic identification algorithm (M_{JS}) in *Essentia*, an audio feature extraction library.

The web links to access the demos and the applications mentioned above are provided in Appendix B.

Most of the outcomes of the work presented in this document have been published in the form of papers in international conferences and journals. The full list of the author’s publications is provided in Appendix A. The compiled music corpora, and the code and tools developed during our work are made publicly available to facilitate reproducible research and comparative studies (Appendix B). The set of tools and output of our approaches are also integrated in Dunya (Section 7.2).

8.3 Future Perspectives

To the best of our knowledge, this is the first time that melodies in **IAM** are computationally analyzed on corpora comprising hundreds of hours of audio recordings. This opens up several unexplored research problems that can be addressed by utilizing the results and resources presented in this thesis. In addition, there are several ways to further improve the methodologies proposed in our work. In this section, we discuss a number of these future directions.

We start by enumerating different avenues for improvement in the pattern processing methodologies discussed in our work. In Chapter 5, we argued that the potential of pattern-based analysis and description of melodic aspects in **IAM** can be exploited by going beyond supervised methodologies for pattern processing. We successfully demonstrated the effectiveness and utility of an unsupervised approach for discovering melodic patterns. However, the lack of a quantitative assessment of such an approach limits its improvement. We believe that a balanced combination of both supervised and unsupervised methodologies would take pattern processing in **IAM** to the next level. One of the ways in which both these approaches can be combined is by using the output of the unsupervised approach for facilitating annotations of large amounts of melodic patterns, which is one of the biggest limitations of the supervised approaches as mentioned in Section 5.1. A possible way to achieve this is to mark every melodic pattern pair (the output of our approach) as melodically similar or dissimilar. Since our method produces millions of such melodic pattern pairs in the sorted order of their melodic similarity, this can readily lead to a large corpus of melodic patterns annotated in terms of the melodic similarity between them.

With sizable annotated datasets, comprising thousands of melodic patterns across different artists, compositions, forms, and *rāgas*, several valuable insights into perception of melodic similarity can be obtained. We provide some concrete examples of such analyses here. One of our learnings while working on the melodic similarity is that not every sample in the string representation of a melodic pattern contributes equally in establishing similarity. There are specific regions and characteristic pitch movements in melodic patterns that are more important and often become the anchor points in determining similarity. A sizable annotated dataset of melodic patterns can facilitate such investigations, which in turn would improve models for computing melodic similarity. Another interesting and important aspect to explore in pattern discovery is to take into account the local melodic context of a pattern. Since melodies

in IAM are constructed in accordance with the rāga grammar, rāga motifs might bear a strong correlation with their melodic context, which can be exploited to improve pattern discovery and also reduce its computational complexity. In addition, determining possible relations between the *sama* locations (downbeat in rhythm cycle) and the locations of rāga motif is also an interesting subject of future investigations. All these analyses can benefit immensely from a well annotated sizable dataset of melodic patterns.

In our work, we considered the predominant pitch in audio as the low-level melody representation. However, as illustrated by an example in Section 4.3.1, timbral and loudness dimensions of melody also influence melodic similarity. Therefore, a possible future investigation is to find ways to incorporate these dimensions in the representation of melodies.

Addressing issues related with computational complexity is fundamental in pattern processing tasks. There are several strategies to make this process more computationally efficient. One of the ways is to devise an enriched melody transcription approach that can parametrically encode different types of melodic ornaments, gamakas and other melodic atomic units (for instance, Widess (1994); Rao et al. (1999)). Such a melody representation will not only make pattern processing computationally less demanding, but can also help improve melodic similarity as melodic elements can be appropriately weighted.

A processing step that is crucial in melodic analysis and is unexplored from a computational perspective in the context of IAM is melody segmentation. We showed in Section 5.2.3 that a meaningful segmentation can tremendously improve melodic similarity computation. However, despite its importance, there are very few studies that address this task. This can be attributed to the difficulties involved in its formulation, specifically in a continuous predominant pitch representation of melody. One of the ways in which segmentation is studied is through music parallelism (Cambouropoulos, 2006; Rodríguez L. et al., 2014). Since the boundaries of repeated melodic patterns can indicate possible boundaries of segmentation, the discovered melodic patterns from our approach can be utilized to study the task of melody segmentation.

One main limitation of our proposed approach for pattern discovery is that it works with fixed duration patterns. This is mainly due to the constraints imposed by the DTW lower-bounding techniques, and also due to the unavailability of reliable melody segmentation approaches. However, once the patterns are discovered, their boundaries can be refined. This can be done by collectively analyzing closely related melodic patterns by extending their boundaries in both directions using a dynamic programming approach similar to Muscariello et al. (2009).

We mentioned in Section 5.5 that several communities in melodic pattern network that comprise a large number of nodes often correspond to the *gamaka* type patterns. In several melodic analyses such patterns might not be useful. In such cases, *gamaka* type patterns can be filtered out during the pre-processing stage, similar to the fil-

tering step we perform for removing pattern candidates that comprise only a single *svara*. This would lead to more number of musically relevant melodic patterns in the output of the pattern discovery system. In addition, this procedure will also reduce the computational complexity of the system.

We now proceed to provide some future directions of research in the context of automatic *rāga* recognition. We saw in Section 6.3.3 that the accuracies of existing approaches are around 90%. These evaluations are performed using full length audio recordings. One of the next steps is to analyze the minimum duration of the audio recording needed to perform this task (Balkwill & Thompson (1999)). Such an investigation would provide useful insights into the applicability of these approaches in the context of real-time *rāga* recognition. In our recent study in Ganguli et al. (2016) we found that different *svaras* of a *rāga* in a melody are explored linearly over the course of the entire music performance. This implies that a segment of audio recording taken from the start might not contain all the *svaras* in the *rāga*, which can severely deteriorate the performance of the *PCD*-based approaches. This aspect of the minimum melodic material required for a reliable *rāga* recognition can be explored in the future.

Existing approaches for *rāga* recognition mainly capitalize on a single type of melodic characteristics (Table 2.3). A possible direction is to combine multiple approaches for *rāga* recognition that utilize different types of information about melody. Our results show that the errors made by *PCD*-based approach *M_{PC}* and phase based approach *M_{VSM}* are complementary. Thus, they can potentially be combined to improve performance. In addition, a hierarchical model that combines these approaches based on the importance of the melodic features can be a promising methodology for *rāga* recognition in the future.

We now proceed to enumerate some novel research problems in the context of *IAM* that can utilize the output of our research work. Being an improvisatory music tradition, it is interesting to study the influence of a teacher's style on the stylistic nuances of their students and other artists. Since melodic patterns act as the basis for constructing melodies, such analyses can benefit immensely from the melodic patterns discovered using our approach. In addition, using melodic patterns one can perform a corpora level characterization of compositions and *rāgas*. Such a characterization will, in turn, help define novel ways to establish similarity measures between artists, compositions and *rāga*.

The improvisatory nature of *IAM* also makes the study of the temporal evolution of melody in a music piece a relevant topic of research. This can be achieved through an intra-recording analysis based on melodic patterns, such as analyzing the degree of pitch and timing transformations across the occurrences of melodic patterns, the time difference between the adjacent recurrences of a melodic pattern and the chronology of the occurrence of melodic patterns. Such insights will help understand the nature of improvisatory *rāga* rendition in general and the related artist specific stylistic nuances.

Analysis of different melodic elements presented in our work can be combined within an ontological framework to build a knowledge base of these melodic aspects in IAM. This in turn can be utilized to develop applications that perform high-level musical-logical queries. Such a system can also enable a number of applications that address pedagogical needs in the context of IAM. These applications can provide an objective description of the melodic structures in recorded performances such as, the overall progression of melody in a piece, time-synchronized display of different melodic units, and the relationship between these units across artists, rāgas, and compositions. Specifically in the context of IAM, where the music nuances are learned implicitly through years of training, such a description of melodic aspects can be immensely helpful to music students in learning from the recorded performances of maestros.

Appendix A

Publications by the Author

We here provide a list of publications by the author related with the thesis work. For the publications where the role is not that of the first author we also specify the contributions. Wherever not specified, the contributions are mainly in the formulation of the research problem, building the dataset, writing the code, performing the experiments, analyzing the results and writing the manuscript.

Peer-reviewed journals

- **Gulati, S.**, Bellur, A., Salamon, J., Ranjani, H. G., Ishwar, V., Murthy, H. A., & Serra, X. (2014). Automatic tonic identification in Indian art music: approaches and evaluation. *Journal of New Music Research*, 43(1), 53–71.
(Companion webpage: <http://compmusic.upf.edu/node/323>)
- Koduri, G. K., **Gulati, S.**, Rao, P., & Serra, X. (2012). Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4), 337–350.
(Formulating methodology, Writing parts of the code)

Full articles in peer-reviewed conferences

- **Gulati, S.**, Serrà, J., Ganguli, K. K., Şentürk, S., & Serra, X. (2016). Time-delayed melody surfaces for rāga recognition. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 751–757. New York, USA.
(Companion webpage: <http://compmusic.upf.edu/node/300>)
- Ganguli, K. K., **Gulati, S.**, Serra, X., & Rao, P. (2016). Data-driven exploration of melodic structures in Hindustani music. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 605–611. New York, USA.
(Formulation of the problem, building dataset, writing the code, writing the manuscript.)

- **Gulati, S.**, Serrà, J., Ishwar, V., Şentürk, S., & Serra, X. (2016). Phrase-based rāga recognition using vector space modeling. In *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 66–70. Shanghai, China.
(Companion webpage: <http://compmusic.upf.edu/node/278>)
- **Gulati, S.**, Serrà, J., Ishwar, V., & Serra, X. (2016). Discovering rāga motifs by characterizing communities in networks of melodic patterns. In *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 286–290. Shanghai, China.
(Companion webpage: <http://compmusic.upf.edu/node/277>)
- **Gulati, S.**, Serrà, J., & Serra, X. (2015). Improving melodic similarity in Indian art music using culture-specific melodic characteristics. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 680–686. Málaga, Spain.
(Companion webpage: <http://compmusic.upf.edu/node/269>)
- **Gulati, S.**, Serrà, J., & Serra, X. (2015). 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)*, pp. 678–682. Brisbane, Australia.
(Companion webpage: <http://compmusic.upf.edu/node/242>)
- **Gulati, S.**, Serrà, J., Ishwar, V., & Serra, X. (2014). Mining melodic patterns in large audio collections of Indian art music. In *Proceedings of the International Conference on Signal Image Technology & Internet Based Systems (SITIS-MIRA)*, pp. 264–271. Marrakesh, Morocco.
(Companion webpage: <http://compmusic.upf.edu/node/210>)
- **Gulati, S.**, Serrà, J., Ganguli, K. K., & Serra, X. (2014). Landmark detection in Hindustani music melodies. In *Proceedings of the International Computer Music Conference / Sound and Music Computing Conference (ICMC-SMC)*, pp. 1062–1068. Athens, Greece.
(Companion webpage: <http://compmusic.upf.edu/node/324>)
- Srinivasamurthy, A., Koduri, G. K., **Gulati, S.**, Ishwar, V., & Serra, X. (2014). Corpora for Music Information Research in Indian Art Music. In *Proceedings of Joint International Computer Music Conference/Sound and Music Computing Conference*, pp. 1029–1036. Athens, Greece.
(Compilation of the research corpora. Companion webpage: <http://compmusic.upf.edu/smcc-2014-corpora>)
- Şentürk, S., **Gulati, S.**, & Serra, X. (2014). Towards alignment of score and audio recordings of Ottoman-Turkish makam music. In *Proceedings of the 4th International Workshop on Folk Music Analysis (FMA)*. Istanbul, Turkey.
(Conceptualization, ideas and discussions)

- Bogdanov, D., Wack, N., Gómez, E., **Gulati, S.**, Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & Serra, X. (2013). Essentia: an audio analysis library for music information retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 493–498. Curitiba, Brazil.
(Implementation of the tonic identification method in Essentia)
- Bogdanov, D., Wack, N., Gómez, E., **Gulati, S.**, Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & Serra, X. (2013). ESSENTIA: an open-source library for sound and music analysis. In *Proceedings of the 21st ACM international conference on Multimedia*, pp. 855–858. Barcelona, Spain.
(Implementation of the tonic identification method in Essentia)
- Şentürk, S., **Gulati, S.**, & Serra, X. (2013). Score informed tonic identification for makam music of turkey. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 175–180. Curitiba, Brazil.
(Conceptualization, ideas and discussions)
- Sordo, M., Koduri, G. K., Şentürk, S., **Gulati, S.**, & Serra, X. (2012). A musically aware system for browsing and interacting with audio music collections. In *Proceedings of the 2nd CompMusic Workshop*. Istanbul, Turkey.
(Compilation of the research corpora)
- Koduri, G. K., **Gulati, S.**, & Rao, P. (2011). A survey of raaga recognition techniques and improvements to the state-of-the-art. In *Proceedings of the Sound and Music Computing Conference (SMC)*. Padova, Italy.
(Formulating methodology, Writing parts of the code)

Other contributions to conferences

- **Gulati, S.**, Serrà, J., Ishwar, V., & Serra, X. (2014). Melodic Pattern Extraction in Large Collections of Music Recordings Using Time Series Mining Techniques. In Late-Breaking Demo Session of the 15th International Society for Music Information Retrieval Conference. Taipei, Taiwan.
- **Gulati, S.**, Ganguli, K. K., Gupta, S., Srinivasamurthy, A., & Serra, X. (2015). Rāgawise: A Lightweight Real-time Raga Recognition System for Indian Art Music. In Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference. Malaga, Spain.
(Companion webpage: <http://compmusic.upf.edu/node/281>)
- Caro, R., Srinivasamurthy, A., **Gulati, S.**, & Serra, X. (2014). Jingju music: Concepts and Computational Tools for its Analysis. A Tutorial in the 15th International Society for Music Information Retrieval Conference, Taipei, Taiwan.
(Presented the melody part of the tutorial, focused on melodic analysis tools for jingju music. Companion webpage: <http://compmusic.upf.edu/jingju-tutorial>)

Appendix B

Resources

In this appendix, we provide the URLs to access the resources related to this thesis, which include the code and tools, the corpora and test datasets, and the applications and demos. An up-to-date links of all these resources is maintained on the companion web page of this thesis: <http://compmusic.upf.edu/node/304> (mirrored at <http://www.sankalpgulati.in/phd-thesis.html>).

