pled at 44.1 kHz. The tracks procured from personal collections have been grouped into musically meaningful short collections and then released as albums. Each recording in the collection has the following accompanying metadata: rāg, tāl, lay/s, artists, form, and if applicable, the bandiś and the composer. It has manually annotated time aligned characteristic melodic phrases and lay based sections. In addition, it has semi-automatically extracted tonic, vocal pitch track, tempo, and time aligned sam annotations.

The Creative Commons collections are useful for several MIR tasks. From a rhythm analysis perspective, the collection is useful for meter inference and tracking, rhythmic and percussion pattern analysis, and rhythm based structural segmentation. To the best of our knowledge, this collection is the largest tāḷa and sama annotated music collection of Indian Art Music.

4.2 Test datasets

For each dataset below, present a basic statistical analysis, links to download. Mention which tasks they are the most useful for. The limitations of the dataset and what can be improved as well.

The test datasets are designed for specific tasks and contain additional information such as annotations and derived data. They are useful for various melody and rhythm analysis tasks. We describe only those test datasets that are useful with rhythm analysis tasks. We describe each dataset briefly emphasizing the primary research task where they can be used.

4.2.1 Carnatic music rhythm dataset

The Carnatic music rhythm dataset (CMR$_f$ dataset)[22] is a rhythm annotated test corpus for many automatic rhythm analysis tasks in Carnatic Music. The collection consists of audio excerpts from the Carnatic research corpus, manually annotated time aligned markers indicating the progression through the tāḷa cycle, and the associated tāḷa related metadata. The dataset has pieces in four popular tāḷas (Table 4.5) that encompass a majority of current day Carnatic music performance. The pieces include a mix of vocal and instrumental recordings, recent and old recordings, and span a wide variety

[22] http://compmusic.upf.edu/carnatic-rhythm-dataset

Tāḷa	# Pieces	Total Duration	$\overline{T_f}$	#Ann.	#Sama
		hours (min)	min		
Ādi	50	4.21 (252.78)	4.85	22793	2882
Rūpaka	50	4.45 (267.45)	4.62	22668	7582
Miśra chāpu	48	5.70 (342.13)	6.59	54309	7795
Khaṇḍa chāpu	28	2.24 (134.62)	4.41	21382	4387
Total	176	16.61 (996.98)	5.06	121602	22646

Table 4.5: CMR$_\text{f}$ dataset showing the total duration and number of annotations. #Sama shows the number of sama annotations and #Ann. shows the number of beat annotations (including samas). $\overline{T_f}$ indicates the median piece length in the dataset.

Tāḷa	$\overline{\tau_s} \pm \sigma_s$	$\overline{\tau_o} \pm \sigma_o$	$[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$
Ādi	5.34 ± 0.723	0.167 ± 0.023	[2.88, 7.07]
Rūpaka	2.13 ± 0.239	0.178 ± 0.020	[1.21, 3.10]
Miśra chāpu	2.67 ± 0.358	0.191 ± 0.026	[1.63, 3.65]
Khaṇḍa chāpu	1.85 ± 0.284	0.185 ± 0.028	[0.91, 2.87]

Table 4.6: Tāḷa cycle length indicators for CMR$_\text{f}$ dataset. $\overline{\tau_s}$ and σ_s indicate the mean and standard deviation of the median inter-sama interval of the pieces, respectively. $\overline{\tau_o}$ and σ_o indicate the mean and standard deviation of the median inter-akṣara interval of the pieces, respectively. $[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$ indicate the minimum and maximum value of τ_s and hence the range of τ_s in the dataset. All values in the table are in seconds.

of forms. All pieces have a percussion accompaniment, predominantly Mridangam. There are also several different pieces by the same artist (or release group), and multiple instances of the same composition rendered by different artists. Each piece is uniquely identified using the MBID of the recording. The pieces are mono WAV files downmixed from stereo recordings, and sampled at 44.1 kHz. The audio is also available as downmixed mono WAV files for experiments. The audio files are full length pieces or a part of the full length pieces. Of the 176 audio files, 120 are full length pieces.

Tāḷa	# Pieces	Total Duration hours (min)	#Ann.	#Sama
Ādi	30	0.98 (58.87)	5452	696
Rūpaka	30	1.00 (60.00)	5148	1725
Miśra chāpu	30	1.00 (60.00)	8992	1299
Khaṇḍa chāpu	28	0.93 (55.93)	9133	1840
Total	118	3.91 (234.80)	28725	5560

Table 4.7: CMR dataset showing the total duration and number of annotations. #Sama shows the number of sama annotations and #Ann. shows the number of beat annotations (including samas).

There are several annotations that accompany each excerpt in the dataset. The primary annotations are audio synchronized time-stamps indicating the different metrical positions in the tāḷa cycle - the sama (downbeat) and other beats shown with numerals in Figure 2.1. The annotations were created using Sonic Visualizer (Cannam, Landone, & Sandler, 2010) by tapping to music and manually correcting the taps. The annotations have been verified by a professional Carnatic musician. Each annotation has a time-stamp and an associated numeric label that indicates the position of the beat marker in the tāḷa cycle. In addition, for each excerpt, the tāḷa of the piece and eḍupu (offset of the start of the piece, relative to the sama) are recorded. The possibly time varying tempo of a piece can be obtained using the beat and sama annotations.

Carnatic music rhythm dataset (CMR$_{\mathrm{f}}$) dataset is described in Table 4.5, showing the four tāḷas and the number of pieces for each tāḷa. The total duration of audio in the dataset is over 16.6 hours, with 121062 time-aligned beat annotations. The median length of a piece is about 5 minutes in the dataset. Table 4.6 shows a basic statistical analysis of the tāḷa cycle length indicators in the dataset, which is useful to understand the tempo characteristics and the range of the metrical cycle lengths in the dataset. The length of the tāḷa cycle is indicative of the tempo of the piece. Despite no notated tempo, we can see from the values of the median inter-akṣara interval, $\overline{\tau_o}$ and its standard deviation that the tempo in Carnatic music does not vary much across the talas. The range of $\overline{\tau_s}$ values show that a wide range of tempi are present in Carnatic music

Tāḷa	$\overline{\tau_s} \pm \sigma_s$	$\overline{\tau_o} \pm \sigma_o$	$[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$
Ādi	5.32 ± 0.868	0.17 ± 0.027	[2.88, 7.07]
Rūpaka	2.12 ± 0.225	0.18 ± 0.019	[1.40, 3.10]
Miśra chāpu	2.81 ± 0.272	0.20 ± 0.019	[2.03, 3.65]
Khaṇḍa chāpu	1.87 ± 0.290	0.19 ± 0.029	[1.00, 2.84]

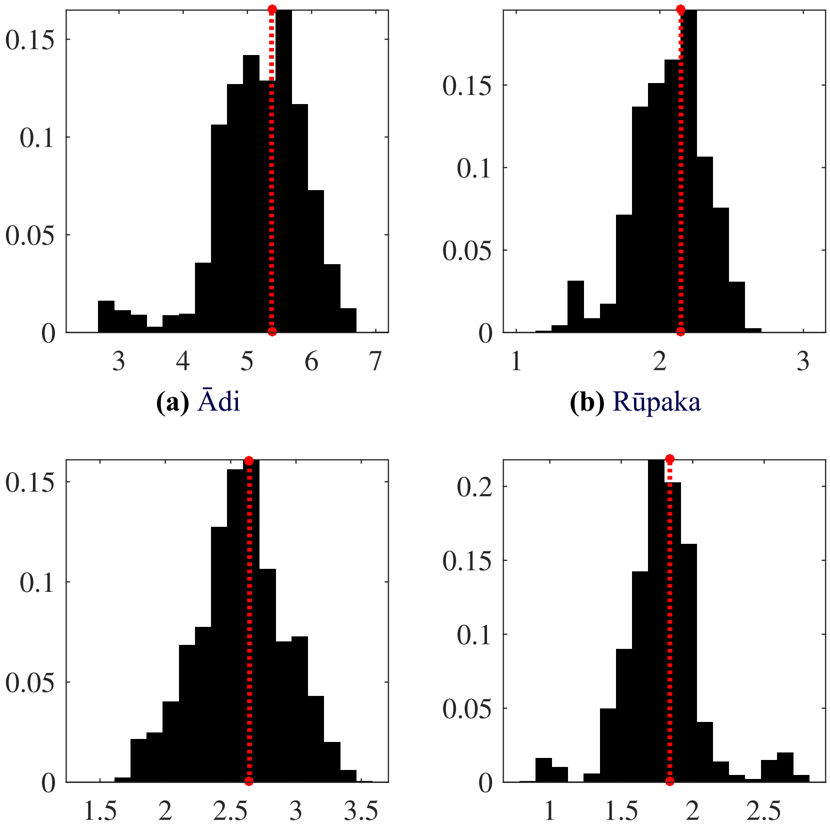
Table 4.8: Tāḷa cycle length indicators for CMR dataset. $\overline{\tau_s}$ and σ_s indicate the mean and standard deviation of the median inter-sama interval of the pieces, respectively. $\overline{\tau_o}$ and σ_o indicate the mean and standard deviation of the median inter-akṣara interval of the pieces, respectively. $[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$ indicate the minimum and maximum value of τ_s and hence the range of τ_s in the dataset. All values in the table are in seconds.

pieces, often over two tempo octaves. The shortest cycle in the dataset is less than second long, while the longest cycle is over 7 seconds long.

A representative subset of the CMR_f dataset is also compiled as Carnatic music rhythm dataset (subset) (CMR), with two minute excerpts of pieces in CMR_f (or the full piece if the piece is shorter than 2 minutes). These short excerpts additionally contain all the annotations of the full dataset, including time aligned sama and beat annotations. The smaller Carnatic music rhythm dataset (subset) (CMR) dataset will be useful for faster testing of approaches and algorithms.

The smaller subset CMR dataset is described in Table 4.7, showing the four tāḷas and the number of pieces for each tāḷa. The total duration of audio in the dataset is about 4 hours, with 28725 time-aligned beat annotations. Table 4.8 shows a basic statistical analysis of the tāḷa cycle length indicators in the CMR dataset, which are similar to the indicators of CMR_f dataset shown in Table 4.6, showing that CMR dataset is a representative subset of CMR_f dataset.

The tempo values are not notated in Carnatic music, and the pieces are not played to a metronome. Hence the tempo varies over a piece in time. Hence, in addition to the median values tabulated in Table 4.6 we present further analysis of the inter-sama interval (τ_s) and inter-beat interval (τ_b) for each tāḷa over the whole CMR_f dataset. A histogram of τ_s and τ_b for each tāḷa is shown in Fig-



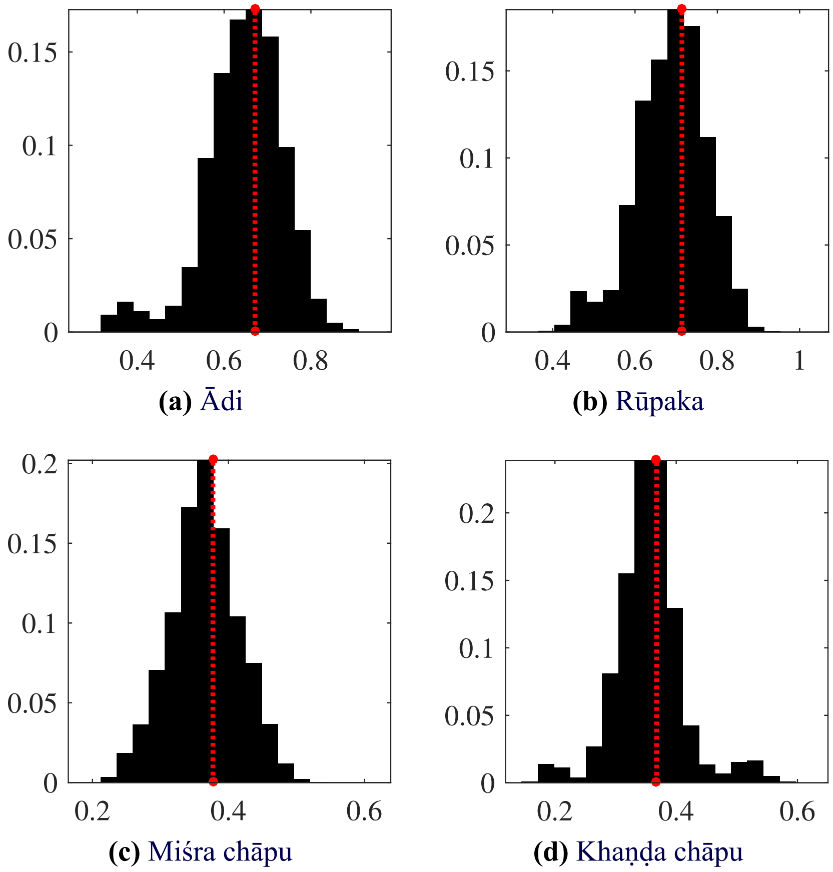
(a) Ādi **(b)** Rūpaka

(c) Miśra chāpu **(d)** Khaṇḍa chāpu

Figure 4.3: A histogram of the inter-sama interval τ_s in the $\texttt{CMR}_\texttt{f}$ dataset for each tāḷa. The ordinate is the fraction of the total count corresponding to the τ_s value shown in abscissa. The median τ_s for each tāḷa is shown as a red dotted line.

ure 4.3 and Figure 4.4 respectively. This shows the distribution of cycle lengths in the dataset over the whole range of τ_s for each tāḷa, around the median value. Despite the large range of τ_s values, the distribution in Figure 4.3 and Figure 4.4 show that the tempo often is limited to a small range of values. Though the musicians are free to choose any tempo, we empirically observe that they tend to choose a narrow range of tempo.

To illustrate and measure the time varying tempo of music pieces in Carnatic music, we normalize all the τ_s and τ_b values in a piece by the median in the piece to obtain median normalized τ_s and τ_b values, a histogram of which is shown in Figure 4.5 and Figure 4.6,



(a) Ādi **(b)** Rūpaka

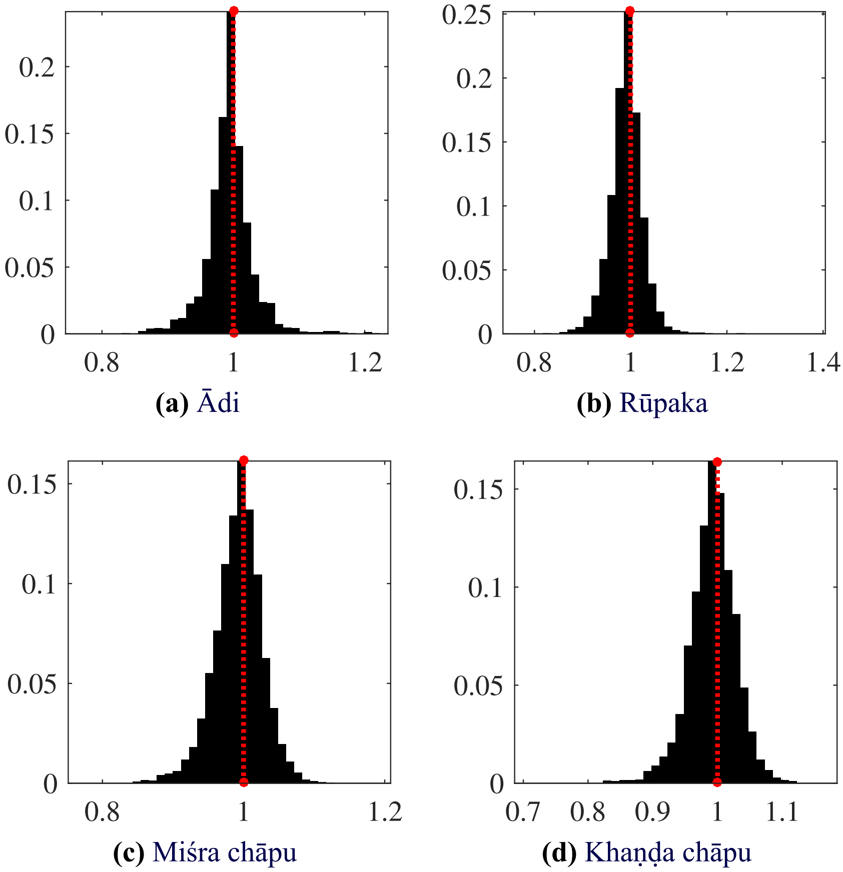
(c) Miśra chāpu **(d)** Khaṇḍa chāpu

Figure 4.4: A histogram of the inter-beat interval τ_b in the $\mathtt{CMR}_\mathrm{f}$ dataset for each tāḷa. The ordinate is the fraction of the total count corresponding to the τ_b value shown in abscissa. The median τ_b for each tāḷa is shown as a red dotted line.

respectively. These histograms are centered around 1, since they are normalized by the median, and the spread of these histograms around the value of 1 is a measure of deviation of tempo from the median value. From the figures, it is clear that the tempo is time varying but with less than about 20% maximum deviation from the median tempo of the piece for all tāḷas.

Rhythm patterns in $\mathtt{CMR}_\mathrm{f}$ and $\mathtt{CMR}$ datasets

With a sizeable annotated corpus of Carnatic music, we can do corpora level analysis of patterns in rhythm and percussion. The idea



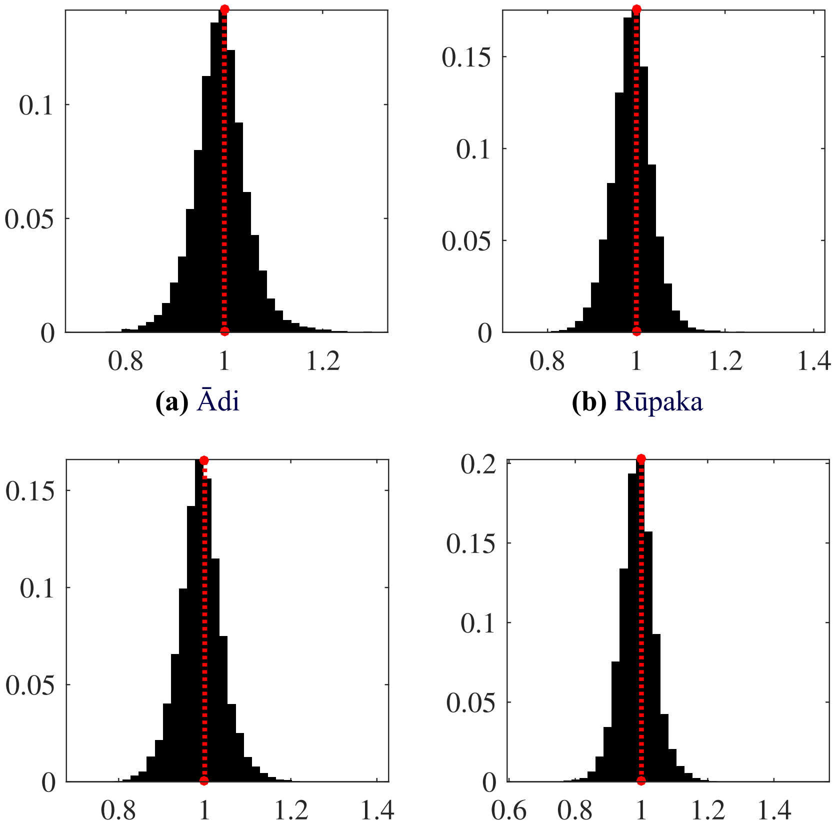
(a) Ādi **(b)** Rūpaka

(c) Miśra chāpu **(d)** Khaṇḍa chāpu

Figure 4.5: A histogram of the median normalized inter-sama interval τ_s in the $\texttt{CMR}_\text{f}$ dataset for each tāḷa. The ordinate is the fraction of the total count corresponding to the normalized τ_s value shown in abscissa.

is to showcase these patterns as the potential of dataset analysis, while showing their utility for meter tracking, musicology, performance analysis, comparative analysis.