Code and Tools

Implementation of Methods

Implementation of the methods presented in this thesis are made publicly available for research purposes. We also indicate (in squared brackets) the language in which the implementation is available.

- Tonic identification (**M_{JS}**) [C]
- Nyās segmentation and classification [Python]
- Predominant pitch post-processing [Python]
- **Tani Segmentation** [Python]
- Pattern processing: melodic similarity, and pattern search and discovery (including cascaded lower-bound computations) [C]
- DTW variants [C, Python-wrapper]
- Melodic pattern characterization [Python]
- Rāga recognition using melodic patterns (**M_{VSM}**) [Python]
- Rāga recognition using TDMS (**M_{TDMS}**) [Python]
- Pitch estimation using the YIN algorithm [JavaScript]

Tools

- Essentia audio analysis library (<http://essentia.upf.edu/>)
- Dunya API (<https://github.com/MTG/pycompmusic>)
- Dunya front end (<http://dunya.compmusic.upf.edu/>)
- Dunya server and back end (<https://github.com/MTG/dunya>)

Corpora and Test-datasets

Corpora

All the research corpora described in this thesis, which include audio recordings, associated metadata and audio features can be accessed through the Dunya API (Section 7.2). Every corpora has a corresponding collection in MusicBrainz.

- ‘Dunya Carnatic’ collection in MusicBrainz forms the Carnatic music corpus (Section 3.2.2)
<https://musicbrainz.org/collection/f96e7215-b2bd-4962-b8c9-2b40c17a1ec6>
- ‘Dunya Hindustani’ collection in MusicBrainz forms the Hindustani music corpus (Section 3.2.2)
<https://musicbrainz.org/collection/213347a9-e786-4297-8551-d61788c85c80>
- ‘Dunya Carnatic CC’ and ‘Dunya Hindustani CC’ collection in MusicBrainz forms the open-access music corpus (Section 3.2.4)
<https://musicbrainz.org/collection/a163c8f2-b75f-4655-86be-1504ea2944c2>
<https://musicbrainz.org/collection/6adc54c6-6605-4e57-8230-b85f1de5be2b>

Test Datasets

All the test datasets described in this thesis are shared as standalone archives (files) that contain relevant annotations and audio features. For the datasets that use audio recordings present in the corpora, the recordings can be accessed though the Dunya API. Otherwise, they are bundled together with the annotations. List of all the datasets which are made publicly available.

- Tonic identification datasets (TID_{CM1} , TID_{CM2} , TID_{CM3} , TID_{IITM1} , TID_{IITM2} , and TID_{IISc})
- Nyās detection dataset (ND_{CM})
- Melodic similarity datasets (MSD_{IITM}^{cmd} , MSD_{IITB}^{hmd} , MSD_{CM}^{cmd} , and MSD_{CM}^{hmd})

- Rāga recognition datasets (RRD_{CMD} and RRD_{HMD})

The corpora and the test datasets compiled in the CompMusic project for all the music traditions are available at:

CompMusic music corpora

<http://compmusic.upf.edu/corpora>

CompMusic test datasets

<http://compmusic.upf.edu/datasets>

Applications and Demos

- Discovered melodic patterns:
 - Metadata-wise browsing: dunya.compmusic.upf.edu/motifdiscovery
 - Network of melodic patterns: dunya.compmusic.upf.edu/pattern_network/
 - We also share the database of the discovered melodic patterns that contains the time-stamps, similarity with the other patterns, and the corresponding recording information for all the patterns.
- Rāgawise:
 - Web interface: <https://dunya.compmusic.upf.edu/ragewise/>
 - Code: <https://github.com/sankalpg/ragewise>
- Mobile applications:
 - Sarāga: <https://play.google.com/store/apps/details?id=com.musicmuni.saraga>
 - Riyāz: <https://play.google.com/store/apps/details?id=com.musicmuni.riyaz>

We reiterate that the links provided for some of these resources may change over time. An up-to-date links of all these resources are maintained on the companion web page of this thesis: <http://compmusic.upf.edu/node/304> (mirrored at <http://www.sankalpgulati.in/phd-thesis.html>).

Appendix C

Additional Figures and Tables

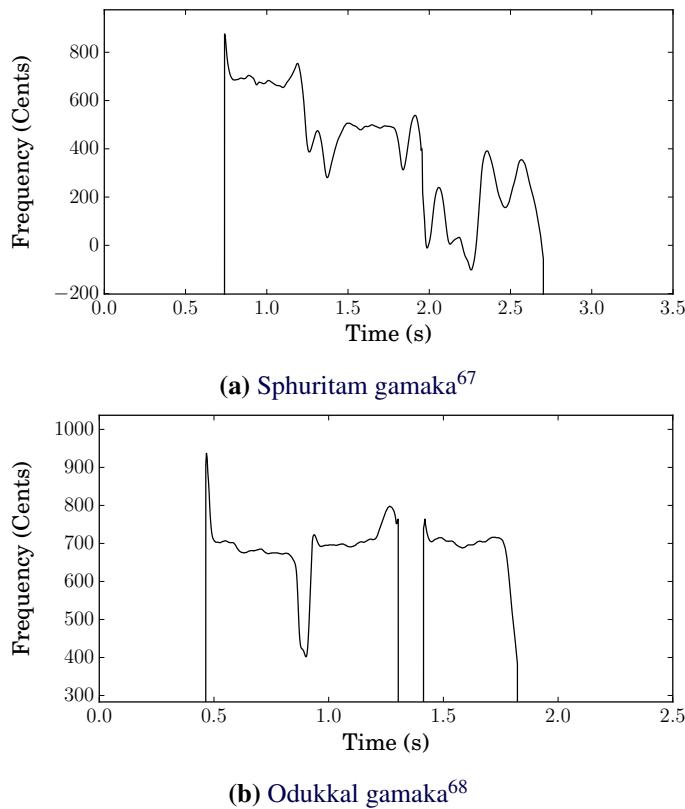


Figure C.1: Examples of the sphuritam and odukkal gamaka in Carnatic music

⁶⁷<https://www.freesound.org/people/sankalp/sounds/360769/>

⁶⁸<https://www.freesound.org/people/sankalp/sounds/360770/>

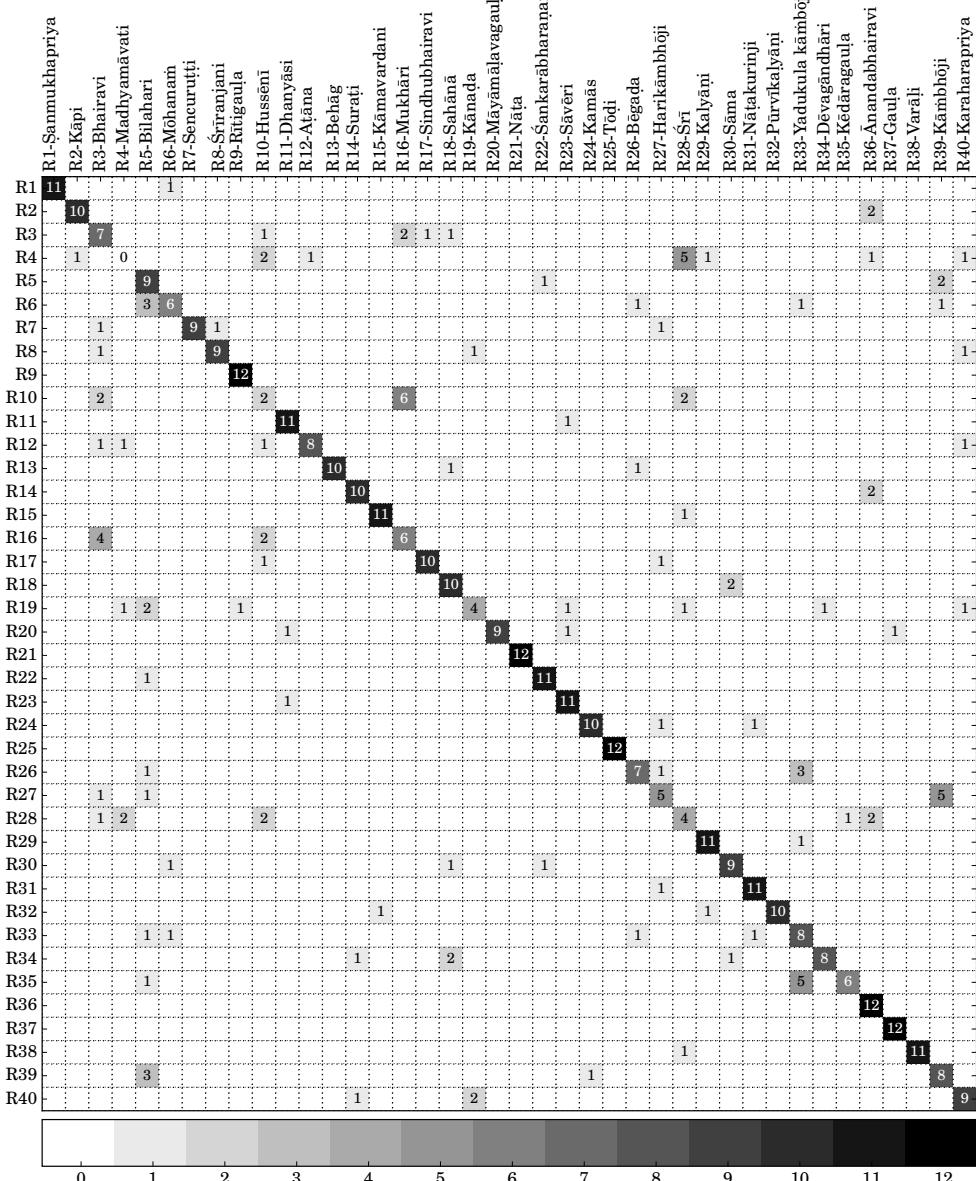


Figure C.2: Confusion matrix of the `rāga` predictions by `MPC` on `RRDCMD` dataset. The different shades of grey are mapped to different number of audio recordings.

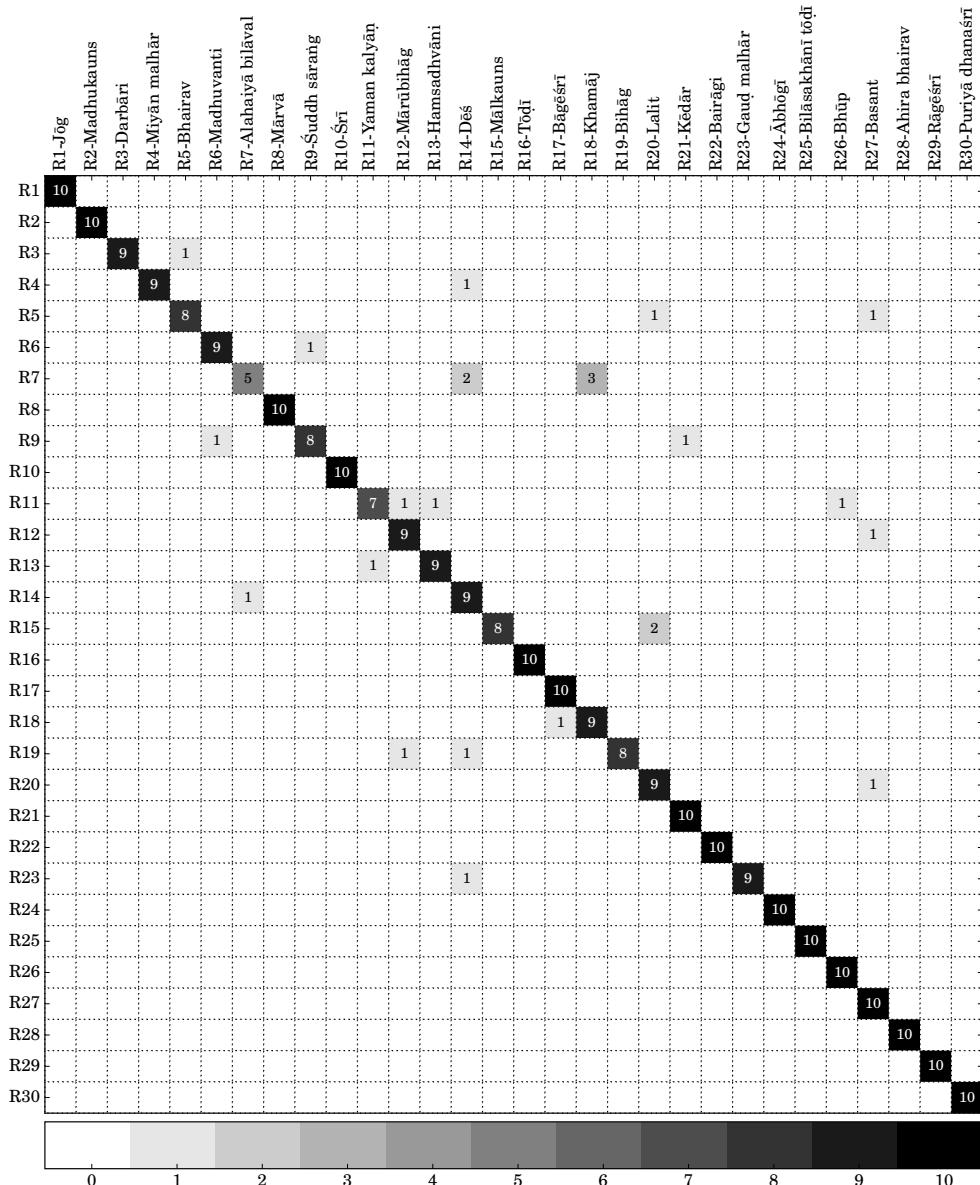


Figure C.3: Confusion matrix of the rāga predictions by Mpc on RRD_{HMD} dataset. The different shades of grey are mapped to different number of audio recordings.

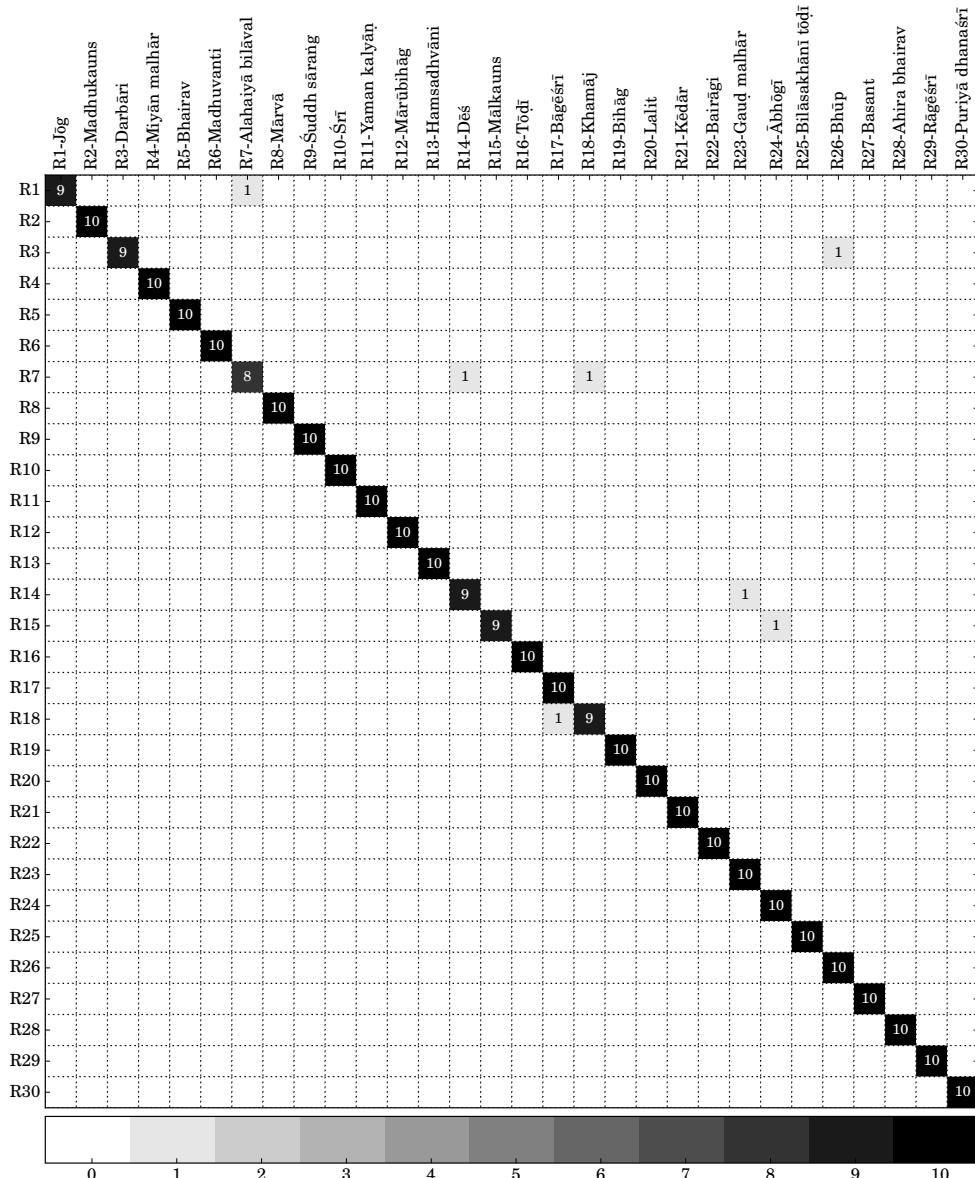


Figure C.4: Confusion matrix of the rāga predictions by MTDMS on RRD_{HMD} dataset. The different shades of grey are mapped to different number of audio recordings.

Svara (full / short name)	Carnatic variant	Carnatic music Notation	Hindustani variant	Hindustani music Notation	Position (w.r.t. the tonic)
Ṣadja (Sa)	Ṣadja	S	Ṣadja	S	0
	Śuddha Rishabha	R1	Kōmal Riṣhabha	r	1
Rishabha (Re)	Chatuśruti Rishabha	R2/G1	Śuddha Riṣhabha	R	2
	Sādhāraṇa gāndhāra	G2/R3	Kōmal gāndhāra	g	3
Gāndhāra (Ga)	Antara gāndhāra	G3	Śuddha gāndhāra	G	4
	Śuddha madhyama	M1	Śuddha madhyama	m	5
Madhyama (Ma)	Prati madhyama	M2	Tivra madhyama	M	6
	Pañchama (Pa)	P	Pañchama	P	7
Dhaivata (Dha)	Śuddha dhaivata	D1	Kōmal dhaivata	d	8
	Chatuśruti dhaivata	D2/N1	Śuddha dhaivata	D	9
Niṣāda (Ni)	Kaiśikī niṣāda	N2/D3	Kōmal niṣāda	n	10
	Kākalī niṣāda	N3	Śuddha niṣāda	N	11

Table C.1: Svara names and notations used in Carnatic and Hindustani music. The precise frequency and the intonation of these svaras depend on the tonic of the lead artist in the recording and on the rāga. Note that in Carnatic music, for some svara positions, there are two possible notations. One of these notations is used in the context of a given music piece, the selection of which depends on the rāga of the piece.

Rāga	S	r	R	g	G	m	M	P	d	D	n	N
Jōg	●				●	●	●		●			●
Madhukauns	●				●		●		●		●	
Darbāri	●		●	●		●		●	●		●	
Miyān malhār	●		●	●		●		●		●	●	●
Bhairav	●	●				●	●		●	●		●
Madhuvanti	●		●	●			●	●		●		●
Alahaiyā bilāval	●		●		●	●		●		●	●	●
Mārvā	●	●				●		●		●		●
Śuddh sāraṅg	●			●		●	●	●		●		●
Śrī	●	●				●	●	●		●		●
Yaman kalyāṇ	●			●		●	●	●		●		●
Mārūbihāg	●			●		●	●	●		●		●
Hamsadhvāni	●				●			●				●
Dēś	●		●			●	●	●		●	●	●
Mālkauns	●				●	●			●		●	
Tōdī	●	●			●			●	●	●		●
Bāgēśrī	●		●	●		●		●		●	●	
Khamāj	●			●		●	●	●		●	●	●
Bihāg	●		●			●	●	●		●		●
Lalit	●	●				●	●	●		●		●
Kēdār	●		●			●	●	●		●	●	●
Bairāgi	●	●				●		●			●	
Gauḍ malhār	●			●		●	●		●	●		●
Ābhōgī	●			●	●		●			●		
Bilāsakhānī tōdī	●	●		●		●		●	●		●	
Bhūp	●		●			●		●		●		
Basant	●	●				●	●	●	●			●
Ahira bhairav	●	●	●			●	●		●	●	●	
Rāgēśrī	●		●			●	●			●	●	
Puriyā dhanaśrī	●	●				●		●	●	●		●

Table C.2: List of the rāgas in RRD_{HMD} dataset along with their constituent set of svaras. The contents of this table are verified by Kaustuv K. Ganguli, a professional musician (vocalist of Hindustani music).

Rāga	Total Duration (hrs)	#Lead Artists	#Releases
Jōg	4.68	8	8
Madhukauns	2.86	7	8
Darbāri	5.39	8	10
Miyān malhār	5.82	9	10
Bhairav	3.28	5	6
Madhuvanti	4.01	10	8
Alahaiyā bilāval	3.02	7	8
Mārvā	4.06	9	10
Śuddh sāraṅg	2.66	8	8
Śrī	6.1	7	8
Yaman kalyāṇ	4.2	9	10
Mārūbihāg	4.15	7	8
Hamsadhvāni	2.41	6	6
Dēś	2.19	5	7
Mālkauns	5.89	13	10
Tōḍī	4.97	13	10
Bāgēśrī	4.86	10	10
Khamāj	2.52	2	4
Bihāg	4.04	5	5
Lalit	5.29	9	8
Kēdār	3.32	6	6
Bairāgi	3.09	7	8
Gauḍ malhār	4.33	8	8
Ābhōgī	3.3	9	9
Bilāsakhānī tōḍī	3.79	9	10
Bhūp	3.49	7	8
Basant	2.74	9	10
Ahira bhairav	3.14	9	10
Rāgēśrī	3.35	5	6
Puriyā dhanaśrī	3.23	11	10
Total	116.2	60	162

Table C.3: Details of RRD_{HMD} dataset for each constituent rāga in terms of the total duration, the number of unique lead artists and the total releases associated with the audio recordings in the collection. There are 10 recordings for every rāga in this dataset.