The aim here is not to seek all musicological insights from data, but to illustrate the possibilities of a corpus level analysis data, and how such analysis tools can help aid and advance musicology. The MIR applications of such datasets is the primary goal of the thesis and discussed in subsequent chapters. Hence, an example of corpus level musicological analysis is presented in this chapter, which amounts to a performance analysis of music in current practice from



(a) Ādi **(b)** Rūpaka

(c) Miśra chāpu **(d)** Khaṇḍa chāpu

Figure 4.6: A histogram of the median normalized inter-beat interval τ_b in the $\mathtt{CMR}_\mathrm{f}$ dataset for each tāḷa. The ordinate is the fraction of the total count corresponding to the normalized τ_b value shown in abscissa.

audio recordings. These analyses can corroborate several musicological inferences, and can provide additional insights into the differences between musicology, music theory and music practice.

The rhythm patterns are computed using a spectral flux feature (called LogFilt- SpecFlux as proposed by Böck, Krebs, and Schedl (2012) and used further by citeAkrebs:13:bpm) that is used for detecting musical onsets in audio recordings. The short time Fourier transform (STFT) of the audio signal with a window size of 46.4 ms (2047 samples of audio at a sampling rate of 44.1 kHz), FFT size of 2048 and hop size of 20 ms is computed from audio. The

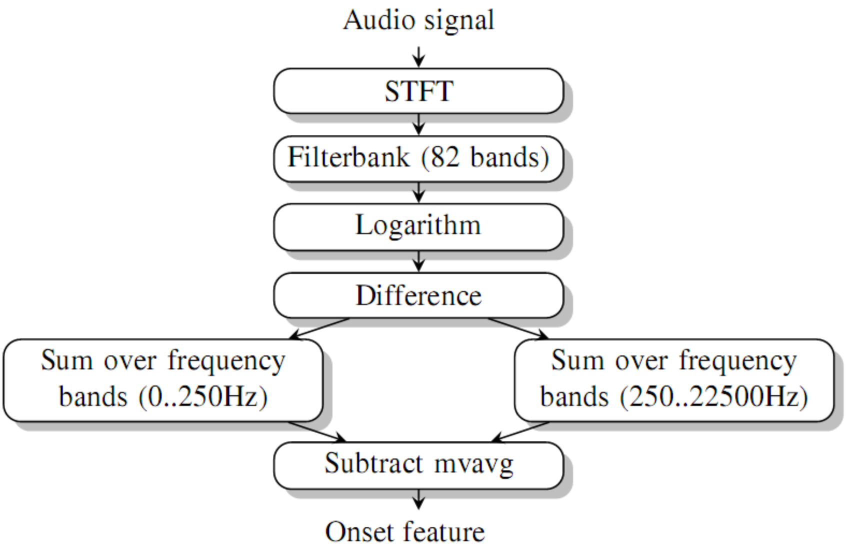


Figure 4.7: Computation of the spectral flux feature in two frequency bands. **Figure to be updated**

successive difference between frames of the logarithm of the filterbank energies in 82 different bands are then computed. Since the bass onsets have significant information about the rhythmic patterns, the features are computed in two frequency bands (Low: $\leq$ 250 Hz, High: $>$ 250 Hz) to additionally consider the bass onsets. The process of computing the spectral flux feature is outlined in Figure 4.7.

Using beat and downbeat annotated training data, the audio features from all music pieces in a specific tāḷa are then grouped into cycle length sequences, and interpolated to equal lengths using a fine grid. A mean of all the pattern instances for a specific tāḷa is computed in both the frequency bands and used as a representative rhythmic pattern illustrated here. Improve: The patterns played in a tāḷa cycle have both “energy accents and timbral characteristics”. The rhythm patterns have been generated using the spectral flux feature and Hence can only explain energy accents with these figures. List down other limitations of this approach. The patterns are indicative of the surface rhythm present in thesis audio recordings.

The Figures 4.8-4.15 show the ensemble average of cycle length patterns over all the pieces in the dataset for each tāḷa, computed using the spectral flux feature in two different frequency bands as outline above. The patterns in each figure pane is normalized so

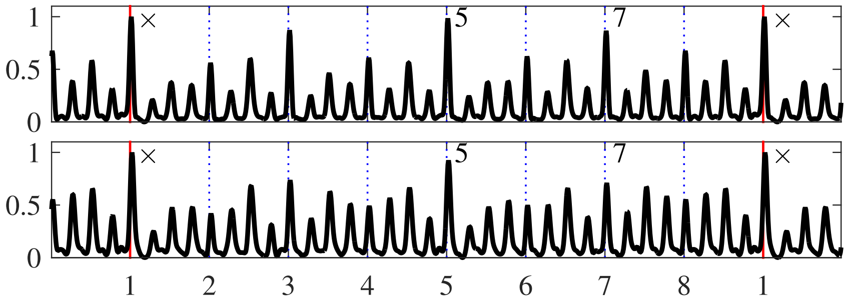


Figure 4.8: Cycle length rhythmic patterns learned from CMR$_f$ dataset for ādi tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

that maximum value is 1, to comment on relative onset strengths at different metrical positions of the cycle. The bottom pane corresponds to the low frequency band (y_l) and the top pane corresponds to the high frequency band (y_h). **Explain how those patterns were obtained with a block diagram: CMD-adi-all-hi250-superflux-mvavg-normZ,CMD-adi-all-lo230-superflux-mvavg-normZ.**

The rhythm patterns are indicative of mridangam strokes played in the cycle. In the figures, the bottom pane that shows the low frequency band has content from the left bass drum while the top pane has content predominantly from the right pitched drum, but additionally from the lead melody. Hence, for the purpose of this discussion, we use the terms left and right accents to refer to the accents in rhythm patterns from the bottom and top pane, respectively. The left and right accents provide interesting insights into the patterns played within a tāḷa cycle. In addition, these rhythm patterns help in meter tracking.

We list down and discuss some salient qualitative observations from figures for each tāḷa, for both CMR$_f$ dataset and its subset CMR. The Figures 4.8-4.15 show the cycle length rhythm patterns for all tāḷas for both CMR$_f$ and CMR datasets. For each tāḷa, we plot the rhythm patterns together to compare patterns across the short

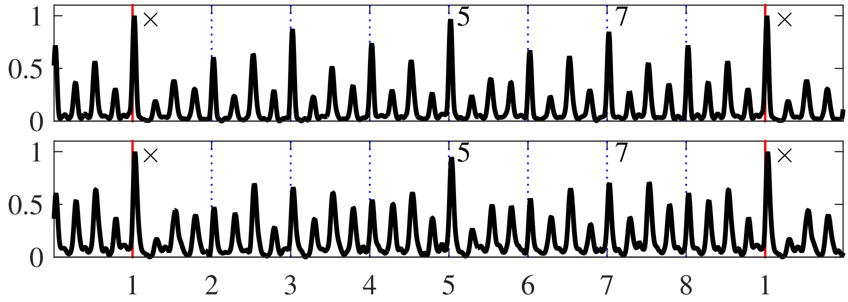


Figure 4.9: Cycle length rhythmic patterns learned from CMR dataset for ādi tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

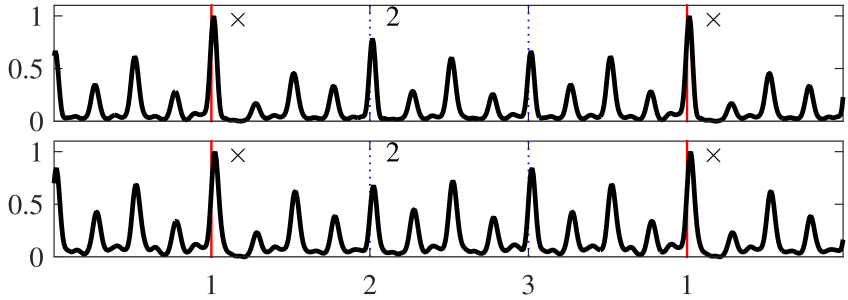


Figure 4.10: Cycle length rhythmic patterns learned from $\mathtt{CMR}_\mathrm{f}$ dataset for rūpaka tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

excerpts in CMR dataset and full length pieces in $\mathtt{CMR}_\mathrm{f}$ dataset.

Overall, we see stronger accents on the akṣaras, with sama having the strongest accent in most cases. We can clearly see the ac-

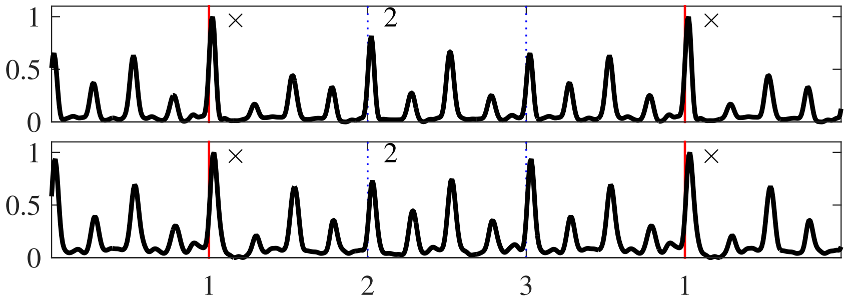


Figure 4.11: Cycle length rhythmic patterns learned from CMR dataset for rūpaka tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

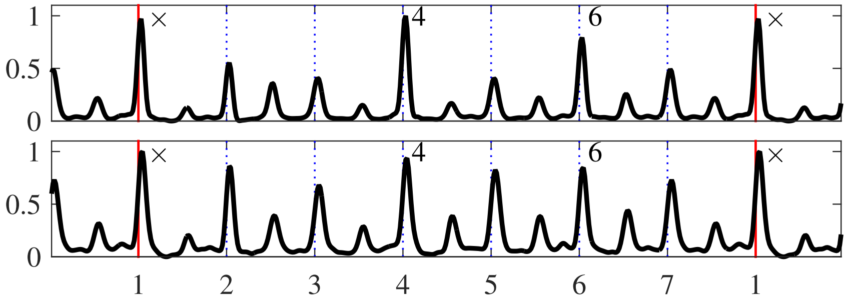


Figure 4.12: Cycle length rhythmic patterns learned from $\mathtt{CMR}_\mathtt{f}$ dataset for miśra chāpu tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

cents organized in three different strengths, reflecting the metrical levels of the a□ga, the beat and the akṣara. The two akṣara long beats in the tāḷas miśra chāpu and khaṇḍa chāpu, and the four akṣara

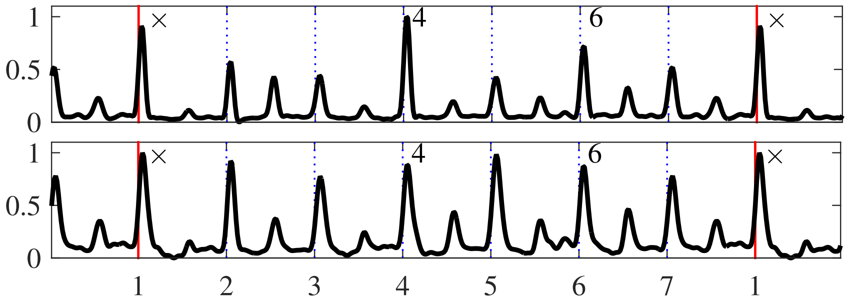


Figure 4.13: Cycle length rhythmic patterns learned from CMR dataset for miśra chāpu tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

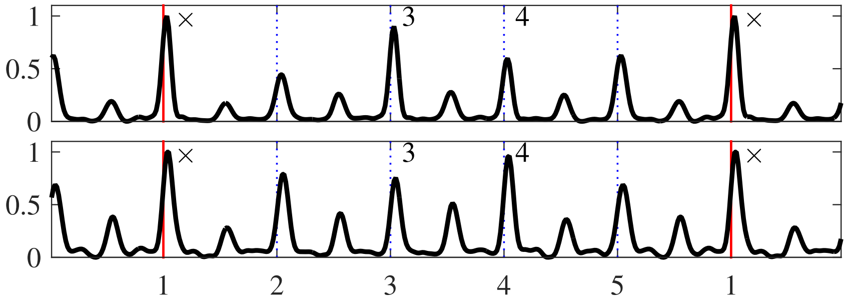


Figure 4.14: Cycle length rhythmic patterns learned from $\mathtt{CMR}_\mathtt{f}$ dataset for khaṇḍa chāpu tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

long beats in tāḷa ādi and rūpaka can be additionally seen. The patterns and ṭhēkās played in Carnatic music are quite diverse, and no obvious "tāḷa pattern" can be inferred, apart from the three levels

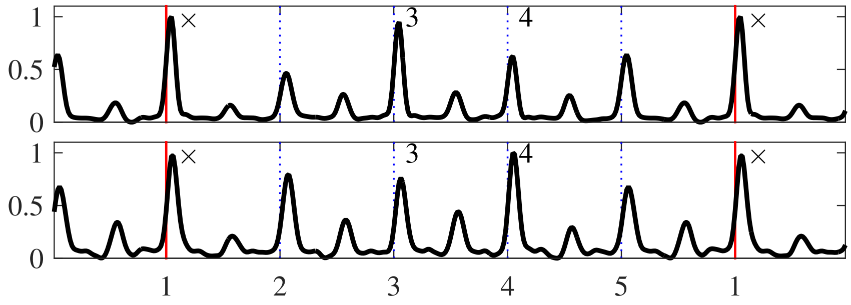


Figure 4.15: Cycle length rhythmic patterns learned from CMR dataset for khaṇḍa chāpu tāḷa, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the beat number within the cycle (dotted lines), with 1 indicating the sama (marked with a red line). The start of each a□ga is indicated with beat numbers at the top of each pane (sama shown as ×).

of accents. This shows that the tāḷas are metrical structures that allow many different patterns to be played, and not a specific rhythm. It is further seen that the first akṣara after sama has softer accents. Fewer strokes are played after the sama, to emphasize that the sama has just passed and a new cycle has begun. It might also perhaps indicate some form of recovery time after the sense stroke-playing towards the end of the cycle. Further, the rhythm patterns computed using CMR dataset are very similar to those computed using CMR$_\text{f}$ dataset, showing that CMR is a good representative subset of the larger CMR$_\text{f}$. Additionally, all the observations we make with patterns from CMR$_\text{f}$ extend to CMR. We now discuss several tāḷa specific observations.

The Figures 4.8-4.9 show the rhythm patterns for glsadi tāḷa. We see that a three level hierarchy of a□ga, beats and akṣaras is well demarcated. The akṣara at half cycle (beat 5) has an accent as strong as the sama. The odd beats (marked 1, 3, 5, 7) have stronger right accents. The left accents are distributed through the cycle, with strong accents at half cycle.

The Figures 4.10-4.11 show the rhythm patterns for glsrupaka tāḷa. Apart from the three level hierararchy of accents that is quite apparent, the half beat accent between the beats 2 and 3 are strong -

indicating the often played 6+6 akṣaragrouping structure of rūpaka, with a ternary meter.

The Figures 4.12-4.13 show the rhythm patterns for glsmishra chapu tāḷa. We see that the a□ga boundaries have strong left and right accents showing their use as anchor points to indicate the progression through the cycle. Though defined with a 3+2+2 akṣara grouping structure, a 1+2+2+2 structure is often seen in miśra chāpu tāḷa, which can be observed here, based on the strong left accent on beat 2. A additional strong left accent on beat 5 shows that it is also used as an anchor.

From the rhythm patterns of khaṇḍa chāpu tāḷa shown in Figures 4.14-4.15 show a strong left accent on beat 4, which is used an anchor. A stronger right accent on beat 3 shows the progression through the unequal a□gas. The 2+1+2 akṣara grouping structure of khaṇḍa chāpu is often played out as 3+2 or 2+3, showing strong accents on beats 3 and 4.

Applications of the dataset

The CMR$_{f}$ dataset and its subset CMR dataset are intended to be test corpora for several computational rhythm analysis tasks in Carnatic music. Possible tasks include tāḷa, sama and beat tracking, tempo estimation and tracking, tāḷa recognition, rhythm based segmentation of musical audio, structural segmentation, audio to score/lyrics alignment, and rhythmic pattern analysis. In this thesis, these two datasets are primarily used for rhythmic pattern analysis and meter inference/tracking. Most of the research results are presented for CMR and then extended to CMR$_{f}$ to verify their applicability to larger datasets.

4.2.2 Hindustani music rhythm dataset

CompMusic Hindustani music rhythm dataset (HMR$_{f}$)[23]is a rhythm annotated test corpus for automatic rhythm analysis tasks in Hindustani Music. The collection consists of audio excerpts from the CompMusic Hindustani research corpus, manually annotated time aligned markers indicating the progression through the tāl cycle, and the associated tāl related metadata. The dataset has pieces from

[23] http://compmusic.upf.edu/hindustani-rhythm-dataset

Tāl	# Pieces	Total Duration hours (min)	# Ann.	# Sam
Tīntāl	54	1.80 (108)	17142	1081
Ēktāl	58	1.93 (116)	12999	1087
Jhaptāl	19	0.63 (38)	3029	302
Rūpak tāl	20	0.67 (40)	2841	406
Total	151	5.03 (302)	36011	2876

Table 4.9: $\mathtt{HMR_f}$ dataset showing the total duration and number of annotations. #Sam shows the number of sam annotations and #Ann. shows the number of mātrā annotations (including sams).