Rāga	S	R1	R2	G2/R3	G3	M1	M2	P	D1	D2/N1	N2/D3	N3
Śanmukhapriya	●	-	-	●	-	-	-	●	●	-	-	●
Kāpi	●	-	-	●	●	●	-	●	-	●	-	●
Bhairavi	-	-	-	-	-	●	●	-	-	-	-	●
Madhyamāvati	●	-	-	-	-	●	-	-	-	-	-	●
Bilahari	●	-	-	-	●	●	-	●	-	-	-	●
Mōhanām	●	-	-	-	●	-	-	-	-	-	-	●
Sencurūtti	●	-	-	-	-	●	-	-	-	-	-	●
Śrīranjani	●	-	-	-	-	●	-	-	-	-	-	●
Rītigauļa	●	-	-	-	-	●	-	-	-	-	-	●
Hussēnī	●	-	-	-	-	●	-	●	-	-	-	●
Dhanyāsi	●	●	-	-	-	●	-	●	-	-	-	●
Aṭāna	●	-	-	-	-	●	-	●	-	-	-	●
Behāg	●	-	-	-	-	●	-	●	-	-	-	●
Suraṭi	●	-	-	-	-	●	-	●	-	-	-	●
Kāmavardani	●	●	-	-	-	●	●	●	-	-	-	●
Mukhāri	●	-	-	-	-	●	-	●	-	-	-	●
Sindhubbhairavi	●	●	●	●	●	●	●	●	●	●	●	●
Sahānā	●	-	-	-	-	●	-	●	-	-	-	●
Kānaḍa	●	-	-	-	-	●	-	●	-	-	-	●
Māyāmālavaṅgaula	●	●	-	-	-	●	●	●	-	-	-	●
Nāṭa	●	-	-	-	-	●	●	●	-	-	-	●
Śankarābharaṇam	●	-	-	-	-	●	●	●	-	-	-	●
Sāvēri	●	●	-	-	-	●	●	●	-	-	-	●
Kamās	●	-	●	-	-	●	●	●	-	-	-	●
Tōḍī	●	●	-	-	-	●	●	●	-	-	-	●
Bēgaḍa	●	-	-	-	-	●	●	●	-	-	-	●
Harikāmbhōji	●	-	-	-	-	●	●	●	-	-	-	●
Śrī	●	-	-	-	-	●	●	●	-	-	-	●
Kalyāṇi	●	-	-	-	-	●	●	●	-	-	-	●
Sāma	●	-	-	-	-	●	●	●	-	-	-	●
Nāṭakurinji	●	-	-	-	-	●	●	●	-	-	-	●
Pūrvikalyāṇi	●	-	-	-	-	●	●	●	-	-	-	●
Yadukula kāmbōji	●	-	-	-	-	●	●	●	-	-	-	●
Dēvagāndhāri	●	-	-	-	-	●	●	●	-	-	-	●
Kēdāragauļa	●	-	-	-	-	●	●	●	-	-	-	●
Ānandabhairavi	●	-	-	-	-	●	●	●	-	-	-	●
Gauļa	●	●	-	-	-	●	●	●	-	-	-	●
Varāli	●	●	-	-	-	●	●	●	-	-	-	●
Kāmbhōji	●	-	-	-	-	●	●	●	-	-	-	●
Karaharapriya	●	●	-	-	-	●	●	●	-	-	-	●

Table C.4: List of the rāgas in RRD_{CMD} dataset along with their constituent set of svaras. The svaras are marked based on the performance practices in Carnatic music. The contents of this table are verified by Vignesh Ishwar, a professional musician (vocalist of Carnatic music).

Rāga	Total Duration (hrs)	#Lead Artists	#Releases
Śānmukhapriya	3.05	12	12
Kāpi	1.19	9	12
Bhairavi	5.51	9	12
Madhyamāvati	3.45	12	12
Bilahari	3.37	11	12
Mōhanam	4.71	8	12
Sencurutti	0.92	10	12
Śīranjani	1.93	12	12
Rītigauļa	3.23	11	12
Hussēnī	1.25	10	12
Dhanyāsi	3.37	8	12
Atāna	1.67	11	12
Behāg	1.26	10	12
Suraṭi	2.3	11	12
Kāmavardani	3.51	11	12
Mukhāri	3.3	12	12
Sindhuhairavi	1.01	10	12
Sahānā	2.63	11	12
Kānaḍa	2.82	9	12
Māyāmālavagauļa	2.58	11	12
Nāṭa	1.74	11	12
Śankarābharaṇam	4.76	8	12
Sāvēri	2.97	10	12
Kamās	2.39	8	12
Tōḍī	7.23	9	12
Bēgaḍa	2.98	9	12
Harikāmbhōji	3.8	9	12
Śrī	1.51	10	12
Kalyāṇi	5.42	9	12
Sāma	1.16	10	12
Nāṭakurinji	1.8	10	12
Pūrvikalyāni	5.87	9	12
Yadukula kāmbōji	2.13	11	12
Dēvagāndhāri	2.27	11	12
Kēdāragauļa	4.08	11	11
Ānandabhairavi	1.84	9	12
Gauļa	2.05	6	11
Varāli	3.92	10	12
Kāmbhōji	6.2	9	12
Karaharapriya	7.28	11	12
Total	124.5	65	188

Table C.5: Details of RRD_{CMD} dataset for each constituent rāga in terms of the total duration, the number of unique lead artists and the total releases associated with the audio recordings in the collection. There are 12 recordings for every rāga in this dataset.

Appendix D

Glossary

D.1 Acronyms

k -NN	k -nearest neighbors
M_B	Method for computing melodic similarity that uses the best set of procedures and parameter settings obtained from the grid-search
M_{CW1}	Method for computing melodic similarity using complexity weighting (variant 1)
M_{CW2}	Method for computing melodic similarity using complexity weighting (variant 2)
M_{DT}	Method for computing melodic similarity using svara duration truncation
MSD	Melodic similarity dataset
MSD_{CM}	Improved melodic similarity dataset
MSD_{CM}^{cmd}	Improved Carnatic music melodic similarity dataset
MSD_{CM}^{hmd}	Improved Hindustani music melodic similarity dataset
MSD_{iitb}^{hmd}	Hindustani music melodic similarity dataset compiled at IIT Bombay, Mumbai
MSD_{iitm}^{cmd}	Carnatic music melodic similarity dataset compiled at IIT Madras, Chennai
NDD_{CM}	Nyas detection dataset
\mathfrak{B}_{R1}	Random baseline method for nyās segmentation (variant 1)
\mathfrak{B}_{R3}	Random baseline method for nyās segmentation (variant 3)
\mathfrak{B}_{R2}	Random baseline method for nyās segmentation (variant 2)
\mathcal{F}_C	Contextual features used in nyās segment classification
\mathcal{F}_L	Local features used in nyās segment classification
M_{TDMS}	Method for rāga recognition using TDMS
M_{TDMS}^B	Method for rāga recognition using TDMS and Bhattacharyya distance
M_{TDMS}^F	Method for rāga recognition using TDMS and Frobenius norm

M_{TDMS}^{KL}	Method for <i>rāga</i> recognition using TDMS and Kullback-Leibler divergence distance
M_{VSM}	Method for <i>rāga</i> recognition using vector space modeling
RRD_{CMD}	<i>Rāga</i> recognition dataset comprising 480 recordings of Carnatic music in 40 <i>rāgas</i>
RRD_{HMD}	<i>Rāga</i> recognition dataset comprising 300 recordings of Hindustani music in 30 <i>rāgas</i>
M_{PC}	Method for <i>rāga</i> recognition proposed by Chordia & Şentürk (2013)
M_{GK}	Method for <i>rāga</i> recognition proposed by Koduri et al. (2014)
TID_{CM1}	Tonic identification dataset comprising three minute instrumental excerpts
TID_{CM2}	Tonic identification dataset comprising three minute vocal excerpts
TID_{CM3}	Tonic identification dataset comprising full length vocal recordings
TID_{IISc}	Tonic identification dataset compiled in IISc (Ranjani et al., 2011)
TID_{IITM1}	Tonic identification dataset comprising full length concerts (Bellur et al., 2012)
TID_{IITM2}	Tonic identification dataset comprising full length recordings (Bellur et al., 2012)
<i>n</i> -Gram	<i>n</i> -gram model
M_{AB1}	Tonic identification method proposed by Bellur et al. (2012) (Variant 1)
M_{AB2}	Tonic identification method proposed by Bellur et al. (2012) (Variant 2)
M_{AB3}	Tonic identification method proposed by Bellur et al. (2012) (Variant 3)
M_{CS}	Tonic identification method proposed by Chordia & Şentürk (2013)
M_{JS}	Tonic identification method proposed by Salamon et al. (2012)
M_{RH1}	Tonic identification method proposed by Ranjani et al. (2011) (Variant 1)
M_{RH2}	Tonic identification method proposed by Ranjani et al. (2011) (Variant 2)
M_{SG}	Tonic identification method proposed by Gulati et al. (2012)
M_{RS}	Tonic identification method proposed by Datta (1996)
1-NN	1-nearest neighbor
ACC	Arkay Convention Center
ACR	autocorrelation
AIR	All India Radio
AMDF	average magnitude difference function
API	application programming interface
BSS	behavioral symbol sequence
CAMUT	Culture Aware MUsic Technologies
CDDTW	context dependent dynamic time warping
cDTW	constrained dynamic time warping

DTW	dynamic time warping
Essentia	an open-source C++ library for audio analysis and content-based MIR (Bogdanov et al., 2013)
FPD	fine-grained pitch distribution
GD	group delay
GMM	Gaussian mixture model
HMM	hidden markov model
HPCP	harmonic pitch-class profiles
IAM	Indian art music
ICM	Indian classical music
IOI	inter onset interval
IOIr	inter onset interval ratio
IT	information technology
ITC-SRA	ITC Sangeet Research Academy
KDE	kernel density estimation
KPD	kernel-density pitch distribution
LB_Keogh	LB_Keogh lower bound (Keogh & Ratanamahatana, 2004)
LB_Keogh_EC	LB_Keogh lower bound for reference to query (Rakthanmanon et al., 2013)
LB_Keogh_EQ	LB_Keogh lower bound for query to reference (Rakthanmanon et al., 2013)
LB_KIM_FL	First-last lower bound (Kim et al., 2001)
LCS	longest common subsequence
LDTW	local dynamic time warping
LR	logistic regression
MAP	mean average precision
MBID	MusicBrainz identifier
Melodia	Predominant melody extraction algorithm proposed by Salamon & Gómez (2012)
MFCC	Mel-frequency cepstral coefficient
MIR	music information research
MMA	Madras Music Academy
MSD	million song dataset
NB	naive Bayes
NBB	Bernoulli naive Bayes
NBG	Gaussian naive Bayes
NBM	multinomial naive Bayes
NCPA	National Centre for the Performing Arts
PCD	pitch-class distribution
PCDD	pitch-class dyad distribution
PCP	pitch-class profile
PDE	probability density estimate
PLS	piece-wise linear segmentation

PSA	phase space analysis
PyCompMusic	Python wrapper around Dunya API
QBE	query-by-example
QBH	query-by-humming
Rāgawise	A light weight web-based real-time rāga recognition system
RESTful	representational state transfer
RF	random forest
Riyāz	A mobile application that facilitate music practice of Indian music forms
RLCS	rough longest common subsequence
ROC	receiver operating characteristic
Sarāga	A music appreciation and infotainment application for students and rasikas
SAX	symbolic aggregate approximation
SGD	stochastic gradient descent
SMBGT	subsequence matching with bounded gaps and tolerances
SSM	self-similarity matrix
SVM	support vector machines
SVML	support vector machines with linear kernel
SVMR	support vector machines with radial basis function kernel
TDMS	time delayed melodic surface
TF-IDF	term frequency inverse document frequency
TMM	Turkish makam music
Tree	decision tree
UTW	uniform time warping
VSM	vector space modeling
WAQ	width-across-query
WAR	width-across-reference

D.2 Music Terms

alankār	Melodic gestures serving as ornaments in Hindustani music
ālāp	Unmetered improvisatory opening section in Hindustani music
ālāpna	Unmetered improvisatory opening section in Carnatic music
ārōhana	Ascending progression of svaras
avrōhana	Descending progression of svaras
chalan	Melodic outline of a rāga
dhaivata	The sixth scale degree svara with respect to the base svara Sa
dhrupad	A vocal form in Hindustani music
gamaka	Melodic gestures in Carnatic music
gāndhāra	The third scale degree svara with respect to the base svara Sa
gharānā	A system of social organization linking musicians by lineage or apprenticeship in Hindustani music

ghatam	A percussion instrument in Carnatic music
kampitam	A kind of gamaka in Carnatic music
kanjira	A percussion instrument in Carnatic music
kārvai	A musical pause in Carnatic music
kachēri	Assembly of musicians and audience in the context of Carnatic music, presented in the concert format
khatkā	A type of alankār in Hindustani music
khyāl	A vocal form in Hindustani music
kīrtana	A musical form (typically religious) in India
lay	Tempo range used in Hindustani music.
madhyama	The fourth scale degree svara with respect to the base svara Sa
mīnd	A type of alankār in Hindustani music
mṛdaṅgam	The main percussion instrument in Carnatic music
mukhda	The opening line of a composition in Hindustani music
murkī	A specific type of alankār in Hindustani music
niśāda	The seventh scale degree svara with respect to the base svara Sa
nyās	The phenomenon of resting/sustaining a svara in melodies of Indian art music
odukkal	A kind of gamaka in Carnatic music
pallavi	A single line composition set to a rāga and a tāla or a thematic line of a song
pañchama	The fifth scale degree svara with respect to the base svara Sa
rāga	Melodic framework in Indian art music
ṛgved	An ancient collection of Vedic Sanskrit hymns in Indian
Riśabha	The second scale degree svara with respect to the base svara Sa
sama	Equivalent to a downbeat in Indian art music (beginning of a tāla cycle)
sāmagān	Singing hymns of Veda in ancient India
samvādi	The second most salient svara in a melody
sāmved	Veda (large body of texts) of melodies and chants in ancient India
sāraṅgi	A fretless instrument in Indian art music
ṣadja	The root (Sa) svara in melodies of Indian art Music
śruti	Tonic pitch of the lead artist used in a performance
sphuritam	A kind of gamaka in Carnatic music
svara	Equivalent to musical note in Indian art music
svarasthānā	The precise pitch and intonation of a svara
tablā	A membranophone percussion instrument used in Hindustani music
tāla	Rhythmic framework in Indian art music
tani	Tani avartanam, solo percussion section in Carnatic music sort
tānpura	A long-necked plucked string instrument used for generating drone sound in Indian art music
thumrī	A light classical vocal form in Hindustani music
vādi	The most salient svara in a melody

vedas
vīṇa

Large body of texts in ancient India
A plucked string instrument mainly used in Carnatic music