Tāl	$\overline{\tau_s} \pm \sigma_s$	$\overline{\tau_o} \pm \sigma_o$	$[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$
Tīntāl	10.36 ± 9.875	0.65 ± 0.617	[2.32, 44.14]
Ēktāl	30.20 ± 26.258	2.52 ± 2.188	[2.23, 69.73]
Jhaptāl	8.51 ± 3.149	0.85 ± 0.315	[4.06, 16.23]
Rūpak tāl	7.11 ± 3.360	1.02 ± 0.480	[2.82, 16.09]

Table 4.10: Tāl cycle length indicators for $\mathtt{HMR_f}$ dataset. $\overline{\tau_s}$ and σ_s indicate the mean and standard deviation of the median inter-sam interval of the pieces, respectively. $\overline{\tau_o}$ and σ_o indicate the mean and standard deviation of the median inter-mātrā interval of the pieces, respectively. $[\tau_{s,\text{min}}\ ,\ \tau_{s,\text{max}}]$ indicate the minimum and maximum value of τ_s and hence the range of τ_s in the dataset. All values in the table are in seconds.

four popular tāls of Hindustani music (Table 4.9), which encompasses a majority of Hindustani khyāl music.

The audio recordings are chosen from the CompMusic Hindustani music collection. The pieces include a mix of vocal and instrumental recordings, new and old recordings, and to span three layas. For each taal, there are pieces in drt (fast), madhya (medium) and vila□bit layas. All pieces have Tabla as the percussion accompaniment. All the audio recordings in the dataset are 2 min excerpts of full length pieces. Each piece is uniquely identified using the MBID of the recording. The pieces are stereo, 160 kbps, mp3 files

Tāl	# Pieces	Total Duration hours (min)	# Ann.	# Sam
Tīntāl	13	0.43 (26)	1020	65
Ēktāl	32	1.07 (64)	967	79
Jhaptāl	6	0.2 (12)	592	59
Rūpak tāl	8	0.27 (16)	701	101
Total	59	1.97 (118)	3280	304

Table 4.11: HMR$_l$ dataset showing the total duration and number of annotations. #Sam shows the number of sam annotations and #Ann. shows the number of mātrā annotations (including sams).

Tāl	$\overline{\tau_s} \pm \sigma_s$	$\overline{\tau_o} \pm \sigma_o$	$[\tau_{s,\text{min}}, \tau_{s,\text{max}}]$
Tīntāl	26.16 ± 7.963	1.63 ± 0.498	[18.57, 44.14]
Ēktāl	52.16 ± 12.531	4.35 ± 1.044	[14.43, 69.73]
Jhaptāl	12.30 ± 1.935	1.23 ± 0.194	[10.20, 16.23]
Rūpak tāl	10.28 ± 3.050	1.47 ± 0.436	[6.95, 16.09]

Table 4.12: Tāl cycle length indicators for HMR$_l$ dataset. $\overline{\tau_s}$ and σ_s indicate the mean and standard deviation of the median inter-sam interval of the pieces, respectively. $\overline{\tau_o}$ and σ_o indicate the mean and standard deviation of the median inter-mātrā interval of the pieces, respectively. $[\tau_{s,\text{min}}, \tau_{s,\text{max}}]$ indicate the minimum and maximum value of τ_s and hence the range of τ_s in the dataset. All values in the table are in seconds.

sampled at 44.1 kHz. The audio is also available as downmixed mono WAV files for experiments.

There are several annotations that accompany each audio file in the dataset. The primary annotations are audio synchronized time-stamps indicating the different metrical positions in the tāl cycle. The sam and mātrās of the cycle are annotated. The annotations were created using Sonic Visualizer by tapping to music and manually correcting the taps. Each annotation has a time-stamp and an associated numeric label that indicates the mātrā position in the tāl cycle, as shown in Figure 2.3. The sams are indicated using the numeral 1. The time varying tempo of the piece can be obtained

Tāl	# Pieces	Total Duration hours (min)	# Ann.	# Sam
Tīntāl	41	1.37 (82)	16122	1016
Ēktāl	26	0.87 (52)	12032	1008
Jhaptāl	13	0.43 (26)	2437	243
Rūpak tāl	12	0.40 (24)	2140	305
Total	92	3.07 (184)	32731	2572

Table 4.13: HMR$_s$ dataset showing the total duration and number of annotations. #Sam shows the number of sam annotations and #Ann. shows the number of mātrā annotations (including sams).

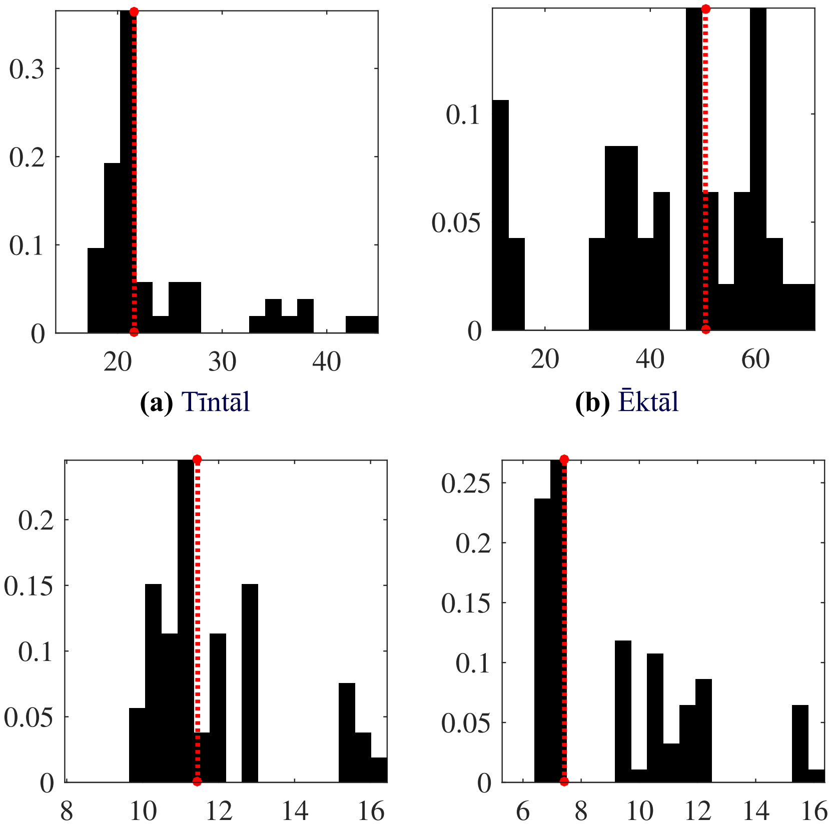
Tāl	$\overline{\tau_s} \pm \sigma_s$	$\overline{\tau_o} \pm \sigma_o$	$[\tau_{s,\text{min}}, \tau_{s,\text{max}}]$
Tīntāl	5.35 ± 1.823	0.33 ± 0.114	[2.32, 9.89]
Ēktāl	3.17 ± 0.471	0.26 ± 0.039	[2.23, 4.11]
Jhaptāl	6.77 ± 1.688	0.68 ± 0.169	[4.06, 9.97]
Rūpak tāl	5.00 ± 1.191	0.71 ± 0.170	[2.82, 6.68]

Table 4.14: Tāl cycle length indicators for HMR$_s$ dataset. $\overline{\tau_s}$ and σ_s indicate the mean and standard deviation of the median inter-sam interval of the pieces, respectively. $\overline{\tau_o}$ and σ_o indicate the mean and standard deviation of the median inter-mātrā interval of the pieces, respectively. $[\tau_{s,\text{min}}, \tau_{s,\text{max}}]$ indicate the minimum and maximum value of τ_s and hence the range of τ_s in the dataset. All values in the table are in seconds.

from the mātrā and sam annotations.

For each excerpt, the tāl and the lay of the piece are recorded. Each excerpt can be uniquely identified and located with the MBID of the recording, and the relative start and end times of the excerpt within the whole recording. The artist, release, the lead instrument, and the rāg of the piece are additional editorial metadata obtained from the release. There are optional comments on audio quality and annotation specifics. The annotations and the associated metadata have been verified for correctness and completeness by a professional Hindustani musician and musicologist.

The HMR$_f$ dataset is described in Table 4.9, showing the four

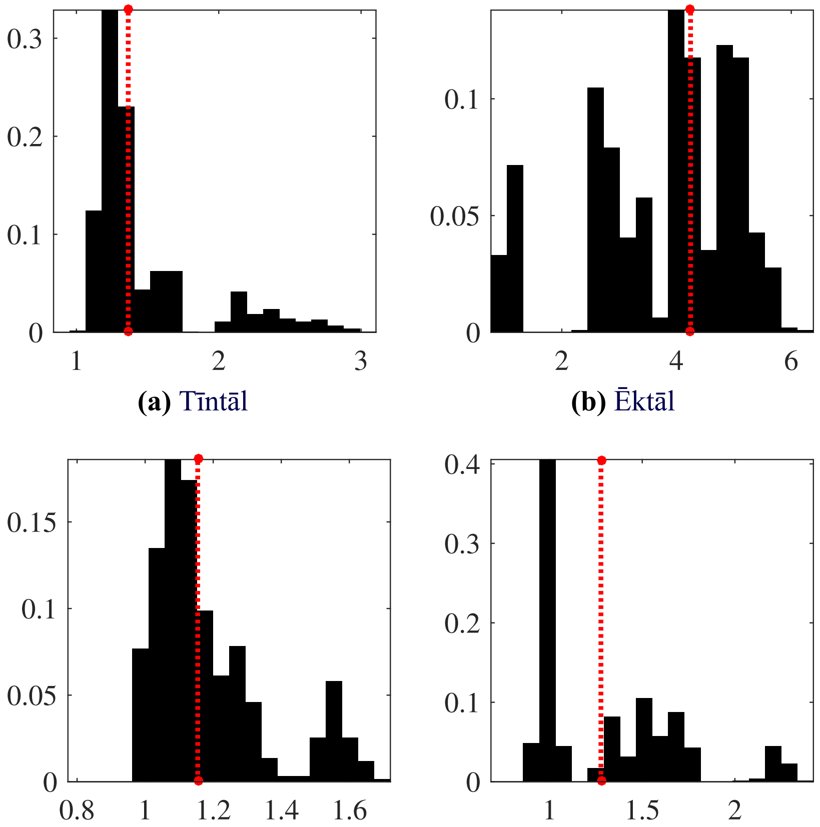


(a) Tīntāl **(b)** Ēktāl

(c) Jhaptāl **(d)** Rūpak tāl

Figure 4.16: A histogram of the inter-sam interval τ_s in the $\texttt{HMR}_l$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the τ_o value shown in abscissa. The median τ_s for each tāl is shown as a red dotted line.

tāls and the number of pieces for each tāl, totaling to 151 pieces. The total duration of audio in the dataset is about 5 hours, with 36011 time-aligned mātrā annotations of which 2876 are sam annotations. Table 4.10 shows a basic statistical analysis of the tāl cycle length indicators in the dataset to understand the tempo characteristics and the range of the metrical cycle lengths in the dataset. The large range of tempi seen in Hindustani music is reflected in the dataset, with the values of median inter-sam interval $\overline{\tau_s}$, ēktāl cycle lengths ranging from 2.2 seconds to 69.7 seconds, which is about 5 tempo octaves. This also shows that the mātrā period can



(a) Tīntāl

(b) Ēktāl

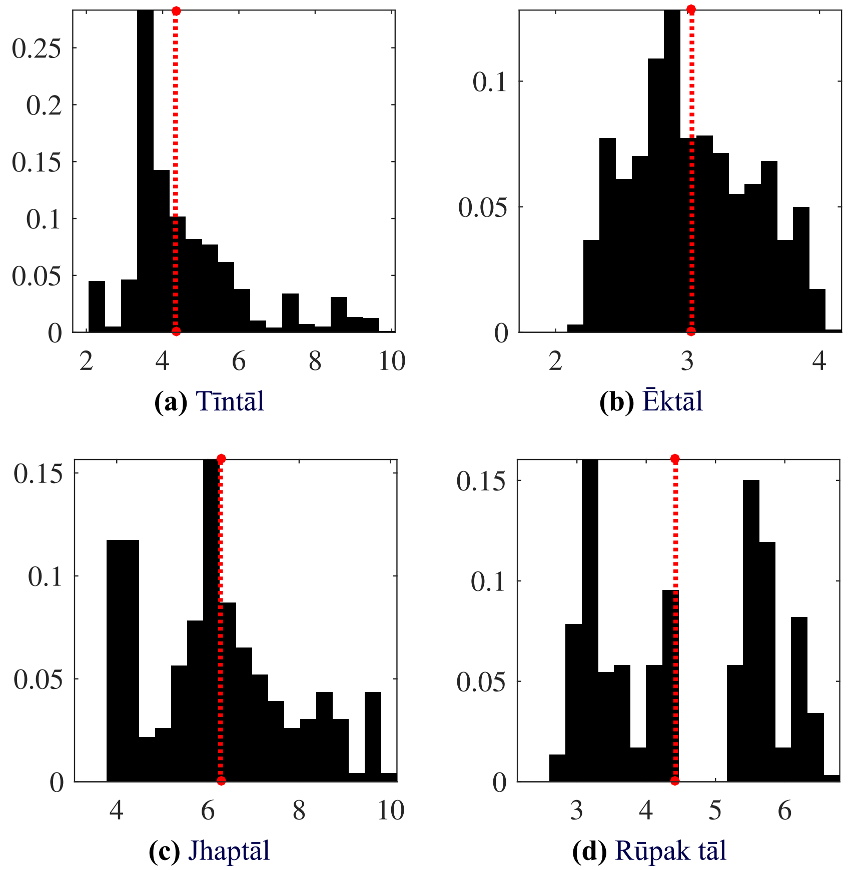
(c) Jhaptāl

(d) Rūpak tāl

Figure 4.17: A histogram of the inter-mātrā interval τ_o in the `HMR`$_1$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the τ_o value shown in abscissa. The median τ_o for each tāl is shown as a red dotted line.

vary from less than 150 ms to over 6 seconds. This huge range of cycle lengths and mātrā periods is a significant challenge in Hindsutani music automatic meter inference. Across different tāls, we see that tīntāl and ēktāl have the largest range of $\overline{\tau_s}$, since they are performed in all the lay classes, vila□bit to dr□t. Jhaptāl and rūpak tāl have smaller $\overline{\tau_s}$ ranges.

The dataset consists of excerpts with a wide tempo range from 10 MPM (matras per minute) to 370 MPM. The lay of a piece has a significant effect on meter tracking and rhythm analysis due to this wide range of possible tempo. To study any effects of the tempo



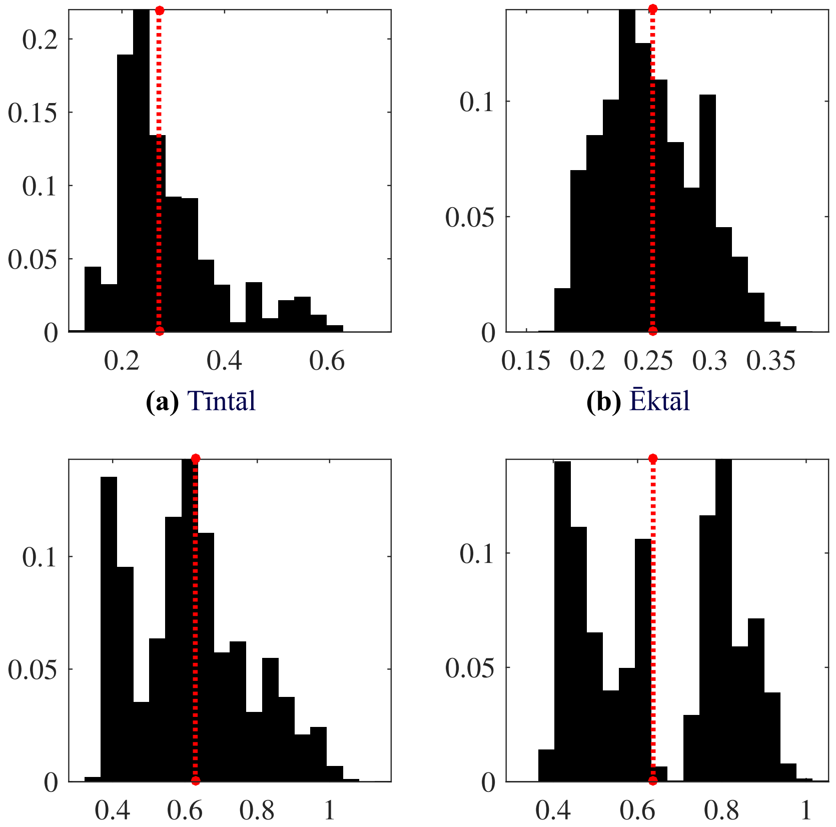
(a) Tīntāl **(b)** Ēktāl

(c) Jhaptāl **(d)** Rūpak tāl

Figure 4.18: A histogram of the inter-sam interval τ_s in the HMR$_\text{s}$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the τ_o value shown in abscissa. The median τ_s for each tāl is shown as a red dotted line.

class, the full HMR$_\text{f}$ dataset is divided into two other subsets - the long cycle subset called the HMR$_\text{l}$ dataset (shown in Table 4.11) consisting of vila□bit pieces with a median tempo between 10-60 MPM, and the short cycle subset HMR$_\text{s}$ dataset (shown in Table 4.13) with madhya lay (60-150 MPM) and the drtlay (150+ MPM) pieces.