Bibliography

- Akkoç, C. (2002). Non-deterministic scales used in traditional Turkish music. *Journal of New Music Research*, 31(4), 285–293. [Cited on page 47.]
- Aucouturier, J. J. & Sandler, M. (2002). Finding repeating patterns in acoustic musical signals : Applications for audio thumbnailing. In *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. [Cited on page 51.]
- Baeza-Yates, R. A. & Perleberg, C. H. (1992). Fast and practical approximate string matching. In *Annual Symposium on Combinatorial Pattern Matching*, pp. 185–192. Springer. [Cited on page 51.]
- Bagchee, S. (1998). *Nād understanding raga music*. Business Publications Inc. [Cited on pages 16, 17, 18, 19, 20, 23, and 110.]
- Balkwill, L. L. & Thompson, W. F. (1999). A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music perception: an interdisciplinary journal*, 17(1), 43–64. [Cited on page 227.]
- Bartsch, M. A. & Wakefield, G. H. (2001). To catch a chorus: Using chroma-based representations for audio thumbnailing. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pp. 15–18. [Cited on page 46.]
- Bartsch, M. A. & Wakefield, G. H. (2005). Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on multimedia*, 7(1), 96–104. [Cited on page 54.]
- Batista, G. E., Wang, X., & Keogh, E. J. (2011). A complexity-invariant distance measure for time series. In *SDM*, vol. 11, pp. 699–710. [Cited on pages 139, 141, and 142.]
- Belle, S., Joshi, R., & Rao, P. (2009). Raga identification by using swara intonation. *Journal of ITC Sangeet Research Academy*, 23. [Cited on pages 38, 39, 41, and 43.]
- Bellur, A., Ishwar, V., Serra, X., & Murthy, H. A. (2012). A knowledge based signal processing approach to tonic identification in Indian classical music. In *2nd CompMusic Workshop*, pp. 113–118. Universitat Pompeu Fabra. [Cited on pages 23, 25, 26, 28, 29, 30, 31, 80, 88, 99, and 248.]

- Bertin-Mahieux, T., Ellis, D. P., Whitman, B., & Lamere, P. (2011). The million song dataset. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 591–596. [Cited on page 62.]
- Bhatkhande, V. N. (1990). *Hindustani Sangeet Paddhati: Kramik Pustak Maalika Vol. I-VI*. Sangeet Karyalaya. [Cited on page 71.]
- Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhyā: the Indian journal of statistics*, pp. 401–406. [Cited on page 55.]
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. [Cited on page 29.]
- Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, (10), P10008. [Cited on pages 168 and 182.]
- Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the institute of phonetic sciences*, 17(1193), 97–110. [Cited on page 26.]
- Boersma, P. & Weenink, D. (2001). Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10), 341–345. [Cited on pages 25, 26, 39, and 42.]
- Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., & Serra, X. (2013). Essentia: an audio analysis library for music information retrieval. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 493–498. [Cited on pages 101, 102, 109, and 249.]
- Bor, J., Delvoye, F. N., Harvey, J., & Nijenhuis, E. T. (Eds.) (2010). *Hindustani Music: Thirteenth to Twentieth Centuries*. New Delhi: Manohar Publishers and Distributors, first edn. [Cited on pages 15 and 17.]
- Bozkurt, B. (2008). An automatic pitch analysis method for Turkish Maqam music. *Journal of New Music Research*, 37(1), 1–13. [Cited on pages 46 and 47.]
- Bozkurt, B., Karaosmanoğlu, M. K., Karaçalı, B., & Ünal, E. (2014). Usul and makam driven automatic melodic segmentation for turkish music. *Journal of New Music Research*, 43(4), 375–389. [Cited on page 34.]
- Brown, H. (1988). The interplay of set content and temporal context in a functional theory of tonality perception. *Music Perception*, 5(3), 219–249. [Cited on page 45.]
- Burgoyne, J. A., Fujinaga, I., & Downie, J. S. (2015). *Music Information Retrieval*, pp. 213–228. John Wiley & Sons, Ltd. [Cited on page 3.]

- Camacho, A. (2007). *SWIPE: A sawtooth waveform inspired pitch estimator for speech and music*. Ph.D. thesis, University of Florida. [Cited on page 39.]
- Cambouropoulos, E. (1996). A formal theory for the discovery of local boundaries in a melodic surface. *Proceedings of the III Journées d'Informatique Musicale*. [Cited on page 49.]
- Cambouropoulos, E. (1997). *Towards a general computational theory of musical structure*. Ph.D. thesis, University of Edinburgh. [Cited on page 48.]
- Cambouropoulos, E. (2001a). The local boundary detection model (lbdm) and its application in the study of expressive timing. In *Proceedings of the international computer music conference*, pp. 17–22. [Cited on pages 49 and 151.]
- Cambouropoulos, E. (2001b). Melodic Cue Abstraction, Similarity, and Category Formation: A Formal Model. *Music Perception: An Interdisciplinary Journal*, 18(3), 347–370. [Cited on pages 48 and 135.]
- Cambouropoulos, E. (2006). Musical parallelism and melodic segmentation: a computational approach. *Music Perception*, 23(3), 249–268. [Cited on pages 5, 6, 34, 48, 49, 147, 151, 163, and 226.]
- Cambouropoulos, E. (2009). How similar is similar? *Musicae Scientiae*, 13(1 Suppl), 7–24. [Cited on page 48.]
- Cambouropoulos, E., Crawford, T., & Iliopoulos, C. S. (2001). Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers and the Humanities*, 35(1), 9–21. [Cited on pages 48 and 49.]
- Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008). Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4), 668–696. [Cited on page 3.]
- Celma, Ò. (2006). Foafing the music: Bridging the semantic gap in music recommendation. In *International Semantic Web Conference*, pp. 927–934. Springer. [Cited on page 3.]
- Chai, W. & Vercoe, B. (2003). Music thumbnailing via structural analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pp. 223–226. ACM. [Cited on page 51.]
- Chakraborty, S. & De, D. (2012). Object oriented classification and pattern recognition of Indian classical ragas. In *1st International Conference on Recent Advances in Information Technology (RAIT)*, pp. 505–510. [Cited on pages 38, 39, 40, and 43.]
- Chen, T., Yap, K.-H., & Zhang, D. (2012). Discriminative bag-of-visual phrase learning for landmark recognition. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 893–896. [Cited on page 111.]

- Chew, E. (2000). *Towards a mathematical model of tonality*. Ph.D. thesis, Massachusetts Institute of Technology. [Cited on page 45.]
- Chordia, P. & Rae, A. (2007). Raag recognition using pitch-class and pitch-class dyad distributions. In *In Proc. of Int. Soc. for Music Information Retrieval Conf.*, pp. 431–436. [Cited on pages 38, 39, 40, 41, 42, and 43.]
- Chordia, P. & Şentürk, S. (2013). Joint recognition of raag and tonic in north Indian music. *Computer Music Journal*, 37(3), 82–98. [Cited on pages 18, 23, 38, 39, 41, 43, 47, 88, 184, 194, 197, 199, and 248.]
- Cohen, A. J. (1991). Tonality and perception: Musical scales primed by excerpts from The Well-Tempered Clavier of JS Bach. *Psychological Research*, 53(4), 305–314. [Cited on page 45.]
- Collins, T. (2011). *Improved methods for pattern discovery in music, with applications in automated stylistic composition*. Ph.D. thesis, Open University. [Cited on pages 47 and 147.]
- Collins, T., Arzt, A., Flossmann, S., & Widmer, G. (2013). SIARCT-CFP: Improving Precision and the Discovery of Inexact Musical Patterns in Point-Set Representations. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 549–554. [Cited on page 48.]
- Collins, T., Böck, S., Krebs, F., & Widmer, G. (2014). Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society. [Cited on pages 50, 135, and 147.]
- Collins, T., Laney, R., Willis, A., & Garthwaite, P. H. (2011). Modeling pattern importance in chopin's mazurkas. *Music Perception: An Interdisciplinary Journal*, 28(4), 387–414. [Cited on pages 47, 49, and 163.]
- Conklin, D. (2010). Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14(5), 547–554. [Cited on pages 48, 49, and 163.]
- Conklin, D. & Anagnostopoulou, C. (2001). Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference*, pp. 479–485. [Cited on pages 47, 48, and 147.]
- Conklin, D. & Anagnostopoulou, C. (2011). Comparative pattern analysis of cretan folk songs. *Journal of New Music Research*, 40(2), 119–125. [Cited on pages 47, 49, 135, and 163.]
- Conklin, D. & Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1), 51–73. [Cited on page 48.]

- Cremer, M. (2004). A System for Harmonic Analysis of Polyphonic Music. In *25th Audio Engineering Society (AES) Conference*. [Cited on page 46.]
- Şentürk, S., Gulati, S., & Serra, X. (2013). Score informed tonic identification for makam music of turkey. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 175–180. Curitiba, Brazil. [Cited on page 101.]
- Danielou, A. (2010). *The ragas of Northern Indian music*. New Delhi: Munshiram Manoharlal Publishers. [Cited on pages 15, 17, 18, and 19.]
- Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., & Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5), 687–701. [Cited on pages 52, 53, 126, and 135.]
- Dannenberg, R. B. & Hu, N. (2003). Pattern discovery techniques for music audio. *Journal of New Music Research*, 32(2), 153–163. [Cited on pages 47, 51, and 147.]
- Datta, A. K. (1996). Generation of musical notations from song using state-phase for pitch detection algorithm. *Journal of Acoustical Society of India*, 24. [Cited on pages 25, 26, and 248.]
- Datta, A. K., Sengupta, R., Dey, N., & Nag, D. (2007). A methodology for automatic extraction of ‘meend’ from the performances in Hindustani vocal music. *Journal of ITC Sangeet Research Academy*, 21, 24–31. [Cited on page 33.]
- De Cheveigné, A. & Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4), 1917–1930. [Cited on pages 25 and 26.]
- Deshpande, V. H. (1989). *Indian Musical Traditions: An Aesthetic Study of the Gharanas in Hindustani Music*. Popular Prakashan, second edn. [Cited on page 16.]
- Deva, B. C. (1980). *The Music of India: A Scientific Study*. Delhi: Munshiram Manoharlal Publishers. [Cited on page 19.]
- Dey, A. K. (2008). *Nyāsa in rāga: the pleasant pause in Hindustani music*. Kanishka Publishers, Distributors. [Cited on pages 19 and 110.]
- Dighe, P., Agrawal, P., Karnick, H., Thota, S., & Raj, B. (2013a). Scale independent raga identification using chromagram patterns and swara based features. In *IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, pp. 1–4. [Cited on pages 38, 39, 41, 42, and 43.]
- Dighe, P., Karnick, H., & Raj, B. (2013b). Swara histogram based structural analysis and identification of Indian classical ragas. In *In Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 35–40. [Cited on pages 38, 39, 42, and 43.]

- Duda, R., Hart, P., & Stork, D. (2000). *Pattern Classification*. Wiley, New York, 2nd edition edn. [Cited on page 29.]
- Duong, N. Q. K. & Thudor, F. (2013). Movie synchronization by audio landmark matching. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3632–3636. [Cited on page 111.]
- Durey, A. & Clements, M. A. (2001). Melody spotting using hidden markov models. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*. [Cited on page 53.]
- Dutta, S., Krishnaraj, S. P., & Murthy, H. A. (2015). Raga verification in Carnatic music using longest common segment set. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 605–611. Málaga, Spain. [Cited on pages 38, 39, 42, and 179.]
- Dutta, S. & Murthy, H. A. (2014a). Discovering typical motifs of a raga from one-liners of songs in Carnatic music. In *Int. Soc. for Music Information Retrieval (ISMIR)*, pp. 397–402. Taipei, Taiwan. [Cited on pages 31, 32, 34, 37, 101, and 135.]
- Dutta, S. & Murthy, H. A. (2014b). A modified rough longest common subsequence algorithm for motif spotting in an alapana of Carnatic music. In *Twentieth National Conference on Communications (NCC)*, pp. 1–6. [Cited on pages 31, 32, 34, 35, 36, 53, 54, and 135.]
- Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, vol. 1, pp. 452–455. [Cited on page 50.]
- Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3), 75–174. [Cited on pages 168 and 182.]
- Ganguli, K. K., Gulati, S., Serra, X., & Rao, P. (2016). Data-driven exploration of melodic structures in Hindustani music. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 605–611. New York, USA. [Cited on pages 216, 218, and 227.]
- Ganguli, K. K., Rastogi, A., Pandit, V., Kantan, P., & Rao, P. (2015). Efficient melodic query based audio search for Hindustani vocal compositions. In *In Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 591–597. Malaga, Spain. [Cited on pages 31, 32, 33, 34, 35, 36, and 54.]
- Ganti, T. (2013). *Bollywood: a guidebook to popular Hindi cinema*. Routledge. [Cited on pages 17 and 177.]
- Gedik, A. C. & Bozkurt, B. (2009). Evaluation of the makam scale theory of Arel for music information retrieval on traditional Turkish art music. *Journal of New Music Research*, 38(2), 103–116. [Cited on page 46.]