Hindustani music rhythm dataset (subset with vila□bit and madhya lay pieces) (HMR$_\text{l}$) dataset shown in Table 4.11 consists of 59 pieces in vila□bit lay, with over 3200 mātrā and sam annotations. A majority of pieces are in ēktāl and tīntāl. Since its very uncom-



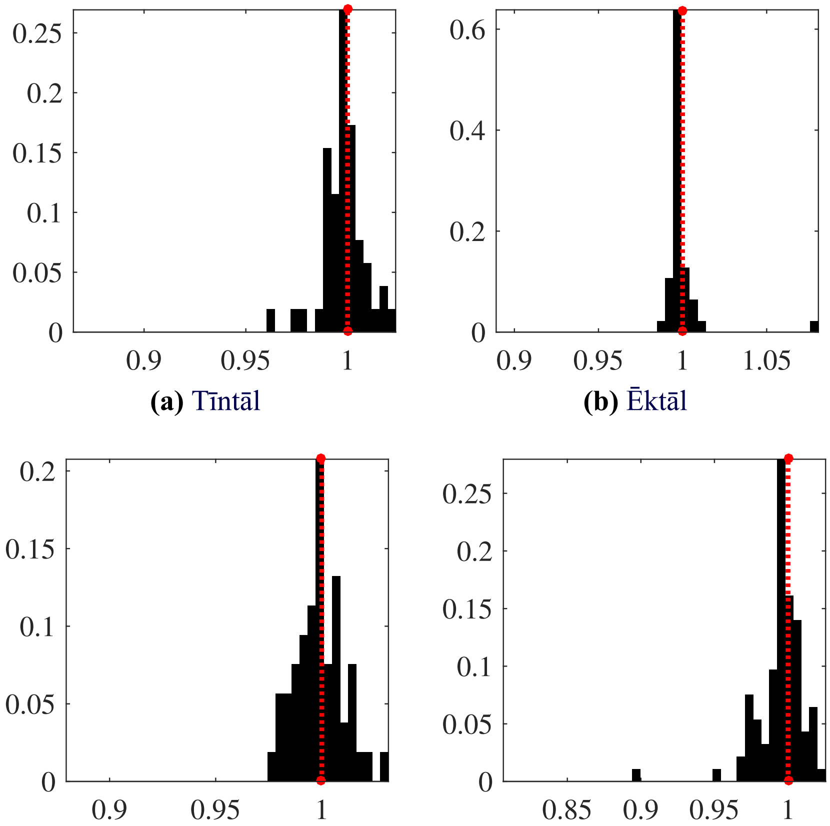
(a) Tīntāl **(b)** Ēktāl

(c) Jhaptāl **(d)** Rūpak tāl

Figure 4.19: A histogram of the inter-mātrā interval τ_o in the HMR$_\text{s}$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the τ_o value shown in abscissa. The median τ_o for each tāl is shown as a red dotted line.

mon for a piece to be performed in vila□bit lay jhaptāl and rūpak tāl, there are only 6 and 8 pieces for those tāls, respectively. As described with HMR$_\text{f}$, a basic statistical analysis of the tāl cycle length indicators in Table 4.12 shows that the median inter-sam interval and its range for jhaptāl and rūpak tāl are less than that for tīntāl and ēktāl.

Hindustani music rhythm dataset (subset with dr□t lay pieces) (HMR$_\text{s}$) dataset consists on 92 pieces in madhya and dr□t lay, with over 3 hours of audio and over 32000 mātrā and sam annotations. A basic statistical analysis of the tāl cycle length indicators in Ta-



(a) Tīntāl

(b) Ēktāl

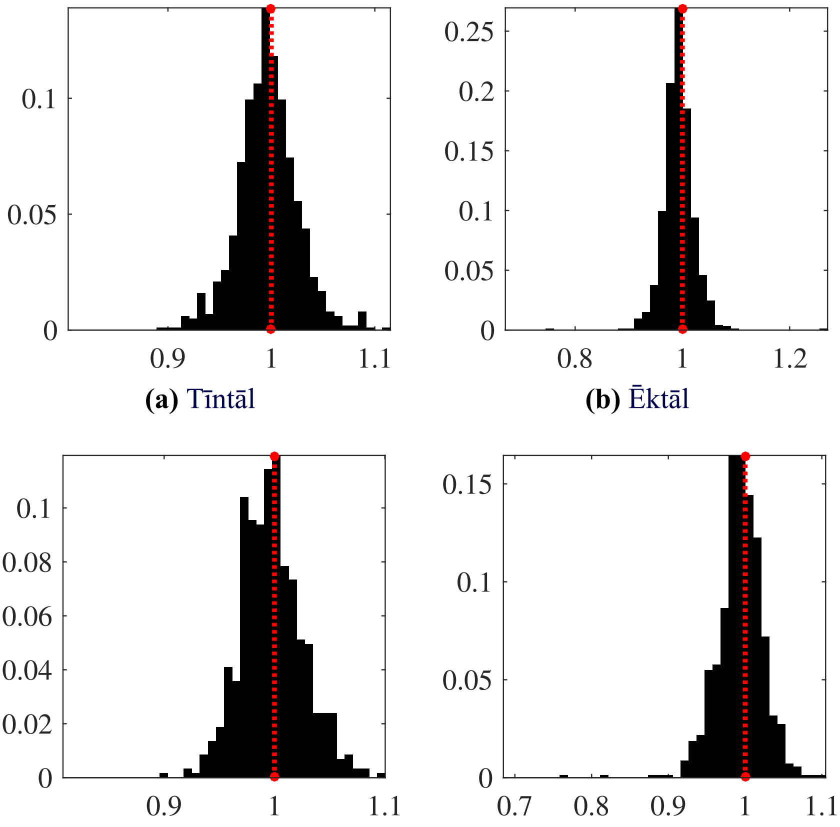
(c) Jhaptāl

(d) Rūpak tāl

Figure 4.20: A histogram of the median normalized inter-sam interval τ_s in the $\texttt{HMR}_1$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the normalized τ_s value shown in abscissa.

ble 4.14 shows that the pieces of tīntāl and ēktāl have higher tempi in the dataset. Comparing the median mātrā period for ēktāl between Table 4.12 (4.35 second) and Table 4.14 (0.26 second) shows that ēktāl is performed either in vila□bit or drt and its rare for a piece to be performed in madhya lay ēktāl.

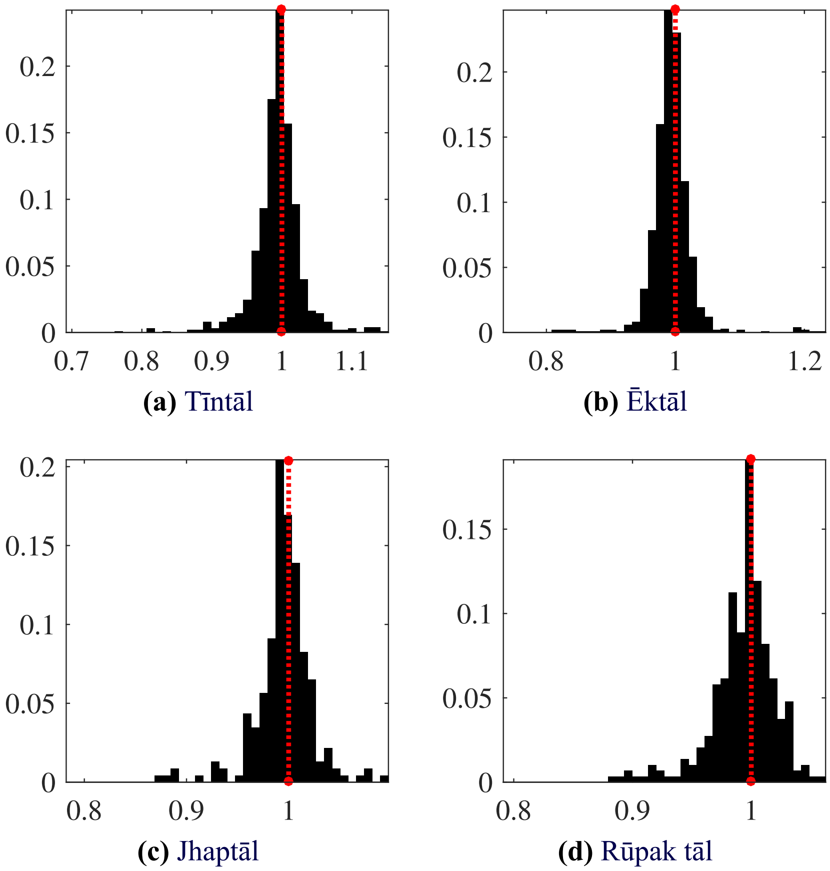
The pieces is Hindustani music have a tempo class indicated but not a specific tempo value, nor are they performed to a metronome. Hence the tempo varies over a piece in time - often the tempo increases with time. Hence, in addition to the median values tabulated in Table 4.6 we present further analysis of the inter-sam inter-



(a) Tīntāl **(b)** Ēktāl **(c)** Jhaptāl **(d)** Rūpak tāl

Figure 4.21: A histogram of the median normalized inter-mātrā interval τ_o in the $\mathtt{HMR_l}$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the normalized τ_o value shown in abscissa.

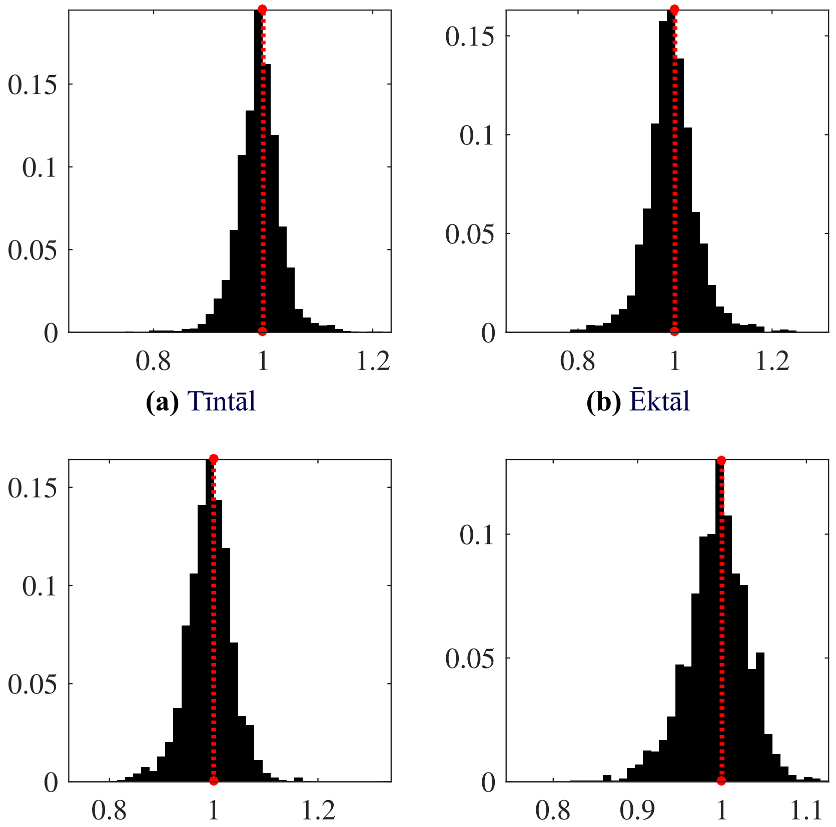
val (τ_s) and inter-mātrā interval (τ_o) for each tāl. For better comparison, we present this analysis for each data subset $\mathtt{HMR_l}$ and $\mathtt{HMR_s}$ separately. A histogram of τ_s and τ_o for each tāl for $\mathtt{HMR_l}$ dataset is shown in Figure 4.16 and Figure 4.19, respectively, and those for $\mathtt{HMR_s}$ dataset is shown in Figure 4.18 and Figure 4.19, respectively. These figures show the distribution of cycle lengths in the dataset over the whole range of τ_s for each tāl, around the median value. The large range of τ_s and τ_o values and an irregular distribution spanning the whole range is seen with both datasets, unlike the Carnatic music $\mathtt{CMR_f}$ dataset with a short tightly defined range of tempo.



(a) Tīntāl **(b)** Ēktāl **(c)** Jhaptāl **(d)** Rūpak tāl

Figure 4.22: A histogram of the median normalized inter-sam interval τ_s in the $\mathtt{HMR_s}$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the normalized τ_s value shown in abscissa.

In addition, similar to what was presented for Carnatic music, to illustrate and measure the time varying tempo of music pieces in Hindustani music, we normalize all the τ_s and τ_o values in a piece by the median in the piece to obtain median normalized τ_s and τ_o values, a histogram of which is shown in Figure 4.20 and Figure 4.21, respectively for $\mathtt{HMR_l}$ dataset and Figure 4.22 and Figure 4.23, respectively for $\mathtt{HMR_s}$ dataset. These histograms are centered around 1 and normalized by the median. From the figures, it is clear that the tempo is time varying but with less than about 10% maximum deviation from the median tempo of the piece for all tāls. This is in contrast to Carnatic music where the median



(a) Tīntāl **(b)** Ēktāl

(c) Jhaptāl **(d)** Rūpak tāl

Figure 4.23: A histogram of the median normalized inter-mātrā interval τ_o in the HMR$_s$ dataset for each tāl. The ordinate is the fraction of the total count corresponding to the normalized τ_o value shown in abscissa.

normalized tempo had a higher deviation ($\sim$ 20 %).

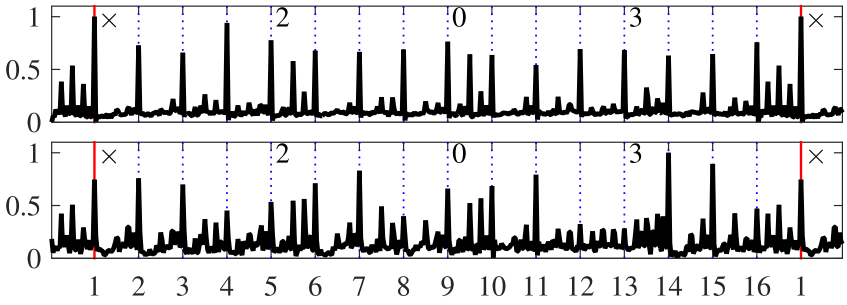


Figure 4.24: Cycle length rhythmic patterns learned from HMR$_l$ dataset for tīntāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

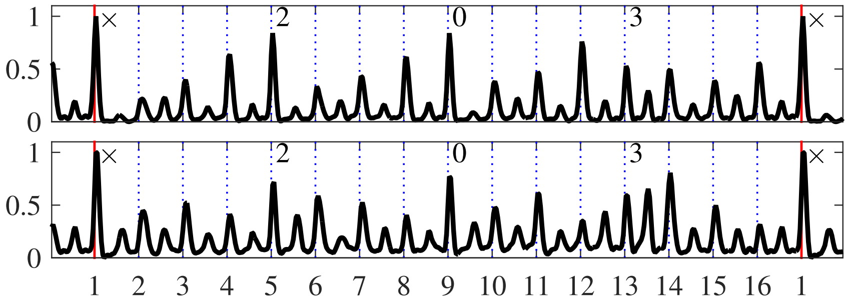


Figure 4.25: Cycle length rhythmic patterns learned from HMR$_s$ dataset for tīntāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

Rhythm patterns in Hindustani rhythm datasets

Similar to Carantic music, we do corpora level analysis of rhythm patterns in Hindustani music and draw several musicological in-

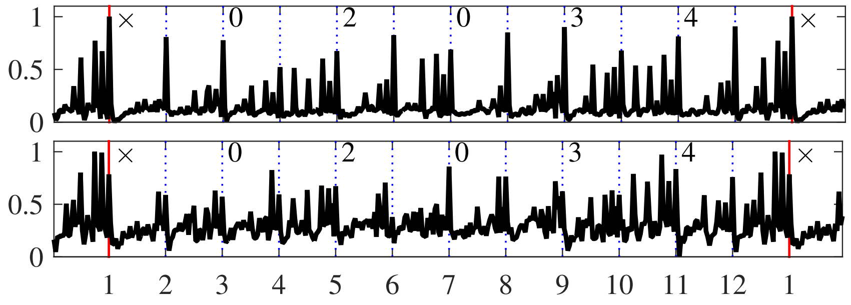


Figure 4.26: Cycle length rhythmic patterns learned from HMR$_l$ dataset for ēktāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

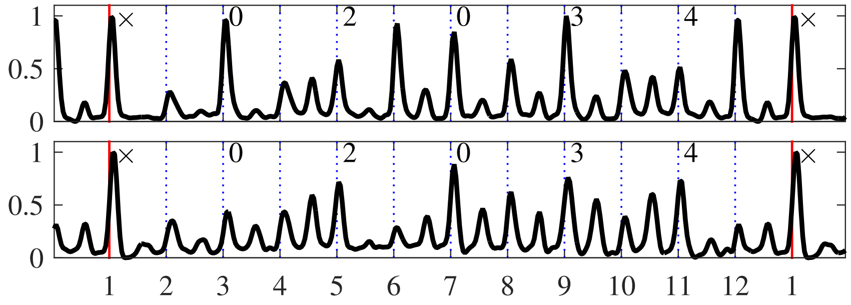


Figure 4.27: Cycle length rhythmic patterns learned from HMR$_s$ dataset for ēktāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

ferences and insights, contrasting the differences between music theory and practice. The rhythm patterns described in this section were obtained using spectral flux, identical to the process described for Carnatic music.

The Figures 4.24-4.31 show the cycle length rhythm patterns

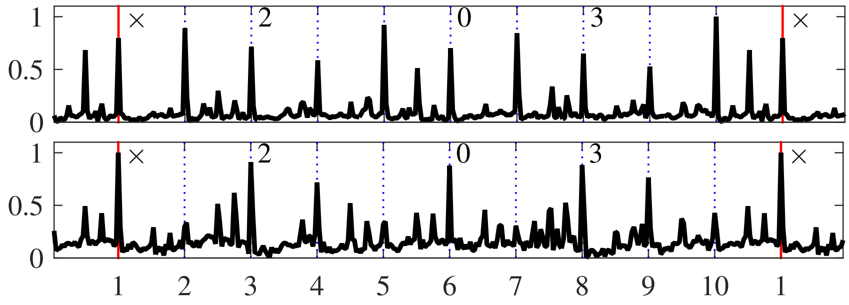


Figure 4.28: Cycle length rhythmic patterns learned from HMR$_l$ dataset for jhaptāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

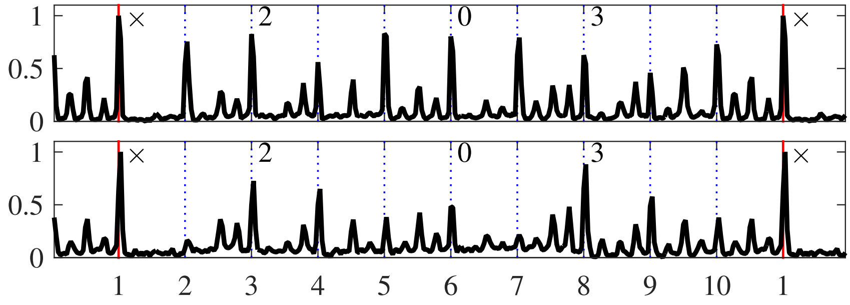


Figure 4.29: Cycle length rhythmic patterns learned from HMR$_s$ dataset for jhaptāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

for all tāls for both HMR$_l$ and HMR$_s$ datasets, using the spectral flux feature computed identically to the way it was computed for Carnatic music rhythm patterns. The rhythm patterns in Hindustani are indicative of tabla strokes played in the cycle. In the figures, the bottom pane that shows the low frequency band has content from

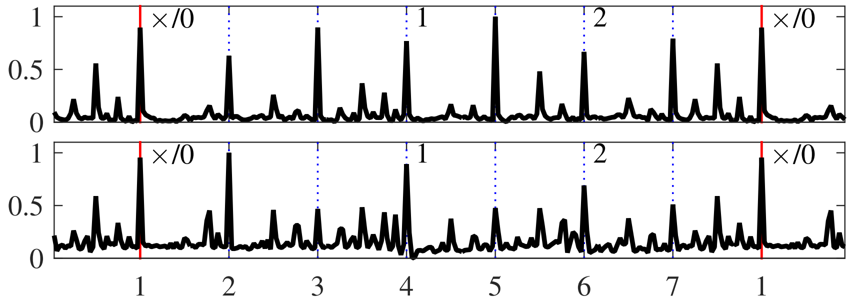


Figure 4.30: Cycle length rhythmic patterns learned from HMR$_l$ dataset for rūpak tāl tāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

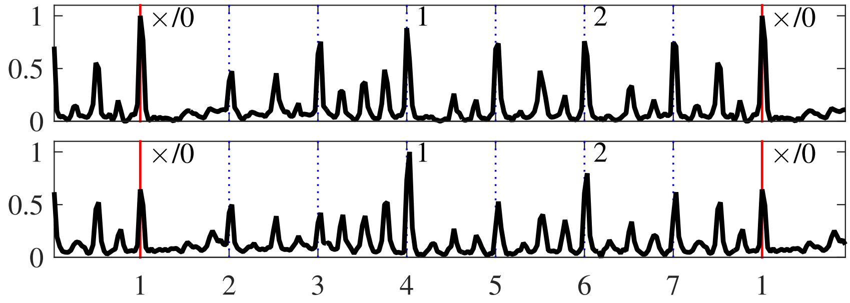


Figure 4.31: Cycle length rhythmic patterns learned from HMR$_s$ dataset for rūpak tāl tāl, computed from spectral flux feature and averaged over all the pieces in the dataset. The bottom/top pane corresponds to the low/high frequency bands, respectively. The abscissa is the mātrā number within the cycle (dotted lines), with 1 indicating the sam (marked with a red line). The start of each vibhāg is indicated at the top of each pane (sam shown as ×).

the bāyān (the left bass drum) of the tabla while the top pane has content predominantly from the dāyān (the right pitched drum) of the tabla, but additionally from the lead melody. Hence, for the purpose of this discussion, we use the terms left and right accents to refer to the accents in rhythm patterns from the bottom and top pane, respectively. The left and right accents provide interesting insights into the patterns played within a tāl cycle. We additionally compare rhythm patterns across the layas by plotting the patterns for $\mathtt{HMR}_l$ dataset (with vila□bit lay pieces) and $\mathtt{HMR}_s$ dataset (madhya and dr□t lay pieces) - for each tāl, the patterns for these two data subsets are plotted in two figures one below the other.

The patterns played in a tāl cycle have both energy accents and timbral characteristics. The rhythm patterns have been generated using the spectral flux feature and Hence can only explain energy accents with these figures. We list down and discuss some salient qualitative observations from the figures for each tāḷa, for both vila□bit lay and dr□t lay. The patterns are indicative of the surface rhythm present in these audio recordings.

Overall, from Figures 4.24-4.31 , we observe across all tāls and layas that accents are stronger on the mātrās, with accents present even at half and fourth divisions of the matra in many cases. The sam most often has the strongest accent. Unlike Carnatic tāḷas, ṭhēkās in Hindustani music are less flaxible, and hence we can infer several concrete conclusions from the rhythm patterns of Hindustani music.

Across all tāls in vila□bit and madhya lay (the figures in the top of the page), we see additional filler strokes present between mātrās. These fillers are also mostly mostly towards to second half of the mātrā. The 1st mātrā (and often the 2nd mātrā) is quite empty with few accents, while the last few mātrās of the cycle have dense accents. This is place a special emphasis on the sam, indicating the approaching of sam with fillers and dense stroke playing and a short recovery period with fewer strokes after the sam has passed. In addition, a dense matra with many fillers is often followed by a sparsely accented mātrā to better contrast the progression through the tāl cycle. Due to the large mātrā period (τ_o) in vila□bit and madhya lay, each mātrā acts as an anchor for timekeeping, and can be played without any effect from the previous strokes (in fast tabla playing in dr□t, the previous stroke can possibly affect the sound, intonation, and playing technique of the following strokes). In addition, due to a large time interval available to play the ṭhēkā, the tabla playing musician focuses on modulation of left bass strokes that can sustain longer. Finally, left and right hand can operate independently, which means modulation of accents through the cycle can be different for left and right accents. The left and right strokes also complement each other. Each of these effects can be observed in the patterns of vila□bit and madhya lay.

In contrast, across all tāls in drt lay (the figures at the bottom of the page), given the short cycles, we see that vibhāgs are anchors. The fillers are largely restricted only to half mātrā, with lower accents. Drt pieces also has a relatively more relaxed timing, and the focus is on right strokes, with the left hand playing the theory defined “textbook” strokes for timekeeping. In addition, the left and right hands are in sync, which can be seen in the modulation of accents through the cycle being well correlated for both left and right accents - the left and right strokes work together here, in contrast to complementing each other as in vila□bit lay.

We now present some tāl specific observations from the rhythm patterns for each tāl. Some of these observation corroborate theory while some of them show the contrast between theory and practice. These inferences mainly address tabla stroke playing during the cycles, while the effects of melody has not been considered into account. This is a valid assumption to make since these patterns are averaged over several cycles, averaging out and reducing the effect of melody on these rhythm patterns.

Vila□bit and madhya lay tīntāl: From Figure 4.24, we see that the 14th matra has the strongest left accent, and the last mātrā (matra 16) has many fillers, both to indicate the arrival of sam - a phenomenon known in music theory as āmad (literal meaning - the approach). A strong left accent on the 9th matra is not defined in theory (the stroke in the ṭhēkā is a right stroke NA), but often a DHA is played instead. This is a difference between theory and practice that is known to musicians and can be observed in the pattern here too. As described earlier, the right stroke fillers are fewer in mātrās 1 and 2, and the left supports the timekeeping task when the right accents are weaker there. 4th mātrā has a strong right accent, to indicate the end of the 1st vibhāg, after a filler-less mātrās 2 and 3. The beginning of the 2nd and 3rd vibhāgs, labeled 2 and 0 have larger number of fillers. The left accents between the 11th and the 14th matra are weak - with the 11th and 14th mātrā accents acting as anchors for the “quiet” created in between them. It is interesting to note the varying modulation of accent levels through the vibhāgs of the cycle. Specifically, we can see that the left and right accent envelopes through the cycle are complementary, indicating that left and right drums are complementary in vila□bit lay.

Drt tīntāl: From Figure 4.25, we see that the filler strokes in drt tīntāl are restricted to a single filler at half mātrā positions in contrast to three of more fillers in vila□bit. The accents are more regular due to higher tempi associated. Similar to vila□bit, the 9th matra has a strong left accent, which again is a well known difference between theory and practice. The 11th and 14th mātrās have high left accents to support the build up of accents through mātrās 12-14 to indicate the arrival of sam (āmad). It is interesting to note the the vibhāg boundary mātrā 13 has a weaker right accent than the previous mātrā 12 right accent. The stroke on mātrā 13 is skipped and a strong left stroke on mātrā 14 is often played to indicate the approaching sam.

Vila□bit and madhya lay ēktāl: From Figure 4.26, we see that the last matra of the cycle before the sam (mātrā 12) has dense accents, with the final filler strokes having stronger left accents than the sam. This is another example of āmad, where the approach of a sam is distinctly indicated. The mātrās 4 and 10 (both with the ṭhēkā bōl TI RA KI TA) have equal accents in theory. However, mātrā 10 has stronger accents than 4 in practice since it is closer to the sam. TI RA KI TA is often played with more than four strokes towards the end of the matra 4 and 10. Since TI RA KI TA is dense, the mātrā following them (mātrās 5 and 11) have less fillers. In addition, only mātrās 4 and 10 have fillers through the mātrā, while the rest have fillers only towards the end. Vibhāgs 2 and 3 (spanning mātrās 3-6) and vibhāgs 5 and 6 (spanning mātrā 9-×) are identical in theory, but we can see several deviations in performance, with vibhāgs 5 and 6 having stronger left accents since they are closer to sam. Further, the strokes DHIN at mātrā 1 and mātrā 2 are identical in theory, but in practice the DHIN at mātrā 2 is played softer to differentiate it from the DHIN at the sam. The modulation of right accent levels through the cycle is interesting, with stronger accents occurring when the mātrā is less dense with lesser number of accents. This has a functional role in timekeeping - aided by stronger accents and denser mātrās, complementing each other.

Drt ēktāl: Though defined with six vibhāgs in theory, drt ēktāl is described better as having four vibhāgs of 3 mātrās each, as shown in Figure 2.4, with the vibhāgs starting at mātrās 1, 4, 7, and 10. As can be seen from Figure 4.27, the strong right accents due to NA stroke at mātrās 3, 6, 9 and 12 are distinctly seen. This suggests that for dr̥t lay, timekeeping is done more with the sharp right strokes (e.g. 'NA' here) and accentuation can even be at non-vibhāg marker mātrās such as 6 and 12. Even though the last vibhāg starts on matra 10, there is strong right accent on matra 9, an indication of the approaching sam (āmad). The four strokes in TI RA KI TA is often not played in dr̥t, replacing it with just two strokes TRE KE - we see only two accents in mātrās 4 and 10. In addition, due to the dense stroke playing on mātrā 4 and 10, the left accents in mātrā 6 and 12 are quiet with relatively weaker accents. Similar to vila□bit ēktāl, though the first and second matra have equal accented DHIN stroke in theory, DHIN on the second mātrā is played considerably softer with weak accent. As with all tāls in dr̥t lay, the accents on left and right through the cycle are correlated.

Vila□bit and madhya lay jhaptāl: From Figure 4.28, we see that all the NA strokes (mātrās 2, 5, 7, 10) have a strong right accent and weak left accents, as described in theory. There are filler strokes to end the vibhāgs at mātrās 2 and 7. This can be explained with the often played variant of the ṭhēkā (DHI NA TRE-KE DHI DHI NA | TI NA TRE-KE DHI DHI NA). There are further strong accented fillers on mātrās 5 and 10 that act as anchors to indicate the end of half and full cycle.

Dr̥t jhaptāl: Figure 4.29 shows the left accents are as defined in theory (a "textbook" bāyān playing) with basic ṭhēkā playing. The envelope of accents through the cycle is more regular than in vila□bit jhaptāl. In theory, the vibhāg 2 (mātrās 3-5) and vibhāg 4 (mātrās 8-10) are identical, but some deviations can be observed in practice.

Vila□bit and madhya lay rūpak tāl: Rūpak tāl is defined in theory with no left accents on mātrās 1 and 2, but in practice left strokes are often played (with closed strokes than modulated sustained left strokes). This also implies that rūpak tāl having a khālī (0) on the sam does not mean it is less accented. Rūpak tāl is defined to have a 3+2+2 structure, but we see from Figure 4.30 that mātrā 2 has a strong left accent, which acts as an anchor, giving the vila□bit rūpak tāl a pseudo 1+2+2+2 structure, which is close to the tapping of miśra chāpu tāḷa of Carnatic music in practice. This could also be because musicians might play with the same accent on both TIN (mātrās 1 and 2) with a KAT stroke to contrast with the NA stoke which is less left-accented. The vibhāg 2 (mātrās 4-5) and vibhāg 3 (mātrā 6-7) are identical in theory, but in practice the accents differ. Mātrā 5 has the strongest right accent (NA stroke), perhaps indicating āmad. Fillers are more on mātrā 3, to end vibhāg 1. In general, we also see that the fillers get more dense towards the end of vibhāgs.

Dr□t rūpak tāl: From Figure 4.31, the left strokes and accents closely follow the description in theory. The strongest left accent is on mātrā 4, as defined in theory. The vibhāg 2 and 3 are identical with similar accents. Interestingly, the fillers grow through the cycle, becoming more dense towards the end of the cycle.

Applications of the HMR$_{\mathrm{f}}$ dataset

The HMR$_{\mathrm{f}}$ dataset and its subsets HMR$_{\mathrm{l}}$ and HMR$_{\mathrm{f}}$ datasets are intended to be test corpora for several computational rhythm analysis tasks in Hindustani music. Possible tasks include Possible tasks where the dataset can be used include tāl, sam and mātrā tracking, tempo estimation and tracking, tāl recognition, rhythm based segmentation of musical audio, audio to score/lyrics alignment, and rhythmic pattern discovery. In this thesis, these datasets are primarily used for rhythmic pattern analysis and meter inference/tracking. Most of the research results are presented for the three subsets separately, to contrast performance of algorithms across different lay.

4.2.3 Tabla solo dataset

The Mulgaonkar Tabla Solo dataset (MTS dataset) is a parallel corpus of tabla solo compositions with time-aligned scores and audio recordings. We built a dataset comprising audio recordings, scores and time aligned syllabic transcriptions of 38 tabla solo compositions of different forms in tīntāl. The compositions were obtained from the instructional video DVD *Shades Of Tabla* by Pandit Arvind Mulgaonkar[24]. Out of the 120 compositions in the DVD, we chose 38 representative compositions spanning all the gharānās of tabla (Ajrada, Benaras, Dilli, Lucknow, Punjab, Farukhabad).

The booklet accompanying the DVD provides a syllabic transcription for each composition. We used Tesseract(Smith, 2007), an open source Optical Character Recognizer (OCR) engine to convert printed scores to a machine readable format. The scores obtained from OCR were manually verified and corrected for errors, adding the vibhāgs (sections) of the tāl to the syllabic transcription. The score for each composition has additional metadata describing the gharānā, composer and its musical form.

We extracted audio from the DVD video and segmented the audio for each composition from the full audio recording. The audio recordings are stereo, sampled at 44.1 kHz and have a soft harmonium accompaniment. A time aligned syllabic transcription for each score and audio file pair was obtained using a spectral flux based onset detector (Bello et al., 2005) followed by manual cor-

[24] http://musicbrainz.org/release/220c5efc-2350-43dd-95c6-4870dc6851f5

bōls	Symbol	# Instances
D, DA, DAA	DA	
N, NA, TAA, TUN	NA	
DI, DIN, DING, KAR, GHEN	DIN	
KA, KAT, KE, KI, KII	KI	
GA, GHE, GE, GHI, GI	GE	
KDA, KRA, KRI, KRU	KDA	
TA, TI, RA	TA	
CHAP, TIT	TIT	
DHA	DHA	
DHE	DHE	
DHET	DHET	
DHI	DHI	
DHIN	DHIN	
RE	RE	
TE	TE	
TII	TII	
TIN	TIN	
TRA	TRA	
Total	-	8200+

Table 4.15: The bōls used in tabla, their grouping, and the symbol we use for the syllable group in this thesis.

rection. The dataset contains about 17 minutes of audio with over 8200 syllables.

The syllables of tabla vary marginally within and across gharānās, several bōls can represent the same stroke on the tabla. To address this issue, we grouped the full set of 41 syllables into timbrally similar groups resulting into a reduced set of 18 syllable groups as shown in Table 4.15. Though each syllable on its own has a functional role, this timbral grouping is presumed to be sufficient for discovery of percussion patterns. For the remainder of the thesis, we limit ourselves to the reduced set of syllable groups and use them to represent patterns. For convenience, when it is clear from the context, we call the syllable groups as just syllables and denote them by the symbols in Table 4.15. The table also lists the number of instances in the dataset for each group syllable.

The dataset is freely available for research purposes through a central online repository[25]. The dataset was created in collaboration with Swapnil Gupta and more details are described in the masters thesis by Gupta (2015). The dataset is useful both for building isolated stroke timbre models and for a comprehensive evaluation of tabla solo pattern transcription and discovery, as used by Gupta, Srinivasamurthy, Kumar, Murthy, and Serra (2015). The scores in the dataset can be used to do symbolic analysis of percussion patterns.

4.2.4 Mridangam datasets

There are two percussion datasets for Carnatic music: a collection of audio examples of mridangam strokes compiled by Akshay Anantapadmanabhan, and a parallel corpus of scores and audio recordings of mridangam solos played by Padmavibhushan Dr. Umayalpuram K. Sivaraman and compiled by IIT Madras, Chennai, India.

Mridangam stroke dataset

The Anantapadmanabhan Mridangam Strokes dataset (AMS dataset)[26] is a collection of 7162 audio examples of individual strokes of the mridangam in various tonics. The dataset can be used for training models for each mridangam stroke (Anantapadmanabhan, Bellur, & Murthy, 2013). The dataset comprises of ten different strokes played on mridangams with six different tonic values. The dataset is described in Table 4.16, with stroke labels along rows and tonic values along columns. The audio examples were recorded from a professional Carnatic percussionist in semi-anechoic studio conditions using SM-58 microphones and an H4n ZOOM recorder. The audio was sampled at 44.1 kHz and stored as 16 bit wav files.

Mridangam solo dataset

The UKS Mridangam Solo dataset (UMS dataset) is a transcribed collection of two tani-āvartanas (solo performance by the percussion ensemble) played by the renowned mridangam maestro Pad-

[25] http://compmusic.upf.edu/tabla-solo-dataset

[26] http://compmusic.upf.edu/mridangam-stroke-dataset

	B	C	C#	D	D#	E	Total
Bheem	5	3	1	0	15	25	**49**
Cha	57	50	54	67	49	53	**330**
Dheem	127	86	78	12	111	54	**468**
Dhin	48	48	63	12	198	113	**482**
Num	81	98	97	18	143	60	**497**
Ta	145	165	217	180	119	105	**931**
Tha	200	185	211	224	196	160	**1176**
Tham	88	80	35	29	92	50	**374**
Thi	438	334	369	283	444	345	**2213**
Thom	136	80	72	91	128	135	**642**
Total	**1325**	**1129**	**1197**	**916**	**1495**	**1100**	**7162**

Table 4.16: The Anantapadmanabhan Mridangam Strokes dataset. The row and column headers are the stroke labels and the tonic values, respectively.

mavibhushan Umayalpuram K. Sivaraman. The audio was recorded at IIT Madras, India and annotated by professional Carnatic percussionists (Kuriakose, Kumar, Sarala, Murthy, & Sivaraman, 2015).