- Gedik, A. C. & Bozkurt, B. (2010). Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing*, 90(4), 1049–1063. [Cited on pages 46 and 47.]
- Ghias, A., Logan, J., Chamberlin, D., & Smith, B. C. (1995). Query by humming: musical information retrieval in an audio database. In *Proc. of the third ACM Int. Conf. on Multimedia*, pp. 231–236. ACM. [Cited on pages 47, 51, and 135.]
- Gómez, E. (2006). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3), 294–304. [Cited on pages 45, 46, 47, and 50.]
- Gómez, E. & Herrera, P. (2004). Estimating The Tonality Of Polyphonic Audio Files: Cognitive Versus Machine Learning Modelling Strategies. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 92–95. [Cited on page 46.]
- Gómez, E., Klapuri, A., & Meudic, B. (2003). Melody description and extraction in the context of music content processing. *Journal of New Music Research*, 32(1), 23–40. [Cited on pages 13 and 33.]
- Gómez, F., Pikrakis, A., Mora, J., Díaz-Báñez, J. M., Gómez, E., Escobar, F., Oramas, S., & Salamon, J. (2012). Automatic detection of melodic patterns in Flamenco singing by analyzing polyphonic music recordings. In *2nd International Workshop of Folk Music Analysis*, pp. 19–20. [Cited on pages 54 and 135.]
- Goto, M. (2006). A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Trans. on Audio, Speech and Language Processing*, 14(5), 1783–1794. [Cited on pages 47, 50, 51, and 147.]
- Griffith, R. T. H. (2004). *Hymns of the Samaveda*. Kessinger Publishing. [Cited on page 15.]
- Gulati, S. (2012). *A tonic identification approach for Indian art music*. Master's thesis, Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain. [Cited on pages 30, 80, and 99.]
- Gulati, S., Bellur, A., Salamon, J., Ranjani, H. G., Ishwar, V., Murthy, H. A., & Serra, X. (2014a). Automatic tonic identification in Indian art music: approaches and evaluation. *Journal of New Music Research*, 43(1), 53–71. [Cited on pages 8, 9, 23, 24, 31, 43, 87, and 88.]
- Gulati, S., Ganguli, K. K., Gupta, S., Srinivasamurthy, A., & Serra, X. (2015a). A lightweight real-time rāga recognition system for Indian art music. In *Late-Breaking Demo Session of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*. Malaga, Spain. [Cited on page 215.]

- Gulati, S., Salamon, J., & Serra, X. (2012). A two-stage approach for tonic identification in indian art music. In *Proc. of the 2nd CompMusic Workshop*, pp. 119–127. Istanbul, Turkey: Universitat Pompeu Fabra. [Cited on pages 23, 24, 25, 26, 27, 30, 88, and 248.]
- Gulati, S., Serrà, J., Ganguli, K. K., Şentürk, S., & Serra, X. (2016a). Time-delayed melody surfaces for rāga recognition. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 751–757. New York, USA. [Cited on pages 9, 85, and 177.]
- Gulati, S., Serrà, J., Ganguli, K. K., & Serra, X. (2014b). Landmark detection in Hindustani music melodies. In *International Computer Music Conference, Sound and Music Computing Conference*, pp. 1062–1068. Athens, Greece. [Cited on pages 9, 34, 87, 111, 136, and 140.]
- Gulati, S., Serrà, J., Ishwar, V., Şentürk, S., & Serra, X. (2016b). Phrase-based rāga recognition using vector space modeling. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 66–70. Shanghai, China. [Cited on pages 9, 84, 177, 178, 180, 184, 185, and 190.]
- Gulati, S., Serrà, J., Ishwar, V., & Serra, X. (2014c). Mining melodic patterns in large audio collections of Indian art music. In *Int. Conf. on Signal Image Technology & Internet Based Systems (SITIS-MIRA)*, pp. 264–271. Morocco. [Cited on pages 9, 87, 124, 147, and 148.]
- Gulati, S., Serrà, J., Ishwar, V., & Serra, X. (2016c). Discovering rāga motifs by characterizing communities in networks of melodic patterns. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 286–290. Shanghai, China. [Cited on pages 9 and 124.]
- Gulati, S., Serrà, J., & Serra, X. (2015b). An evaluation of methodologies for melodic similarity in audio recordings of Indian art music. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 678–682. Brisbane, Australia. [Cited on pages 9, 124, 126, 148, and 150.]
- Gulati, S., Serrà, J., & Serra, X. (2015c). Improving melodic similarity in Indian art music using culture-specific melodic characteristics. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 680–686. Spain. [Cited on pages 9, 124, and 137.]
- Gupta, C. & Rao, P. (2012). Objective assessment of ornamentation in Indian classical singing. In *Speech, Sound and Music Processing: Embracing Research in India*, pp. 1–25. Springer. [Cited on page 33.]
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *In Proc. of the 7th Python in Science Conf.*, pp. 11–15. Pasadena, CA USA. [Cited on pages 168 and 182.]

- Hastie, T., Tibshirani, R., & Friedman, J. (2009a). *The elements of statistical learning*. Berlin, Germany: Springer, 2nd edn. [Cited on pages 109, 115, and 183.]
- Hastie, T., Tibshirani, R., & Friedman, J. (2009b). Unsupervised learning. In *The elements of statistical learning*, pp. 485–585. Springer. [Cited on page 200.]
- Herley, C. (2006). ARGOS: automatically extracting repeating objects from multi-media streams. *IEEE Transactions on Multimedia*, 8(1), 115–129. [Cited on page 47.]
- Herrera-Boyer, P., Peeters, G., & Dubnov, S. (2003). Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1), 3–21. [Cited on page 108.]
- Hiraga, Y. (1997). Structural recognition of music by pattern matching. In *Proceedings of International Computer Music Conference*, pp. 426–429. [Cited on page 48.]
- Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 6(2), 65–70. [Cited on pages 116, 131, 143, 157, 184, and 198.]
- Hsu, J. L., Liu, C. C., & Chen, A. L. P. (2001). Discovering Nontrivial Repeating Patterns in Music Data. *IEEE Transactions on Multimedia*, 3(3), 311–325. [Cited on pages 47 and 147.]
- Huang, X., Acero, A., & Hon, H. W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR. [Cited on page 29.]
- Humphrey, E. J., Salamon, J., Nieto, O., Forsyth, J., Bittner, R., & Bello, J. P. (2014). JAMS: A JSON annotated music specification for reproducible MIR research. In *Int. Soc. for Music Info. Retrieval Conf.*, pp. 591–596. Taipei, Taiwan. [Cited on page 62.]
- Iliopoulos, C. S. & Kurokawa, M. (2002). String matching with gaps for musical melodic recognition. In *Stringology*, pp. 55–64. [Cited on page 53.]
- Ishwar, V., Bellur, A., & Murthy, H. A. (2012). Motivic analysis and its relevance to raga identification in Carnatic music. In *Proceedings of the 2nd CompMusic Workshop*, pp. 153–157. [Cited on pages 31, 32, 35, and 135.]
- Ishwar, V., Dutta, S., Bellur, A., & Murthy, H. (2013). Motif spotting in an Alapana in Carnatic music. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 499–504. [Cited on pages 31, 32, 34, 36, 81, 82, 101, 135, 142, 156, and 223.]
- Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1), 67–72. [Cited on page 57.]
- Izmirli, O. (2005). Template based key finding from audio. In *Proceedings of the International Computer Music Conference (ICMC)*, pp. 211–214. [Cited on page 46.]

- Janakiraman, S. R. (2008). *Essentials of musicology in south Indian music*. Madras: Indian Music Publishing House. [Cited on page 20.]
- Jang, J. S. R. & Gao, M. Y. (2000). A query-by-singing system based on dynamic programming. In *Proceedings of international workshop on intelligent system resolutions (8th bellman continuum)*, Hsinchu, pp. 85–89. [Cited on page 52.]
- Jang, J. S. R., Hsu, C. L., & Lee, H. R. (2005). Continuous HMM and its enhancement for singing/humming query retrieval. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 546–551. [Cited on page 53.]
- Jansen, A. & Niyogi, P. (2008). Modeling the temporal dynamics of distinctive feature landmark detectors for speech recognition. *Journal of the Acoustical Society of America*, 124(3), 1739–1758. [Cited on page 111.]
- Janssen, B., Haas, W. B. D., Volk, A., & Kranenburg, P. V. (2013). Discovering repeated patterns in music: state of knowledge, challenges, perspectives. In *Proc. of the 10th International Symposium on Computer Music Multidisciplinary Research*, pp. 225–240. Marseille. [Cited on pages 49 and 147.]
- Jha, R. (2001). *Abhinav Geetanjali Vol. I-V*. Sangeet Sadan. [Cited on page 71.]
- Juhász, Z. (2009). Motive identification in 22 folksong corpora using dynamic time warping and self organizing maps. In *Int. Soc. for Music Information Retrieval*, pp. 171–176. [Cited on page 135.]
- Kantz, H. & Schreiber, T. (2004). *Nonlinear time series analysis*. Cambridge, UK: Cambridge University Press. [Cited on page 193.]
- Karydis, I., Nanopoulos, A., & Manolopoulos, Y. (2006). Finding maximum-length repeating patterns in music databases. *Multimedia Tools and Applications*, 32(1), 49–71. [Cited on pages 49 and 163.]
- Kaul, D. M. (2007). *Hindustani and Persio-Arabian Music*. Kanishka Publishers, Distributors, first edn. [Cited on page 15.]
- Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3), 263–286. [Cited on page 52.]
- Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting time series: A survey and novel approach. *Data Mining in Time Series Databases*, 57, 1–22. [Cited on pages 112 and 113.]
- Keogh, E. & Ratanamahatana, C. A. (2004). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3), 358–386. [Cited on pages 56, 57, 58, 153, 154, and 249.]

- Kim, S. W., Park, S., & Chu, W. W. (2001). An index-based approach for similarity search supporting time warping in large sequence databases. In *17th International Conference on Data Engineering*, pp. 607–614. [Cited on pages 153, 154, and 249.]
- Klapuri, A. (2010). Pattern induction and matching in music signals. In *International Symposium on Computer Music Modeling and Retrieval*, pp. 188–204. [Cited on page 47.]
- Knopke, I. & Jürgensen, F. (2009). A System for Identifying Common Melodic Phrases in the Masses of Palestrina. *Journal of New Music Research*, 38(2), 171–181. [Cited on page 49.]
- Koduri, G. K., Gulati, S., & Rao, P. (2011). A survey of raaga recognition techniques and improvements to the state-of-the-art. In *Sound and Music Computing*. [Cited on pages 38, 39, 40, 43, and 44.]
- Koduri, G. K., Gulati, S., Rao, P., & Serra, X. (2012). Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4), 337–350. [Cited on pages 38, 39, 41, and 43.]
- Koduri, G. K., Ishwar, V., Serrà, J., & Serra, X. (2014). Intonation analysis of rāgas in Carnatic music. *Journal of New Music Research*, 43(1), 72–93. [Cited on pages 38, 39, 41, 43, 101, 184, and 248.]
- Kotsifakos, A., Papapetrou, P., Hollmén, J., & Gunopoulos, D. (2011). A subsequence matching with gaps-range-tolerances framework: a query-by-humming application. *Proceedings of the VLDB Endowment*, 4(11), 761–771. [Cited on page 53.]
- Kotsifakos, A., Papapetrou, P., Hollmén, J., Gunopoulos, D., & Athitsos, V. (2012). A survey of query-by-humming similarity methods. In *Proc. of the 5th Int. Conf. on Pervasive Tech. Related to Assistive Environments*, pp. 5:1–5:4. [Cited on pages 53 and 129.]
- Krishna, T. M. & Ishwar, V. (2012). Karnāṭic music: Svara, gamaka, motif and rāga identity. In *Proc. of the 2nd CompMusic Workshop*, pp. 12–18. [Cited on pages 19, 20, 21, 84, 172, 186, 188, and 189.]
- Krishnaswamy, A. (2003a). Application of pitch tracking to South Indian classical music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 557–560. [Cited on page 29.]
- Krishnaswamy, A. (2003b). On the twelve basic intervals in South Indian classical music. In *Audio Engineering Society Convention 115*. Audio Engineering Society. [Cited on page 29.]