Since percussion in Carnatic music is organized and transmitted orally with the use of onomatopoeic syllables representative of the different strokes of the mridangam, a syllabic representation of the tani and the patterns provides a musically meaningful representation for analysis. The dataset uses such a representation. The dataset consists of two tani-āvartanas played on a mridangam tuned to tonic C#, one played in vila□bita ādi tāḷa (a cycle of 16 beats) and the other played in rūpaka tāḷa. Each tani is about 12 minutes long. Each tani has been segmented into short phrases and each phrase has been transcribed into its constituent strokes, represented as syllables. The trancriptions also include pauses (denoted by ,) and change in speed (denoted by { and }). The combined duration of both the tanis is about 24 minutes and consists of 8863 strokes.

Both tanis were recorded in studio-like conditions using a Zoom H4n recorder with an SM 57 for the treble head (right) and SM 58 for the base head (left) of the mridangam. The audio files are mono, sampled at 44.1KHz, and stored in 16 bit .wav format.

Solkaṭṭu	Symbol	# Instances	
		tani-1	tani-2
achapu	ACH	400?	450?
achaputha, achaputhom	ACHt		
chapu	CHP		
chaputha, chaputhom	CHPt		
dheem, dheemtha	DHM		
dhi3	DH3		
dhi3g, dhi3tha, dhi3thom	DH3t		
dhi3m, dhi3mg	DH3m		
dhi4, dhi4p	DH4		
dhi4g, dhi4tha	DH4t		
dhin	DHIN		
dhing, dhint, dhintha, dhinthom, dhintom, dot	DHINt		
lf, lgm	LF		
lfg, lfthom	LFt		
nam, rnam	NAM		
namg, namtha, namthom, rnamtha, rnamthom	NAMt		
ot, tha	THA		
ta, tam	TA		
tatha, tathom	TAT		
thom	THOM		
tmg, 3mg	TMG		
Total	-		

Table 4.17: The solkaṭṭus used in mridangam, their grouping, and the symbol we use for the syllable group in this thesis.

The audio file has been segmented into short musically relevant phrases by professional musicians. The syllabic transcription of each phrase was done by professional Carnatic percussionists. The transcription is not time aligned, but only a sequence of the strokes played in the phrase.

Similar to the MTS dataset, we grouped the full set of XX syllables into timbrally similar groups resulting into a reduced set of XX syllable groups as shown in Table 4.15, assuming this timbral

Dataset	Bangu	Daluo	Naobo	Xiaoluo	**Total**
Training	59	50	62	65	**236**
Test	1645	338	747	291	**3021**

Table 4.18: The Jingju percussion instrument dataset (JPI dataset) showing the number of examples for each instrument in the training and test dataset.

grouping to be sufficient for discovery of timbrally similar percussion patterns. The entire set of strokes, and their notation used in the transcription files can be seen in Table 4.17. The list also specifies the number of occurences of each stroke in the tanis.

The dataset can be used for several MIR tasks such as onset detection, percussion transcription, rhythm and percussion pattern analysis, and mridangam stroke modeling. The dataset (audio + annotations) is freely available for research purposes [27] and has been recently used by Kuriakose et al. (2015) in their work.

4.2.5 Jingju percussion instrument dataset

The Jingju percussion instrument dataset (JPI dataset) is an annotated collection of Beijing opera percussion instruments, with audio and time aligned onset annotations. The dataset is split into training set with audio files containing single strokes of individual percussion instruments and a test dataset that has the whole percussion ensemble playing together.

The dataset was built by with Mi Tian at the Centre for Digital Music (C4DM), Queen Mary University of London. The dataset was built by recording sound samples with professional musicians in studio conditions at C4DM. The audio was recorded in mono using an AKG C414 microphone at a sampling rate of 44.1 KHz.

The dataset, shown in Table 4.18, consists of recordings of the four percussion instrument classes: bangu, daluo, naobo and xiaoluo. Unlike pitched instruments, most idiophones cannot be tuned. These percussion instruments are made from metal casting or wood carving hence subtle differences might exist between the physical properties of individual instruments even of the same kind.

[27] http://compmusic.upf.edu/mridangam-tani-dataset

For each kind of the above instruments, sound samples of 2-4 individual instruments were recorded, played with different playing styles commonly used in Beijing Opera performances with a hope to achieve a better coverage of timbre and variations of playing techniques.

The training set consists of short audio samples with single strokes of each individual instrument that capture most of the possible timbres of the instrument that exist in Beijing Opera. For the test dataset, the individually recorded instrument examples were manually mixed together using Audacity[28] into 30-second long tracks, with possibly simultaneous onsets to closely reproduce the real world conditions. The examples in training and test dataset are mutually exclusive.

For the onset annotations, manual labeling of onset locations is tedious and time consuming, especially for complex ensemble music consisting of instruments with diverse properties. The onset ground truth was constructed by the taking the average onset locations marked by three participants without any Beijing Opera background. Participants were asked to mark the onset locations in each recording using the audio analysis tool Sonic Visualiser (Cannam et al., 2010) displaying the waveform and corresponding spectrogram.

The set of training examples are freely available for research and reuse [29]. The dataset can be used for training models for each percussion instrument class, and MIR tasks such as percussion instrument identification, source separation, and instrument-wise enhanced onset detection, as used by Tian, Srinivasamurthy, Sandler, and Serra (2014).

4.2.6 Jingju percussion pattern dataset

The Jingju percussion pattern dataset (JPP dataset) is a collection of audio examples and scores of percussion patterns played by the percussion ensemble in Jingju. The dataset was built from commercial jingju aria recordings with the help of Rafael Caro, a musicologist working on jingju.

[28] http://audacity.sourceforge.net

[29] http://compmusic.upf.edu/bo-perc-dataset

Pattern Class	ID	Instances	$\overline{LEN}$ (σ)
dǎobǎn tóu 【导板头】	1	66	8.70 (1.73)
màn chángchuí 【慢长锤】	2	33	13.99 (4.47)
duótóu 【夺头】	3	19	7.18 (1.49)
xiǎoluó duótóu 【小锣夺头】	4	11	8.16 (2.15)
shǎnchuí 【闪锤】	5	8	10.31 (3.26)
Total		**133**	**9.85 (3.69)**

Table 4.19: The Jingju percussion pattern dataset (JPP dataset). The last column is the mean pattern length and standard deviation in seconds.

The dataset is a collection of 133 audio percussion patterns spanning five different pattern classes described in Section 2.2.6. The audio files are short segments containing one of the above mentioned patterns. The audio is stereo, sampled at 44.1 kHz, and stored as wav files. The segments were chosen from the introductory parts of arias. The recordings of arias are from commercially available releases spanning various artists. The audio and segments were chosen carefully by a musicologist to be representative of the percussion patterns that occur in Jingju. The audio segments contain diverse instrument timbres of percussion instruments (though the same set of instruments are played, there can be slight variations in the individual instruments across different ensembles), recording quality and period of the recording. Though these recordings were chosen from introductions of arias where only percussion ensemble is playing, there are some examples in the dataset where the melodic accompaniment starts before the percussion pattern ends.

Each of the audio patterns has an associated syllable level transcription of the audio pattern. The transcription is obtained from the score for the pattern and is not time aligned to the audio. The transcription is done using the reduced set of five syllables described in Table 2.1 and is sufficient to computationally model the timbres of all the syllables. The annotations are stored as Hidden Markov model Toolkit (HTK)[30] label files. There is also a single master label file provided for batch processing using HTK.

The annotations are publicly shared and available to all. The

[30] http://htk.eng.cam.ac.uk/

audio is from commercially available releases and can be easily accessed using the associated MusicBrainz IDs. The dataset can be used for instrument-wise onset detection and percussion pattern transcription and classification(Srinivasamurthy, Caro, Sundar, & Serra, 2014).

4.2.7 Other evaluation datasets

Are descriptions of Turkish and Cretan datasets needed, since not many results are included ? There are other datasets on which we present some evaluation results.

Turkish rhythm dataset

The Turkish rhythm dataset was compiled and annotated by Andre Holzapfel (citation needed) and is an extended version of the annotated data used by Srinivasamurthy, Holzapfel, and Serra (2014). It includes 82 excerpts of one minute length each, and each piece belongs to one of three rhythm classes that are referred to as *usul* in Turkish Art music. 32 pieces are in the $9/8$-usul *Aksak*, 20 pieces in the $10/8$-usul *Curcuna*, and 30 samples in the $8/8$-usul *Düyek*.

Cretan music dataset

The corpus of Cretan music consists of 42 full length pieces of Cretan leaping dances compiled and annotated by Andre Holzapfel (citation needed). While there are several dances that differ in terms of their steps, the differences in the sound are most noticeable in the melodic content, and we consider all pieces to belong to one rhythmic style. All these dances are usually notated using a $2/4$ time signature, and the accompanying rhythmical patterns are usually played on a Cretan lute. While a variety of rhythmic patterns exist, they do not relate to a specific dance and can be assumed to occur in all of the 42 songs in this corpus.

Ballroom dataset

The ballroom dataset includes beat and bar annotations audio recordings of several dance styles sourced from `BallroomDancers.com` and was first introduced by Gouyon et al. (2006). The beat and bar annotations were then added by Krebs, Böck, and Widmer (2013).

The ballroom dataset contains eight different dance styles (Cha cha, Jive, Quickstep, Rumba, Samba, Tango, Viennese Waltz, and (slow) Waltz) and has widely used for several MIR tasks such as genre classification, tempo tracking, beat and downbeat tracking XX citations.

It consists of 697 30 seconds-long audio excerpts (sampled at 11.025 kHz) and has tempo and dance style annotations. The dataset contains two different meters (3/4 and 4/4) and all pieces have constant meter. The tempo restrictions given the dance style label from http://www.ballroomdancers.com/Dances/ were used to annotate the beats and downbeats at the correct metrical level.

The ballroom dataset is used as a state of the art dataset to present several evaluations of the algorithms and approaches presented in thesis - to compare performance with the state of the art, and to test if the proposed approaches scale and extend to different music genres and cultures. ***Chapter summary***

Chapter **5**

Meter inference and tracking

> ...the first beat (sam) is highly significant structurally, as it frequently marks the coming together of the rhythmic streams of soloist and accompanist, and the resolution point for rhythmic tension.
>
> Clayton (2000, p. 81)

Meter analysis of audio music recordings is an important MIR task. It provides useful musically relevant metadata not only for enriched listening, but also for pre-processing of music for several higher level tasks such as section segmentation, structural analysis, and defining rhythm similarity measures.

To recapitulate, meter analysis aims to time-align a piece of audio music recording with several defined metrical levels such as tatum, tactus, measure (bar). In addition, it also tags the recording with additional meter and rhythm related metadata such as time signature, median tempo and salient rhythms in the recording. Within the context of Indian music, meter analysis aims to time-align and tag a music recording with tāḷa related events and metadata.

This chapter aims to address some of these important tasks related to meter analysis within the context of Indian art music, presenting several approaches and a comprehensive evaluation of those approaches. The main aims of the chapter are:

1. To address meter analysis tasks for the music cultures under study - Carnatic and Hindustani music. The tasks of meter inference, meter tracking, and informed meter tracking are addressed in detail - formulation of these tasks, and propose several approaches to address the tasks.

2. To present a detailed description of the state of the art and the proposed Bayesian Models and inference schemes for meter analysis.

3. To present an evaluation of the state of the art meter tracking approaches based on Bayesian models and explore extensions to those approaches, for the rhythm annotated datasets of Carnatic and Hindustani music. A comprehensive performance analysis is presented for these approaches, identifying their strengths and limitations in the tasks under study.

5.1 The meter analysis tasks

We describe the meter analysis tasks addressed in thesis, from the least informed to the most informed. This order of tasks also emphasizes different practical scenarios for such tasks, and hence the results can indicate the type of task and the additional information to be provided to achieve the level of performance required for an application. We will also describe how the set of tools and approaches described in the chapter can be adapted and used in each of these tasks, making the task of meter analysis flexible to the audio data and the related additional metadata that we can obtain. We will also describe how the set of tools and approaches described in the chapter can be adapted and used in each of these tasks, making the task of meter analysis flexible to the audio data and the related additional metadata that we can obtain.

Meter Inference

Given an audio music recording, meter inference aims to estimate the rhythm class (or meter type), possibly time-varying tempo, beats and downbeats. In the context of Carnatic music, the task of meter inference aims to recognize the tāḷa, and estimate the time varying tempo (τ_o or τ_b), the beat locations, and the sama (downbeat) locations. Since some of the beats correspond to the a□ga boundaries, with the sama and numbered beat locations (beat number in the cycle), the a□ga (section) boundaries can be indirectly inferred, e.g. the beats 1 (sama), 5, 7 mark the start of the three sections of the ādi tāḷa. Similarly, for Hindustani music, meter inference task aims to recognize the tāl, and estimate the time varying tempo (τ_o), the mātrā and the sam locations. With the numbered mātrā and sam locations, the vibhāg boundaries can be indirectly inferred, e.g. the mātrās 1 (sam), 3, 6, 8 indicate the start of the four sections in jhaptāl. For Carnatic music, in addition to the beats, we can also estimate the sub-division akṣaras, which can be grouped into beats.

Without any prior information on metrical structure, meter inference is a difficult task owing to the large range of tempi, different tāḷas. The problem is further made harder due to several tāḷa having similar structure. In Carnatic music, it is quite often possible that the same composition is performed in two different tāḷas, which further can lead to confusion. From a practical application point of view, most of commercially released music in both Carnatic and Hindustani music has the name of the tāḷa as a part of the editorial metadata, and hence tāḷa recognition is a redundant task. Even within a live concert, the musician announces the tāḷa of the piece, or shows it with hand gestures in Carnatic music. Meter inference is used a baseline task to understand the complexity of uninformed meter analysis.

Meter tracking

Given that the tāḷa of an audio music piece is often available as editorial metadata, the most relevant meter analysis task for Indian art music is meter tracking. Given an audio music recording and the rhythm class (or meter type) of the music piece, meter tracking aims to estimate the time varying tempo, the beat and the downbeat locations. In the context of Carnatic music, meter tracking aims to track the time varying tempo, beats and the sama from an audio music recording, given the tāḷa. For Hindustani music, given the tāl, the task aims to track the time varying τ_o, the mātrā and sam. The section boundaries of the tāḷa can be indirectly inferred as explained earlier. Assuming that the tāḷa, and hence the metrical structure is known in advance is a fair and practical assumption to make, and we explore if providing this information helps to track the metrical structure better. Meter tracking is the main problem and most comprehensively addressed task in this thesis. We explore different approaches and evaluate them on the rhythm annotated datasets for Carnatic and Hindustani music. The proposed extensions and enhancements are also evaluated for the task of meter tracking.

Informed meter tracking

Informed meter tracking is a sub-task of meter tracking in which some additional information apart from the meter type is provided along with the audio recording. The additional information could be in the form of a tempo range, a few instances of beats and downbeats annotated, or even partially tracked metrical cycles. These additional metadata could come from manual annotation or as - an output of other automatic algorithms, e.g. the median tempo of a piece an be obtained from a standalone tempo estimation algorithm, or some melodic analysis algorithms might output (with a high probability) some beats/downbeats as a byproduct. Even from a practical standpoint, it is useful to explore informed meter tracking. While it is prohibitively resource intensive to manually annotate all the beats and downbeats of a large music collection, it might be possible to seed the meter tracking algorithms with the first few beats and downbeats, which could improve meter tracking performance. We aim to explore these questions, to see whether providing additional information can improve meter tracking performance. In specific, we explore two variants of informed meter tracking:

1. Tempo-informed meter tracking in which the median tempo of the piece is provided as an additional input to the meter tracking algorithm. Providing the median tempo intends to help reduce tempo octave errors, tracking the metrical cycles at the correct metrical level instead of tracking half and double cycles. The median tempo can be obtained through simpler state of the art tempo estimation algorithms outlined in Section 2.3.4 (one such algorithm for Carnatic music is also described later in the chapter in Section 5.2.1). Since the tempo of a piece can vary over time, a narrow range of tempo in the piece can also be provided in addition or in lieu of the median tempo.

2. Tempo-sama-informed meter tracking in which the median tempo and the first downbeat location in the excerpt are provided as additional inputs to the meter tracking algorithm. The practical scenario for such a case is a semi-supervised meter tracking system, where a human listener can tap along to one or some of the downbeats of the piece and an automatic meter tracker can track the rest of the piece. In this thesis, we only explore the use of first downbeat of the piece in informed meter tracking.

There are other meter analysis tasks that have been addressed in MIR, such as beat tracking, and downbeat tracking from the set of known beats. The task of beat tracking as defined in the state of art is ill defined in Indian art music, due to possibly non-isochronous pulsation. We can adapt the task and track a uniform pulsation as the beat. Howeve, since the tasks of meter inference and meter tracking aim to track all the relevant events of the metrical cycle, the task of beat tracking is subsumed in those tasks. We do not address specifically the task of beat tracking in Indian music directly, but as a sub-task of the meter tracking/inference tasks. Estimating the downbeats and the start of measure from a set of beats, as done by Davies and Plumbley (2006); Hockman et al. (2012) is also handled as a sub-task within the joint estimation of tempo, beats and the downbeats.

We now describe the approaches to these tasks, starting with some preliminary approaches followed by Bayesian models. With Bayesian models, several different extensions are proposed over the state of the art models.

5.2 Preliminary experiments

The preliminary experiments around the task of meter analysis are exploratory experiments with existing features, rhythm descriptors, methods and algorithms to gain insights into the problem and test their relevance and utility in these tasks. The aim of including them in thesis is to gain useful insights and understand the limitations of those algorithms in meter analysis tasks in Indian art music. Only a selection of them are described here, primarily for Carnatic music, as a base for improved Bayesian models for meter analysis. We proposed a novel meter tracking algorithm in Carnatic music (Srinivasamurthy & Serra, 2014) using pre-existing tools and rhythm descriptors, which is described in detail. The features and tools are explained as a part of the proposed meter tracking algorithm, emphasizing on their utility.

5.2.1 Meter tracking using dynamic programming

The primary philosophy of meter tracking is to incorporate specific knowledge of the rhythmic structures we aim to estimate, which is also used in this approach. However, it aims to estimate the components of meter separately using a descriptor for each music concept. Using Carnatic music as an illustration, the algorithm estimates the akṣara period τ_o, the akṣara pulse locations, and the sama. For estimating these components, a set of rhythm descriptors is first computed from the audio that indicates the possible candidates for each musical concept. The periodicity and the relationships between these structures are then utilized to estimate the components. This framework can be generalized to estimating other rhythmic structures by suitably modifying the audio descriptor for the specific music culture and the rhythmic structure under consideration.

The algorithm for Carnatic music is explained in detail in this section. A hypothesis is that the akṣara pulses can be estimated from the onsets of mridangam, and hence a percussion onset based rhythm descriptor (Bello et al., 2005) is useful for tracking the akṣara pulses. Tempogram, a mid-level tempo representation for music signals was proposed by (Grosche & Müller, 2011), is used to track the time-varying akṣara period. A novelty function is computed using a self similarity matrix constructed using frame level onset and timbral features. These are then used to estimate possible akṣara and sama candidates, followed by a candidate selection based on periodicity constraints, which leads to the final estimates. The features and the approach are explained further in detail.

Pre-processing: Percussion enhancement

The akṣara pulse most often coincides with the onsets of mridangam strokes. To enhance the mridangam onsets, percussion enhancement is performed on the downmixed mono audio signal $f[n]$, as it has been shown to improve beat tracking performance in pieces with predominant vocals by J. Zapata and Gómez (2013). The

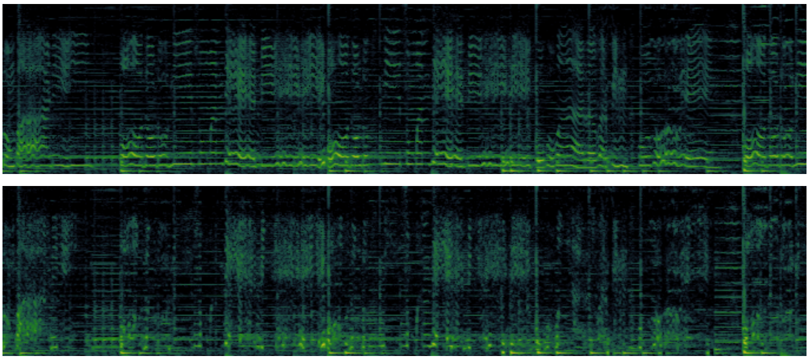


Figure 5.1: An illustration of percussion enhancement on a short audio excerpt of Carnatic music. The figure shows the spectrogram of the audio excerpt, before percussion enhancement (top panel) and after percussion enhancement by suppressing the lead melody (bottom panel). The lead melody is suppressed, while the tambūra (drone) is still present.

predominant melody is estimated using the algorithm proposed by Salamon and Gómez (2012) using which, the harmonic component of the signal is extracted using a sinusoidal+residual model proposed by (Serra, 1997). The percussion enhanced signal $f_p[n]$, with the harmonic component suppressed, is used for further processing (Figure 5.2(c)). An illustration of percussion enhancement for a short audio exceprt of Carnatic music is shown in Figure 5.1.

Akṣara period and pulse tracking

The spectrogram of $f_p[n]$ is used to compute two frame level spectral-flux based onset detection functions (Bello et al., 2005) computed every 11.6 ms. The first function ($d_f[k]$) uses the whole frequency range of the spectrogram and the other function computes the spectral flux only in the range of 0-120 Hz ($d_l[k]$) and captures the low frequency onsets of the left (bass)-side of the mridangam.

The function $d_f[k]$ is used to compute a Fourier-based Tempogram proposed by (Grosche & Müller, 2011), computed every 0.25 second using a 8 second long window (Figure 5.2(d)). If the time indices at which the tempogram is computed is denoted with k, $(1 \leq k \leq K)$, the most predominant τ_o curve can be tracked by estimating the best path $\Gamma = \{\gamma_i : k = 1, 2, \cdots, K\}$ through the

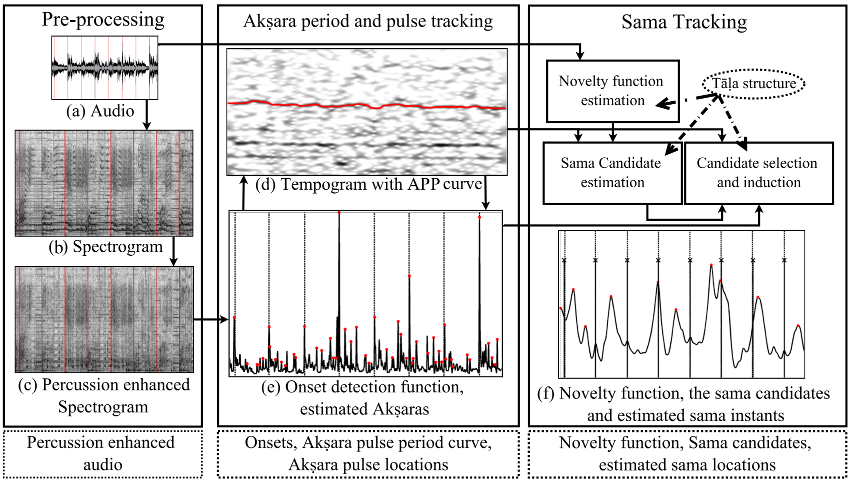

Pre-processing
(a) Audio
(b) Spectrogram
(c) Percussion enhanced Spectrogram
Akṣara period and pulse tracking
(d) Tempogram with APP curve
(e) Onset detection function, estimated Akṣaras
Sama Tracking
Novelty function estimation
Tāḷa structure
Sama Candidate estimation
Candidate selection and induction
(f) Novelty function, the sama candidates and estimated sama instants
Percussion enhanced audio
Onsets, Akṣara pulse period curve, Akṣara pulse locations
Novelty function, Sama candidates, estimated sama locations

Figure 5.2: Block Diagram of the algorithm showing the signal flow and representative illustrations of different stages of the algorithm. The important outputs at each stage are also shown at the bottom. In each panel, the vertical lines that run through the panel indicate the sama ground truth instants. The estimated sama/akṣara candidates are shown with red dots and the estimated sama are shown with ×.

tempogram matrix **G** that provides a balance between tempogram amplitude at time index k, $\mathbf{G}_{\gamma_k,k}$, and the local continuity of τ_o. A cost function, that is an extended version of the one used by Wu et al. (2011), is defined as shown in Eq. 5.1.

$$J_1\left(\Gamma,\theta_1,\theta_2\right) = \sum_{k=1}^{K}\mathbf{G}_{\gamma_k,k} - \sum_{k=1}^{K-1}\left(\theta_1\left|\gamma_k-\gamma_{k+1}\right| + \theta_2\,\mathfrak{D}\left(\frac{\gamma_k}{\gamma_{k+1}}\right)\right) \tag{5.1}$$

The function $\mathfrak{D}(\gamma_k/\gamma_{k+1})$ is an extra penalty term to penalize tempo doubling and halving between adjacent frames, and the weights θ_1(=0.01) and θ_2(=10^6) provide different weights to the three terms. Based on observations from the $\texttt{CMR}_\mathrm{f}$ dataset, the search for the best path through the tempogram is restricted between the range of 120 to 600 APM (akṣaras per minute). The above cost function is solved using a dynamic programming (DP) based approach to obtain a τ_o curve that is then corrected for tempo doubling/halving errors, if any, to obtain the final curve Γ^* (Figure 5.2(d), Γ^* is shown as a thick red line). Using the τ_o and the tāḷa information, we can obtain the time varying τ_s curve for the piece by multiplying the τ_o by the

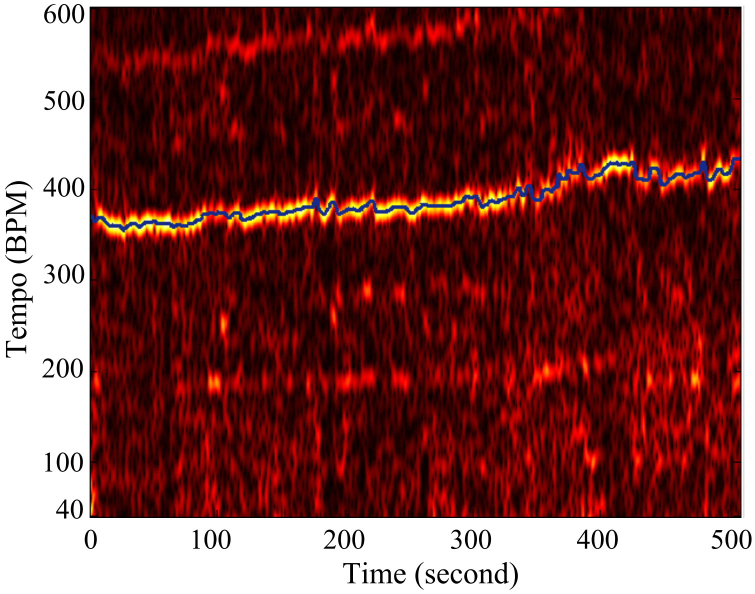


Figure 5.3: Estimated time varying tempo curve (shown in blue) plotted on top of a tempogram, for a Carnatic music piece (see footnote). In the piece, apart the local tempo variations, we can see that the tempo increases with time. The tempogram shows high values in tempo octave related bands, with the highest value (in yellow) at the estimated τ_o.

number of akṣaras in a cycle of the tāḷa. A further example of a tempogram and the estimated time varying τ_o curve for a piece of Carnatic music[1] from $\texttt{CMR}_\texttt{f}$ dataset is shown in Figure 5.3. The figure shows the variations in tempo through a Carnatic music piece.

The akṣara pulse locations predominantly lie on strong Mridangam onsets. The akṣara pulse candidates are estimated as the peaks of the function $d_f[k]$. Using these κ candidate peaks $\{o_i\}$, $i = 1, 2, 3, \cdots, \kappa$, with locations t_i and peak amplitude ξ_i, a cost function is setup as shown in Eq. 5.2 to select the best candidates that provide a balance between the amplitude of these candidates and a periodicity provided by the estimated akṣara period. The best set of candidates $\{o_i^*\} \subset \{o_i\}$ are estimated using a DP approach

[1] Kamalamba, a kṛti in ānandabhairavi rāga and miśra chāpu tāḷa, from the album Madrasil Margazhi 2005 by Aruna Sairam: http://musicbrainz.org/recording/3baa722d-480e-4ae7-8559-a88dce41e1d4

(Figure 5.2(e)).

$$J_2\left(\{o_i\},\delta\right) = \sum_{i\subset\{1,2,\cdots,\kappa\}} \left(\xi_i + \delta\,\Upsilon(t_i, t_{i+1}, \Gamma)\right) \tag{5.2}$$

The function $\Upsilon(t_i, t_{i+1}, \Gamma)$ is a function that returns a exponentially decaying weight based on the time difference between t_i and t_{i+1} in relation to the local akṣara period, γ_{t_k}. The parameter δ(=3) provides a tradeoff between the two terms.

Sama Tracking

The use of Mel-Frequency cepstral co-efficients (MFCC) as features for timbral characteristics is explored. As a detection function for sama ($d_s[k]$), a novelty function is computed through the diagonal processing of a self similarity matrix (Foote, 2000) constructed using frame level z-score MFCC features from audio (using audio processing library *Essentia* (Bogdanov et al., 2013)) as shown in Figure 5.2(f). Based on the $\bar{\tau}_s$ shown in Table 4.6, a checkerboard kernel with size of 7, 3, 4, and 3 seconds is used for the tāḷas ādi, rūpaka, miśra chāpu and khaṇḍa chāpu respectively so that the novelty function is computed over about an āvartana.

The peaks of the novelty function $d_s[k]$ indicate a significant change of timbre at that time. Starting with the premise that timbral change is an important indicator of sama location, the peaks of the novelty function are used to estimate sama candidates. Two methods are explored to estimate the candidates. In Method-A, to uniformly choose sama candidates throughout a piece, the piece is cut into segments of length 120, 40, 40 and 30 seconds for ādi, rūpaka, miśra chāpu and khaṇḍa chāpu respectively ($\sim$10 āvartanas), and the top five most prominent peaks in each segment of the piece are estimated as sama candidates ($\{s_i^A\}$).

Another approach, Method-B, is also proposed for candidate estimation that enforces a periodicity constraint while estimating sama candidates. Starting from the peaks of $d_s[k]$ and estimated τ_s curve, for a specific peak, the tāḷa cycle is induced starting from it. The number of other peaks that would support such an induced tāḷa is assigned as the weight of the specific peak. The peaks are then rank ordered using this weight and the top ten ranked peaks are chosen as the sama candidates ($\{s_i^B\}$). In addition, two random baseline methods RB-1 and RB-2 are created to compare the performance. In RB-1, a randomly chosen constant τ_s between 1-8 seconds is used, and a random starting time between 0-2 seconds to induce periodic samas. In RB-2, the estimated τ_s is used with 10 randomly chosen akṣara locations from $\{o_i\}$ as sama candidates. RB-1 neither uses the τ_s, nor the candidate estimation using $d_s[k]$, while RB-2 uses the estimated τ_s but not the candidate estimation using $d_s[k]$.

Starting with the sama candidates obtained either from Method-A or Method-B, for each candidate, the tāḷa cycles are induced based on local τ_s period obtained from the τ_s curve. For each seed, the next and previous three estimated cycle periods are searched for onset peaks in $d_f[k]$ that support a sama. If a supporting onset is found, it is marked as a sama and the algorithm proceeds further with the new estimated onset as the new anchor. The induction is stopped from a candidate when it does not lead to such a supporting onset. Hence for each candidate, an estimated sama sequence is obtained. Since all candidates are not necessarily sama locations, though the estimated τ_s is right, the sequences can have different offsets.

The final step of the algorithm is to shift, align and merge these sequences obtained from each candidate. Starting with the longest sama sequence that has been estimated, other sequences are merged into this based on maximum correlation between the sequences. The merging of these sequences often leads to many sama estimates concentrated around the true location of sama due to small offsets. Since the left bass onsets on the Mridangam are often strong at the samas, all groups of sama estimates that are closer than 1/3rd of τ_s are merged into a single sama estimate aligned with the closest left stroke onset obtained from $d_l[k]$. This forms the final set of sama locations $\{s_{t_i}\}$ estimated from the candidates and the onset detection function, as shown in Figure 5.2(f) with ×.

Results

The annotated `CMR`$_{\mathtt{f}}$ dataset has annotations only for beats and samas of the piece. From the sama locations, we can obtain the ground truth for τ_s curve, and hence the ground truth for τ_o curve. Since we do not have the ground truth for akṣara locations, we present the results only for iai and sama tracking.

Measure	CML	AML
$\bar{\tau}_o$ estimation	0.812	0.989
τ_o tracking	0.804	0.963

Table 5.1: Accuracy of akṣara period tracking on the CMR$_\mathrm{f}$ dataset. The values are measured using a 5% tolerance, at both correct metrical level (CML) and allowed metrical levels (AML).

Variant	$\mathfrak{p}$	$\mathfrak{r}$	$\mathfrak{f}$	$\mathfrak{I}$(bits)	Cand. Accu. (%)
Method-A	0.290	0.190	0.216	1.17	20.46
Method-B	0.246	0.202	0.215	1.25	27.85
RB-1	0.155	0.175	0.137	0.40	-
RB-2	0.228	0.200	0.206	1.11	15.3

Table 5.2: Accuracy of sama tracking. The measures $\mathfrak{p}$: Precision, $\mathfrak{r}$: Recall, $\mathfrak{f}$: f-measure, $\mathfrak{I}$: Information Gain, are shown. The values are mean performance over the whole CMR$_\mathrm{f}$ dataset. The last column shows the fraction (as a percentage) of the estimated sama candidates that are true samas.

The performance of akṣara period tracking is measured by comparing the ground truth akṣara period curve with the estimated curve with an error tolerance of 5%. The results of median akṣara period estimation computed from the whole akṣara period curve is also reported. Further, since there can be tempo doubling and halving errors, the accuracies are reported at the annotated correct metrical level (CML) and then using a weaker AML measure that allows tempo halving and doubling (AML - allowed metrical levels). The results are presented in Table 5.1. We see that an acceptable level accuracy is achieved at CML for both median akṣara period estimation and akṣara period tracking and further, there is not a significant difference between their performances, indicating that the algorithm can track changes in tempo effectively. Even when the akṣara period tracking fails at CML, the algorithm tracks a metrically related akṣara period, as indicated by a high AML accuracy.

For sama tracking, the accuracy of estimation is reported with a margin of 7% the annotated τ_s of the piece. Given the ground truth and the estimated sama time sequence, we use the common evaluation measures used in beat tracking - precision, recall, f-measure and Information Gain (McKinney, Moelants, Davies, & Klapuri, 2007) to measure the performance. The results are shown in Table 5.2, which also shows the accuracy of sama candidate estimation. The results for RB-1 and RB-2 show mean performance over 100 and 10 experiments for each piece, respectively.

We see that the performance of sama candidate estimation and sama tracking is poor in general, with samas correctly tracked only in about a fifth of cases. The precision is higher than recall in all cases, and Information Gain is lower than a perceptually acceptable threshold (J. R. Zapata, Holzapfel, Davies, Oliveira, & Gouyon, 2012). Both methods perform better than RB-1, but have comparable results with RB-2, with a slightly better f-measure performance (statistically significant in a Mann–Whitney U test at $p < 0.05$). This shows that the estimated inter-sama interval (τ_s) is useful for sama estimation, whereas candidate estimation using novelty function is only marginally useful. The poor performance can be mainly attributed to poor sama candidate estimation with either of Method-A or Method-B. This is further substantiated by the fact that Method-B achieves an F-measure of 0.436 and an information gain of 1.70 bits when at least half the estimated candidates are true samas. This clearly shows that the performance of sama tracking crucially depends on sama candidate estimation. There are only four pieces (among all pieces with accurate τ_s estimation) in which all the estimated candidates are true samas, for which an F-measure of 0.894 and a information gain of 3.51 bits is achieved. This clearly indicates that the novelty function from which the sama candidates were estimated is not a very good indicator of sama, and better descriptors need to be explored.

Conclusions

The presented approach to meter tracking with relevant rhythm descriptors for tempo, akṣara, and sama and a hierarchical framework is promising, but has several limitations. The onset detection functions have information about surface rhythms and hence can be utilized for tempo tracking and akṣara pulse tracking, but the novelty function used presently is not a good indicator for sama. Further, it is observed that akṣara pulse period tracking performs to an acceptable accuracy for practical applications, while sama tracking is challenging and performs poorly primarily due to poor sama candidate estimation. Though the tempo, akṣara and sama are related, they were tracked separately. Even though information from tempo estimation was used in estimating the sama, a joint estimation of the meter components is desired, since it can tightly couple these related components together.

The approach uses the musical characteristics in isolation, without considering the interdependence between them. A model that can more effectively model the underlying metrical structure is needed, one that would consider the problem of meter inference and tracking more holistically. Such a model would also be adaptable to different metrical structures and handle variations in real world scenarios. The tracking algorithm based on dynamic programming is also ad hoc and loosely uses the tightly coupled information between the tempo, akṣaras and the sama.