- Kroher, N., Díaz-Báñez, J. M., Mora, J., & Gómez, E. (2016). Corpus COFLA: A research corpus for the computational study of Flamenco music. *Journal on Computing and Cultural Heritage (JOCCH)*, 9(2), 10:1–10:21. [Cited on page 62.]
- Kroher, N., Pikrakis, A., Moreno, J., & Díaz-Báñez, J. M. (2015). Discovery of repeated vocal patterns in polyphonic audio: A case study on flamenco music. In *23rd European Signal Processing Conference (EUSIPCO)*, pp. 41–45. [Cited on page 54.]
- Krumhansl, C. L. (2000). Tonality induction: A statistical approach applied cross-culturally. *Music Perception: An Interdisciplinary Journal*, 17(4), 461–479. [Cited on page 45.]
- Krumhansl, C. L. (2001). *Cognitive foundations of musical pitch*. Oxford University Press. [Cited on pages 45 and 46.]
- Krumhansl, C. L. & Kessler, E. J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review*, 89(4), 334–368. [Cited on pages 45 and 46.]
- Krumhansl, C. L. & Shepard, R. N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of experimental psychology: Human Perception and Performance*, 5(4), 579–594. [Cited on page 45.]
- Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79–86. [Cited on page 55.]
- Kumar, V., Pandya, H., & Jawahar, C. V. (2014). Identifying ragas in indian music. In *International Conference on Pattern Recognition*, pp. 767–772. [Cited on pages 38, 39, 41, and 43.]
- Lartillot, O. (2005a). An Efficient Algorithm For Motivic Pattern Extraction Based on a Cognitive Modeling. *Journées d'Informatiques Musicales*. [Cited on page 48.]
- Lartillot, O. (2005b). Multi-dimensional motivic pattern extraction founded on adaptive redundancy filtering. *Journal of New Music Research*, 34(4), 375–393. [Cited on pages 48, 147, and 151.]
- Lartillot, O. & Ayari, M. (2006). Motivic pattern extraction in music, and application to the study of Tunisian modal music. *South African Computer Journal*, 36, 16–28. [Cited on page 135.]
- Lartillot, O., Toiviainen, P., & Eerola, T. (2008). A matlab toolbox for music information retrieval. In *Data analysis, machine learning and applications*, pp. 261–268. Springer. [Cited on page 39.]

- Lee, K. (2006). Automatic chord recognition from audio using enhanced pitch class profile. In *International Computer Music Conference, (ICMC)*. [Cited on page 39.]
- Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge: MIT Press. [Cited on page 48.]
- Levitin, D. J. (2002). Memory for musical attributes. *Foundations of cognitive psychology: Core readings*, pp. 295–310. [Cited on page 14.]
- Levy, M. & Sandler, M. (2008). Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2), 318–326. [Cited on pages 50 and 116.]
- Lijffijt, J., Papapetrou, P., Hollmén, J., & Athitsos, V. (2010). Benchmarking dynamic time warping for music retrieval. In *Proceedings of the 3rd international conference on pervasive technologies related to assistive environments*, pp. 59:1–59:7. ACM. [Cited on page 52.]
- Lin, H. J., Wu, H. H., & Wang, C. W. (2011). Music matching based on rough longest common subsequence. *Journal of Information Science and Engineering*, 27(1), 95–110. [Cited on page 53.]
- Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11. New York, USA. [Cited on pages 33 and 147.]
- Longuet-Higgins, H. C. & Steedman, M. J. (1971). On interpreting bach. *Machine intelligence*, 6, 221–241. [Cited on page 45.]
- MacMullen, W. J. (2003). Requirements definition and design criteria for test corpora in information science. Tech. rep., School of Information and Library Science, University of North Carolina at Chapel Hill. [Cited on page 61.]
- Manikandan, T. V. (2004). *Lakshana and Laksya of Carnatic Music: A Quest*. New Delhi, India: Kanishka Publishers and Distributors. [Cited on page 29.]
- Mann, H. B. & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1), 50–60. [Cited on pages 116 and 157.]
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*, vol. 1. Cambridge university press Cambridge. [Cited on pages 131, 143, 157, and 158.]
- Marsden, A. (2012a). Counselling a better relationship between mathematics and musicology. *Journal of Mathematics and Music*, 6(2), 145–153. [Cited on pages 49 and 162.]

- Marsden, A. (2012b). Interrogating Melodic Similarity: A Definitive Phenomenon or the Product of Interpretation? *Journal of New Music Research*, 41(4), 323–335. [Cited on pages 48 and 135.]
- Martinez, J. L. (2001). *Semiosis in Hindustani Music*. Motilal Banarsi Dass Publishers. [Cited on pages 6 and 17.]
- Maslov, S. & Sneppen, K. (2002). Specificity and stability in topology of protein networks. *Science*, 296(5569), 910–913. [Cited on pages 167 and 181.]
- Mazzoni, D. & Dannenberg, R. B. (2001). Melody matching directly from audio. In *2nd Annual International Symposium on Music Information Retrieval*, pp. 17–18. [Cited on pages 51, 52, 129, and 135.]
- McCallum, A. & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI Workshop on learning for text categorization*, vol. 752, pp. 41–48. [Cited on page 186.]
- McNab, R. J., Smith, L. A., & Witten, I. H. (1996). Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the first ACM international conference on Digital libraries*, pp. 11–18. [Cited on page 51.]
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157. [Cited on pages 184 and 198.]
- Meer, W. V. D. (1980). Hindustani music in the twentieth century. [Cited on pages 21, 23, and 84.]
- Mehta, R. (2008). *Indian Classical Music and Gharana Tradition*. Readworthy Publications Pvt. Ltd., first edn. [Cited on page 16.]
- Meredith, D. (2006). Point-set algorithms for pattern discovery and pattern matching in music. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. [Cited on page 48.]
- Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4), 321–345. [Cited on pages 48, 49, 135, 147, and 163.]
- Mitchell, T. M. (1997). *Machine Learning*. New York, USA: McGraw-Hill. [Cited on pages 184, 196, 197, and 199.]
- Moelants, D., Cornelis, O., & Leman, M. (2009). Exploring african tone scales. In *10th International Society for Music Information Retrieval Conference (ISMIR-2009)*, pp. 489–494. International Society for music Information Retrieval. [Cited on pages 47 and 101.]

- Mueen, A., Keogh, E., Zhu, Q., Cash, S., & Westover, B. (2009). Exact discovery of time series motifs. In *Proc. of SIAM Int. Con. on Data Mining (SDM)*, pp. 1–12. [Cited on pages 147 and 148.]
- Müller, M. (2007). *Dynamic Time Warping*, pp. 69–84. Berlin, Heidelberg: Springer. [Cited on pages 56 and 57.]
- Müller, M., Grosche, P., & Wiering, F. (2009). Robust segmentation and annotation of folk song recordings. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 735–740. [Cited on page 151.]
- Muller, M., Jiang, N., & Grosche, P. (2013). A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Transactions on audio, speech, and language processing*, 21(3), 531–543. [Cited on page 51.]
- Müller, M. & Kurth, F. (2006). Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Advances in Signal Processing*, 2007(1), 1–18. [Cited on page 51.]
- Muscarello, A., Gravier, G., & Bimbot, F. (2009). Variability tolerant audio motif discovery. In *International Conference on Multimedia Modeling*, pp. 275–286. Springer. [Cited on page 226.]
- Narayan, A. & Singh, N. (2014). Detection of micro-tonal ornaments in dhrupad using dynamic programming approach. In *Proceedings of the 9th Conference on Interdisciplinary Musicology—CIM14. Berlin, Germany*, pp. 388–391. [Cited on page 33.]
- Narayanaswami, P. P. & Jayaraman, V. (2011). Sangita Sampradaya Pradarsini. [Cited on page 20.]
- Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), 443–453. [Cited on page 54.]
- Newman, M. E. (2003). The structure and function of complex networks. *Society for Industrial and Applied Mathematics (SIAM) review*, 45(2), 167–256. [Cited on pages 167 and 170.]
- Nieto, O. & Farbood, M. M. (2012). Perceptual evaluation of automatically extracted musical motives. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pp. 723–727. [Cited on page 49.]
- Nieto, O., Humphrey, E. J., & Bello, P. B. (2012). Compressing music recordings into audio summaries. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 313–318. [Cited on pages 47 and 51.]

- Noland, K. & Sandler, M. B. (2006). Key estimation using a hidden markov model. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 121–126. [Cited on page 46.]
- Ong, B. S. & Herrera, P. (2005). Semantic segmentation of music audio contents. In *Proc. of the Int. Computer Music Conf. (ICMC)*. [Cited on page 116.]
- Orio, N. (2006). Music retrieval: A tutorial and review. *Foundations and Trends® in Information Retrieval*, 1(1), 1–90. [Cited on page 3.]
- Pachet, F. & Aucouturier, J. J. (2004). Improving timbre similarity: How high is the sky. *Journal of negative results in speech and audio sciences*, 1(1), 1–13. [Cited on page 3.]
- Paiva, R. P., Mendes, T., & Cardoso, A. (2006). Melody detection in polyphonic musical signals: Exploiting perceptual rules, note salience, and melodic smoothness. *Computer Music Journal*, 30(4), 80–98. [Cited on page 14.]
- Pandey, G., Mishra, C., & Ipe, P. (2003). Tansen: A system for automatic raga identification. In *In Proc. of the 1st Indian Int. Conf. on Artificial Intelligence*, pp. 1350–1363. [Cited on pages 38, 39, 41, 42, and 43.]
- Papadopoulos, H. & Peeters, G. (2007). Large-scale study of chord estimation algorithms based on chroma representation and hmm. In *2007 International Workshop on Content-Based Multimedia Indexing*, pp. 53–60. [Cited on page 46.]
- Pardo, B. & Birmingham, W. P. (2002). Encoding timing information for musical query matching. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 267—268. [Cited on page 52.]
- Park, A. S. & Glass, J. R. (2008). Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1), 186–197. [Cited on page 47.]
- Patel, A. D. (2007). *Music, language, and the brain*. Oxford, UK: Oxford University Press. [Cited on page 109.]
- Paulus, J. & Klapuri, A. (2006). Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pp. 59–68. ACM. [Cited on page 46.]
- Paulus, J., Müller, M., & Klapuri, A. (2010). Audio-based music structure analysis. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 625–636. [Cited on pages 50 and 147.]
- Pauws, S. (2004a). Cubyhum: Algorithms for query by humming. In *Algorithms in Ambient Intelligence*, pp. 71–87. Springer. [Cited on page 51.]

- Pauws, S. (2004b). Musical key extraction from audio. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 96–99. [Cited on page 46.]
- Pearce, M., Müllensiefen, D., & Wiggins, G. A. (2008). A comparison of statistical and rule-based models of melodic segmentation. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 89–94. [Cited on page 151.]
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. [Cited on pages 109, 115, and 183.]
- Peeters, G. (2006a). Chroma-based estimation of musical key from audio-signal analysis. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 115–120. [Cited on page 46.]
- Peeters, G. (2006b). Musical key estimation of audio signal based on hidden markov modeling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pp. 127–131. [Cited on page 46.]
- Peeters, G. & Fort, K. (2012). Towards a (better) definition of the description of annotated MIR corpora. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 25–30. [Cited on page 62.]
- Perng, C. S., Wang, H., Zhang, S. R., & Parker, D. S. (2000). Landmarks: a new model for similarity-based pattern querying in time series databases. In *Proc. of the Int. Conf. on Data Engineering (ICDE)*, pp. 33–42. [Cited on page 111.]
- Pikrakis, A., Gómez, F., Oramas, S., Díaz-Báñez, J. M., Mora, J., Escobar-Borrego, F., Gómez, E., & Salamon, J. (2012). Tracking melodic patterns in Flamenco singing by analyzing polyphonic music recordings. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 421–426. [Cited on pages 53, 54, 101, and 135.]
- Pikrakis, A., Kroher, N., & Díaz-Báñez, J. M. (2016). Detection of melodic patterns in automatic transcriptions of Flamenco singing. In *6th International Workshop on Folk Music Analysis*, pp. 14–17. Dublin: Dublin Institute of Technology. [Cited on pages 53, 54, and 135.]
- Pikrakis, A., Theodoridis, S., & Kamarotos, D. (2003). Recognition of isolated musical patterns using context dependent dynamic time warping. *IEEE Transactions on Speech and Audio Processing*, 11(3), 175–183. [Cited on pages 53, 54, 101, and 135.]
- Porter, A., Bogdanov, D., Kaye, R., Tsukanov, R., & Serra, X. (2015). Acousticbrainz: a community platform for gathering music information obtained from audio. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 786–792. [Cited on page 62.]

- Porter, A., Sordo, M., & Serra, X. (2013). Dunya: A system for browsing audio music collections exploiting cultural context. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 101–106. Curitiba, Brazil. [Cited on page 208.]
- Powers, H. S. (1959). *The background of the South Indian Rāga-System*. Ph.D. thesis, Princeton University, Barcelona, Spain. [Cited on pages 6 and 18.]
- Pratyush (2010). *Analysis and Classification of Ornaments in North Indian (Hindustani) Classical Music*. Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain. [Cited on page 33.]
- Quinlan, J. R. (1993). *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. [Cited on page 30.]
- Rabiner, L. & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice hall. [Cited on page 57.]
- Raja, D. S. (2012). *Hindustani Music Today*. D.K. Printworld (P) Ltd., first edn. [Cited on page 15.]
- Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., & Keogh, E. (2013). Addressing big data time series: mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3), 10:1–10:31. [Cited on pages 147, 154, and 249.]
- Ramanathan, N. (1999). *Musical forms in the Sangita Ratnakara*. Chennai: Sampradaaya. [Cited on page 20.]
- Ranjani, H. G., Arthi, S., & Sreenivas, T. V. (2011). Carnatic music analysis: Shadja, swara identification and raga verification in alapana using stochastic models. In *IEEE Workshop on applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 29–32. [Cited on pages 23, 24, 25, 29, 31, 38, 39, 40, 41, 43, 80, 88, 99, and 248.]
- Rao, P., Ross, J. C., & Ganguli, K. K. (2013). Distinguishing raga-specific intonation of phrases with audio analysis. *Ninād, Journal of ITC SRA*, 26, 64. [Cited on pages 31, 32, and 34.]
- Rao, P., Ross, J. C., Ganguli, K. K., Pandit, V., Ishwar, V., Bellur, A., & Murthy, H. A. (2014). Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research*, 43(1), 115–131. [Cited on pages 31, 32, 34, 35, 37, 82, 101, 126, 127, 135, 137, 142, 146, and 156.]
- Rao, S., Bor, J., van der Meer, W., & Harvey, J. (1999). *The raga guide: a survey of 74 Hindustani ragas*. Nimbus Records with Rotterdam Conservatory of Music. [Cited on pages 5, 20, 196, and 226.]

- Rao, S. & Rao, P. (2014). An overview of Hindustani music in the context of computational musicology. *Journal of New Music Research*, 43(1), 24–33. [Cited on page 20.]
- Rao, T. K. G. (1995a). *Compositions of Muddusvami Dikshitar*. Chennai, India: Ganamandir Publications. [Cited on page 66.]
- Rao, T. K. G. (1995b). *Compositions of Tyagaraja*. Chennai, India: Ganamandir Publications. [Cited on page 66.]
- Rao, T. K. G. (1997). *Compositions of Syama Sastri*. Chennai, India: Ganamandir Publications. [Cited on page 66.]
- Rao, V., Pant, S., Bhaskar, M., & Rao, P. (2009). Applications of a semiautomatic melody extraction interface for Indian music. In *International Symposium on Frontiers of Research in Speech and Music (FRSM)*. [Cited on page 81.]
- Rao, V. & Rao, P. (2009). Improving polyphonic melody extraction by dynamic programming based dual f0 tracking. In *12th International Conference on Digital Audio Effects (DAFx)*, pp. 78–84. Como, Italy. [Cited on page 39.]
- Rao, V. & Rao, P. (2010). Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8), 2145–2154. [Cited on pages 39 and 81.]
- Ratanamahatana, C. A. & Keogh, E. (2004). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*, pp. 22–25. [Cited on page 130.]
- Rockstro, W. S., Dyson, G., Drabkin, W., Powers, H. S., & Rushton, J. (2001). Cadence. In L. Macy (Ed.) *Grove music online*. Oxford University Press. [Cited on page 109.]
- Rodríguez L., M., Volk, A., & de Haas, B. (2014). Comparing repetition-based melody segmentation models. In *Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM14)*, pp. 143–148. [Cited on pages 34, 151, and 226.]
- Rolland, P. Y. (1999). Discovering patterns in musical sequences. *Journal of New Music Research*, 28(4), 334–350. [Cited on page 49.]
- Ross, J. C. & Rao, P. (2012). Detection of raga-characteristic phrases from Hindustani classical music audio. In *Proc. of 2nd CompMusic Workshop*, pp. 133–138. [Cited on pages 31, 32, 34, 36, 110, 111, 127, 135, and 136.]
- Ross, J. C., Vinutha, T. P., & Rao, P. (2012). Detecting melodic motifs from audio for Hindustani classical music. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 193–198. [Cited on pages 31, 32, 33, 34, 35, 36, 53, 54, 82, 102, 127, 128, 135, 136, 142, and 223.]

- Sakoe, H. & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech, and Language Processing*, 26(1), 43–50. [Cited on pages 56, 57, 58, 129, and 141.]
- Salamon, J. (2013). *Melody Extraction from Polyphonic Music Signals*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain. [Cited on pages 13 and 101.]
- Salamon, J. & Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6), 1759–1770. [Cited on pages 25, 26, 39, 42, 101, 127, and 249.]
- Salamon, J., Gómez, E., & Bonada, J. (2011). Sinusoid extraction and salience function design for predominant melody estimation. In *Proc. of Int. Conf. on Digital Audio Effects (DAFx)*, pp. 73–80. [Cited on pages 25 and 26.]
- Salamon, J., Gulati, S., & Serra, X. (2012). A multipitch approach to tonic identification in Indian classical music. In *Proc. of Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pp. 499–504. Porto, Portugal. [Cited on pages 23, 24, 25, 26, 27, 30, 80, 88, 98, and 248.]
- Salamon, J., Serrà, J., & Gómez, E. (2013). Tonal representations for music retrieval: from version identification to query-by-humming. *International Journal of Multimedia Information Retrieval*, 2(1), 45–58. [Cited on page 53.]
- Sambamoorthy, P. (1998). *South Indian music vol. I-VI*. The Indian Music Publishing House. [Cited on page 109.]
- Saraf, R. (2011). *Development of Hindustani Classical Music (19th & 20th centuries)*. Vidyavidhi Prakashan, first edn. [Cited on pages 15 and 16.]
- Schenker, H., Jonas, O., & B., E. M. (1980). *Harmony*. Neue musikalische Theorien und Phantasien. University of Chicago Press. [Cited on pages 6 and 121.]
- Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. *Computing in musicology: a directory of research*, (11), 3–64. [Cited on page 1.]
- Sengupta, R. (1990). Study on some aspects of the singer's formant in north Indian classical singing. *Journal of Voice*, 4(2), 129–134. [Cited on page 39.]
- Sengupta, R., Dey, N., Nag, D., Datta, A. K., & Mukerjee, A. (2005). Automatic tonic (SA) detection algorithm in Indian classical vocal music. In *National Symposium on Acoustics*, pp. 1–5. [Cited on pages 19, 23, 24, 25, 26, 27, 30, and 88.]
- Serrà, J. (2011). *Identification of versions of the same musical composition by processing audio descriptions*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona. [Cited on page 46.]

- Serrà, J., Corral, A., Boguñá, M., Haro, M., & Arcos, J. L. (2012). Measuring the evolution of contemporary western popular music. *Scientific reports*, 2, 521. [Cited on page 62.]
- Serrà, J., Müller, M., Grosche, P., & Arcos, J. L. I. (2012). Unsupervised detection of music boundaries by time series structure features. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1613–1619. AAAI Press. [Cited on pages 47 and 147.]
- Serrà, J., Müller, M., Grosche, P., & Arcos, J. L. I. (2014). Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, 16(5), 1229–1240. [Cited on pages 47, 50, and 51.]
- Serra, X. (2011). A multicultural approach to music information research. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 151–156. [Cited on pages 3 and 126.]
- Serra, X. (2014). Creating research corpora for the computational study of music: the case of the Compmusic project. In *Proc. of the 53rd AES Int. Conf. on Semantic Audio*. London. [Cited on pages 61, 62, and 63.]
- Serra, X., Magas, M., Benetos, E., Chudy, M., Dixon, S., Flexer, A., Gómez, E., Gouyon, F., Herrera, P., Jordà, S., Paytuvi, O., Peeters, G., Schluter, J., Vinet, H., & Widmer, G. (2013). Roadmap for music information research. Creative Commons BY-NC-ND 3.0 license. [Cited on page 3.]
- Serrano, M. A., Boguñá, M., & Vespignani, A. (2009). Extracting the multiscale backbone of complex weighted networks. in *Proc. of the National Academy of Sciences of the USA*, 106(16), 6483–6488. [Cited on page 166.]
- Shetty, S. & Achary, K. K. (2009). Raga mining of Indian music by extracting arohana-avarohana pattern. *Int. Journal of Recent Trends in Engineering*, 1(1), 362–366. [Cited on pages 38, 39, 41, 42, and 43.]
- Singh, J. (1995). *Indian Music*. Munshiram Manoharlal Publishers Pvt Ltd, first edn. [Cited on page 15.]
- Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, 147(1), 195–197. [Cited on pages 35 and 53.]
- Sridhar, R. & Geetha, T. V. (2006). Swara identification for south indian classical music. In *9th International Conference on Information Technology (ICIT)*, pp. 143–144. [Cited on page 39.]
- Sridhar, R. & Geetha, T. V. (2009). Raga identification of Carnatic music for music information retrieval. *International Journal of Recent Trends in Engineering*, 1(1), 571–574. [Cited on pages 38, 39, 41, and 42.]

- Srinivasamurthy, A., Koduri, G. K., Gulati, S., Ishwar, V., & Serra, X. (2014). Corpora for music information research in Indian art music. In *Int. Computer Music Conf./Sound and Music Computing Conf.*, pp. 1029–1036. Athens, Greece. [Cited on pages 63 and 67.]
- Srinivasamurthy, A. & Serra, X. (2014). A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5217–5221. IEEE. [Cited on pages 34 and 136.]
- Sturm, B. L. (2012). A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pp. 29–66. Springer. [Cited on page 62.]
- Subramanian, S. K., Wyse, L., & Mcgee, K. (2012). A Two-Component Representation For Modeling Gamakas Of Carnatic Music. In *Proceedings of the 2nd CompMusic Workshop, Istanbul, Turkey*, pp. 147–152. [Cited on page 33.]
- Sun, X. (2000). A pitch determination algorithm based on subharmonic-to-harmonic ratio. In *the 6th International Conference of Spoken Language Processing*, vol. 4, pp. 676–679. [Cited on page 39.]
- Takens, F. (1981). Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Warwick 1980*, pp. 366–381. [Cited on pages 191 and 193.]
- Tanaka, Y., Iwamoto, K., & Uehara, K. (2005). Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3), 269–300. [Cited on page 33.]
- Temperley, D. (1999). What's key for key? the krumhansl-schmuckler key-finding algorithm reconsidered. *Music Perception: An Interdisciplinary Journal*, 17(1), 65–100. [Cited on page 45.]
- Temperley, D. & Marvin, E. W. (2008). Pitch-class distribution and the identification of key. *Music Perception: An Interdisciplinary Journal*, 25(3), 193–212. [Cited on page 45.]
- Tenney, J. & Polansky, L. (1980). Temporal gestalt perception in music. *Journal of Music Theory*, 24(2), 205–241. [Cited on page 48.]
- Trivedi, R. (Ed.) (2008). *Bharatiya Shastriya Sangit: Shastra, Shikshan Va Prayog*. Sahitya Sangam, New 100, Lookerganj, Allahabad. India, first edn. [Cited on page 15.]
- Uitdenbogerd, A. & Zobel, J. (1999). Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pp. 57–66. ACM. [Cited on pages 52 and 53.]

- Uitdenbogerd, A. L. & Zobel, J. (1998). Manipulation of music for melody matching. In *Proceedings of the sixth ACM international conference on Multimedia*, pp. 235–240. ACM. [Cited on page 51.]
- Viswanathan, T. & Allen, M. H. (2004). *Music in South India*. Oxford University Press. [Cited on pages 15, 16, 18, 20, 65, 160, and 200.]
- Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 216–225. ACM. [Cited on page 153.]
- Widdess, R. (1994). Involving the performers in transcription and analysis: a collaborative approach to Dhrupad. *Ethnomusicology*, 38(1), 59–79. [Cited on pages 5, 33, 42, 50, and 226.]
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, pp. 80–83. [Cited on pages 131 and 143.]
- Zhu, Y. & Shasha, D. (2003a). Query by humming: a time series database approach. In *Proc. of SIGMOD*, p. 675. [Cited on pages 52, 129, and 130.]
- Zhu, Y. & Shasha, D. (2003b). Warping indexes with envelope transforms for query by humming. In *proc. of the ACM SIGMOD Int. Conf. on Management of data*, pp. 181–192. [Cited on page 155.]