Considering these insights and limitations, we explore Bayesian models for meter inference, which provide an effective probabilistic framework for the task, with several useful inference algorithms and well studied formulations that can be utilized to our benefit.

5.3 Bayesian models for meter analysis

Recently, Bayesian models have been applied successfully to meter analysis tasks. The effectiveness of such models stem from their ability to accurately model metrical structures and their adaptability to different metrical structures, music styles and variations. These advantages are supplemented by the huge literature on Bayesian models and efficient exact and approximate inference algorithms. Since metrical structures are mostly mental constructs, the use of such generative graphical probabilistic models can even perhaps be hypothesized that they closely (better than other approaches to meter analysis) emulate the mechanisms of progression through metrical cycles.

With a fundamental dependence on time, any model that aims to accurately represent metrical structures should work on sequential data from audio features, and must be able to incorporate several different variables within one probabilistic framework. A Dynamic Bayesian network (DBN) (Murphy, 2002) is well suited in such cases, since it relates variables over time through conditional (in)dependence relations. A DBN is a generalization of the traditional linear state-space models such as Kalman filters and stochastic models such as the Hidden Markov model (HMM) and provide a general probabilistic representation and inference schemes for arbitrary non-linear and non-normal time-dependent processes.

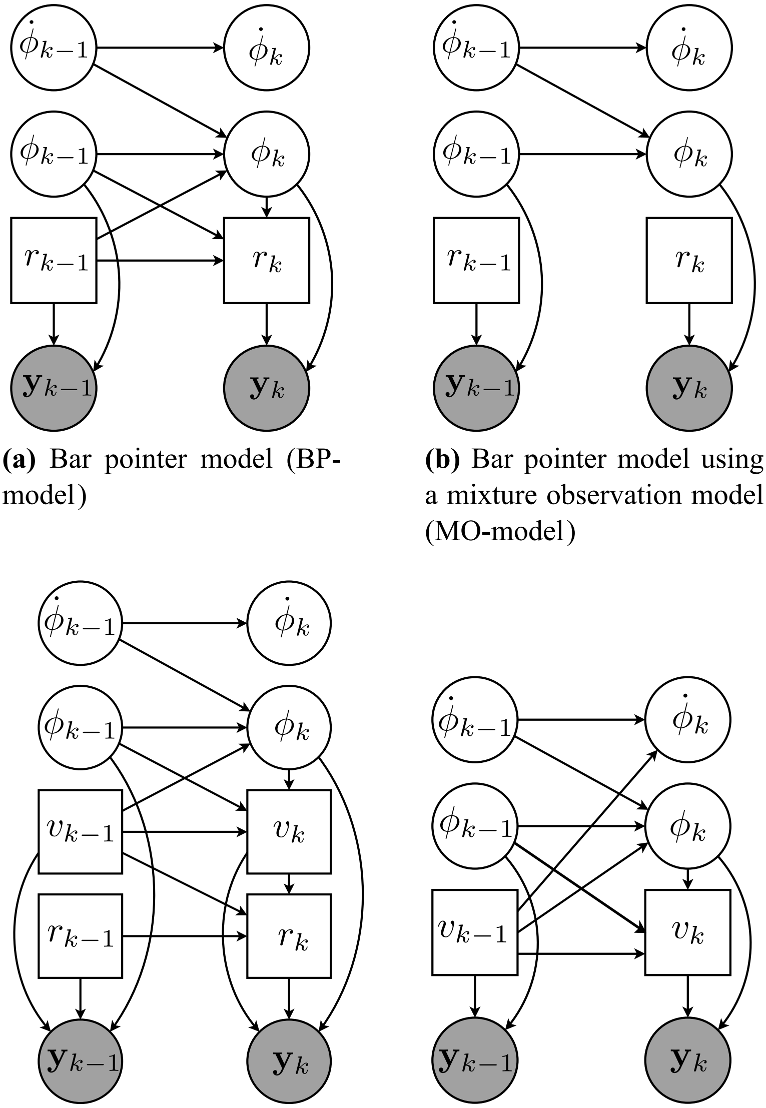
The bar pointer model is one such DBN model that has been successfully applied to meter analysis. Proposed by Whiteley, Cemgil, and Godsill (2006), it has been improved since then and applied to various meter analysis tasks over different music styles (Whiteley, Cemgil, & Godsill, 2007; Krebs et al., 2013; Krebs, Holzapfel, Cemgil, & Widmer, 2015; Böck, Krebs, & Widmer, 2014; Holzapfel, Krebs, & Srinivasamurthy, 2014; Krebs, Böck, & Widmer, 2015; Srinivasamurthy, Holzapfel, Cemgil, & Serra, 2015, 2016). In the thesis, we start with the bar pointer model and present several extensions and explore different inference schemes for those extensions, all in the context of Indian art music. The performance of such models and inference schemes are evaluated on the Carnatic and Hindustani music test datasets presented in Chapter 4, with additional evaluations to test for generalization and to baseline performance on the Ballroom dataset. The primary focus of the thesis is on the most relevant task of meter tracking, while meter inference, informed meter tracking tasks being addressed to a limited extent.

The remainder of the chapter is organized as follows. The bar pointer model is first described, explaining its model structure and inference schemes (Section 5.3.1). Extensions and enhancements to the model structure are then proposed and described in Section 5.3.2:

1. A simplified bar pointer model with a mixture observation model
2. The section pointer model

Extensions and enhancements to inference schemes on the bar pointer model extensions are then proposed and described in Section 5.3.3:

1. End of bar rhythm pattern sampling
2. Hop inference for fast meter tracking



(a) Bar pointer model (BP-model)

(b) Bar pointer model using a mixture observation model (MO-model)

(c) Section pointer model (SP-model)

(d) Simplified section pointer model

Figure 5.4: The meter analysis models used in the dissertation. In each of these DBNs, circles and squares denote continuous and discrete variables, respectively. Grey nodes and white nodes represent observed and latent variables, respectively.

5.3.1 The bar pointer model

The bar pointer model (referred to in short as BP-model) is a generative model that has been successfully applied for meter analysis tasks. The model assumes a hypothetical time pointer within a bar, progressing at the speed of the tempo traversing through the bar and reinitializing at the end of the bar to track the next bar. The model also assumes that specific bar length rhythm patterns are played in a bar depending on the rhythmic style, and uses these patterns to track the progression through the bar. These rhythmic patterns can be fixed *a priori* or learned from data to build an observation model for each position in the bar. When learned from data, the rhythmic patterns are built using a rhythm descriptor derived from audio, most often from frame level audio features to preserve the temporal information in features. Progressing through the bar, the model can hence be used to sample the observation model and generate a rhythmic pattern that is possible with the rhythm style. It allows for different metrical structures, tempi ranges, and rhythm styles, providing a flexible framework for meter analysis. Though applied only for meter analysis from audio recordings in this dissertation, the BP-model can be applied even to symbolic music. BP-model can be represented as a DBN with specific conditional dependence relations between the variables that lead to several variants and extensions of the model. The structure of the BP-model is shown in Figure 5.4a.

In a DBN, an observed sequence of features derived from an audio signal $\mathbf{y}_{1:K} = \{\mathbf{y}_1, \ldots, \mathbf{y}_K\}$ is generated by a sequence of hidden (latent) variables $\mathbf{x}_{1:K} = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$, where K is the length of the feature sequence (number of audio frames in an audio excerpt). The joint probability distribution of hidden and observed variables factorizes as,

$$P(\mathbf{y}_{1:K}, \mathbf{x}_{0:K}) = P(\mathbf{x}_0) \cdot \prod_{k=1}^{K} P(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, P(\mathbf{y}_k \mid \mathbf{x}_k) \qquad (5.3)$$

where, $P(\mathbf{x}_0)$ is the initial state distribution, $P(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition model, and $P(\mathbf{y}_k|\mathbf{x}_k)$ is the observation model.

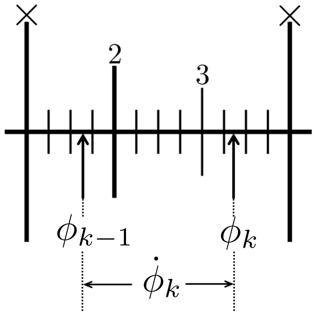


Figure 5.5: An illustration of the progression of bar position and instantaneous tempo variables over two consecutive audio frames in a cycle of rūpaka tāḷa. The effect of instantaneous tempo is greatly exaggerated for clarity in the illustration.

Hidden variables

In the bar pointer model, at each audio frame k, the hidden variable vector $\mathbf{x}_k$ describes the state of a hypothetical bar pointer $\mathbf{x}_k = [\phi_k \; \dot{\phi}_k \; r_k]$, representing the bar position, instantaneous tempo and a rhythmic pattern indicator, respectively (see Figure 5.5 for an illustration).

- *Rhythmic pattern indicator*: The rhythmic pattern variable $r \in \{1, \ldots, R\}$ is an indicator variable to select one of the R observation models corresponding to each bar (cycle) length rhythmic pattern of a rhythm class. Each pattern $r$has an associated length of cycle M_r and number of beat (or mātrā) pulses B_r. In the scope of this dissertation, all rhythmic patterns are learned from training data and not fixed a priori. We can infer the rhythm class or meter type (tāḷa) by allowing rhythmic patterns of different lengths from different rhythm classes to be present in the model, as used by Krebs, Holzapfel, et al. (2015). However, it is to be noted that for the problem of meter tracking, we assume that the cycle length is known and that all the R rhythmic patterns belong to the same rhythm class (tāḷa), $M_r = M$ and $B_r = B \, \forall \, r$.

- *Bar position*: The bar position $\phi \in [0, M_r)$ variable indicates a position in the bar at any audio frame and tracks the progression through the bar. Here, M_r is the length of the bar (cycle), which is also the length of the bar length rhythmic pattern being tracked. The bar position variable traverses the whole bar and wraps around to zero at the end of the bar to track the next bar. The maximum value of bar (cycle) length, M, depends on the longest bar (cycle) that is tracked. We set the length of the longest bar being tracked to a fixed value, and scale other bar (cycle) lengths accordingly.

- *Instantaneous tempo*: Instantaneous tempo $\dot{\phi}$ (measured in positions per time frame) is the rate at which the bar position variable progresses through the cycle at each time frame, measured in bar positions per time frame. The range of the variable $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ depends on the length of the cycle M and the hop size ($h = 0.02$ second used in this thesis), and can be preset or learned from data. A tempo value of $\dot{\phi}_k$ corresponds to a bar (cycle) length of $(h \cdot M_r/\dot{\phi}_k)$ seconds and $(60 \cdot {}^{B\cdot\dot{\phi}_k}/_{(M\cdot h)})$ beats (mātrās) per minute. The range of the variable can be used to restrict the range of tempi that is allowed within each rhythm class.

Initial state distribution

The initial state distribution $P(\mathbf{x}_0)$ can be used to incorporate prior information about the metrical structure of the music into the model. Different initializations are explored depending on the meter analysis task under conisderation.

Transition model

Given the the conditional dependence relations between the variables of the BP-model in Figure 5.4a, the transition model factorizes as,

$$\begin{aligned} P(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) P(\dot{\phi}_k \mid \dot{\phi}_{k-1}) \\ P(r_k \mid r_{k-1}, \phi_k, \phi_{k-1}) \end{aligned} \tag{5.4}$$

The individual terms of the equation can be expanded as,

$$P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) = \mathbb{1}_{\phi} \tag{5.5}$$

where $\mathbb{1}_{\phi}$ is an indicator function that takes a value of one if $\phi_k = (\phi_{k-1} + \dot{\phi}_{k-1}) \bmod(M_{r_k})$ and zero otherwise. The tempo transition is given by,

$$P(\dot{\phi}_k \mid \dot{\phi}_{k-1}) \propto \mathcal{N}(\dot{\phi}_{k-1}, \sigma^2_{\dot{\phi}}) \times \mathbb{1}_{\dot{\phi}} \tag{5.6}$$

where $\mathbb{1}_{\dot{\phi}}$ is an indicator function that equals one if $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ and zero otherwise, restricting the tempo to be between a predefined range. $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. The value of $\sigma_{\dot{\phi}}$ depends on the value of tempo and the length of the pattern. We set $\sigma_{\dot{\phi}} = \sigma_n \cdot \dot{\phi}_{k-1} \cdot ({}^{M_{r_{k-1}}}/_{M})$, where σ_n is a user parameter that controls the amount of local tempo variations we allow in the music piece.

$$P(r_k \mid r_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbb{A}(r_{k-1}, r_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_r & \text{else} \end{cases} \tag{5.7}$$

where, $\mathbb{A}$ is the $R \times R$ time-homogeneous transition matrix with $\mathbb{A}(i, j)$ being the transition probability from r_i to r_j, and $\mathbb{1}_r$ is an indicator function that equals one when $r_k = r_{k-1}$ and zero otherwise. Since the rhythmic patterns are one bar (cycle) in length, pattern transitions are allowed only at the end of the bar (cycle). The pattern transition probabilities are learned from data.

Observation Model

The observation model aims to model the underlying rhythmic patterns present in the metrical structure being inferred/tracked, explaining the possible rhythmic events at each position in the bar. Some of the positions in a bar have a higher probability of an onset occurring than other parts (the positions corresponding to downbeats, beats, e.g.). Further, the strength of these onsets also vary depending on accent patterns of a rhythm class (which can be modeled from labeled data). The observation model used in this dissertation aims to address both these aspects (the locations and strengths of the rhythmic events), and closely follows the observation model proposed by Krebs et al. (2013).

The utility of spectral flux based rhythmic audio features was outlined in preliminary experiments Section 5.2. A similar audio derived spectral flux feature is used in this dissertation as well, identical to features used by Krebs et al. (2013), as explained in (see Figure 4.7). Since the bass onsets have significant information about the rhythmic patterns, the features are computed in two frequency bands (Low: $\leq$ 250 Hz, High: $>$ 250 Hz).

It is assumed that the audio features depend only on the bar position and rhythmic pattern variables, without any influence from tempo. While this assumption is not completely true, it simplifies the observation model and helps to train better models with limited training data. Further, it is assumed that the audio features do not vary too much over short changes in position in cycle (e.g. the spectral flux variations within a small fraction of an akṣara might be negligible), which additionally helps to tie several positions to have the same observation probability and helps train models with limited training data.

Using beat and downbeat annotated training data, the audio features are then grouped into bar length patterns, and a k-means clustering algorithm clusters and assigns each bar of the dataset (represented by a point in a 128-dimensional space) to one of the R rhythmic patterns. The bar is then discretized into 64th note cells (or four cells per akṣara for Carnatic music, and XX for Hindustani music, corresponds to 25 bar positions with $M = 1600$), and all the features within the cell are collected for each pattern, and maximum likelihood estimates of the parameters of a two component Gaussian mixture model (GMM) are obtained. The observation probability within a 64th note cell is assumed to be constant, and computed as,

$$P(\mathbf{y} \mid \mathbf{x}) = P(\mathbf{y} \mid \phi, r) = \sum_{i=1}^{2} \pi_{\phi,r,i} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\phi,r,i}, \boldsymbol{\Sigma}_{\phi,r,i}) \qquad (5.8)$$

where, $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution and for the mixture component i, $\pi_{\phi,r,i}$, $\boldsymbol{\mu}_{\phi,r,i}$ and $\boldsymbol{\Sigma}_{\phi,r,i}$ are the component weight, mean (2-dimensional) and the covariance matrix (2×2), respectively.

Inference in bar pointer model

The goal of inference in meter analysis tasks is to find a hidden variable sequence that maximizes the posterior probability of the hidden states given an observed sequence of features: a maximum *a posteriori* (MAP) sequence $\mathbf{x}^*_{1:K}$ that maximizes $P(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K})$. The inferred hidden variable sequence $\mathbf{x}^*_{1:K}$ can then be translated into a sequence downbeat (sama) instants ($\phi^*_k = 0$), beat instants ($\phi^*_k = i \cdot {}^{M_r}/{}_{B_r}, i = 1, \ldots, B_r$), the local instantaneous tempo ($\dot{\phi}^*_k$), and the sequence of estimated rhythmic patterns r^*

Two different inference schemes are now described, an exact inference using the Viterbi algorithm in a discretized state space, and an approximate inference using particle filters in the continuous space of ϕ and $\dot{\phi}$.

Viterbi algorithm

The continuous variables of bar position and tempo can be discretized, which transforms the DBN into an HMM over the cartesian product space of the discretized variables. In the HMM, an exact inference can be performed using the Viterbi algorithm to compute the most likely sequence of hidden states given the observed data.

We follow the discretization that closely follows the method proposed by Krebs, Holzapfel, et al. (2015), by replacing the continuous variables ϕ and $\dot{\phi}$ by their discretized counterparts m and n, respectively, as

$$\begin{aligned} m &\in \{1, 2, \ldots, \lceil M_r \rceil\} && (5.9) \\ n &\in \{n_{\text{min}}, n_{\text{min}}+1, n_{\text{min}}+2, \cdots, N-1, N\} && (5.10) \end{aligned}$$

Here, $n_{\text{min}} = \lfloor \dot{\phi}_{\text{min}} \rfloor$ and $N = n_{\text{max}} = \lceil \dot{\phi}_{\text{max}} \rceil$ is the discrete minimum and maximum tempo values allowed, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote floor and ceil operations, repsectively.

With such a discretization in place, the transition model equations Eq. 5.4, Eq. 5.5 and Eq. 5.7 remain as defined. However, the tempo transition probability is redefined within the allowed tempo range as,

$$P(n_k \mid n_{k-1}) = \begin{cases} 1 - p_n & \text{if } n_k = n_{k-1} \\ \frac{p_n}{2} & \text{if } n_k = n_{k-1} \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

where p_n is the probability of tempo change. It is to be noted that that the discretization of ϕ and $\dot{\phi}$ need not be done on an integer or on a uniform grid. It is possible that the tempo range can be non-uniformly sampled, as was proposed by Krebs, Böck, and Widmer (2015). In this dissertation, however, only a uniform discretization is explored in the context of the HMM. Viterbi algorithm (Rabiner, 1989) is then used to obtain a MAP sequence of states with the HMM. The HMM based exact inference in bar pointer model as described in the section will be denoted as $\texttt{HMM}_0$in the dissertation.

The drawback of this approach is that the discretization has to be on a very fine grid in order to guarantee good performance, which leads to a prohibitively large state space and, as a consequence, to a computationally demanding inference. The size of the state space is $\mathfrak{S} = M \cdot N \cdot R$ and needs an $\mathfrak{S} \times \mathfrak{S}$ sized transition matrix. As an example, dividing a bar into $M = 1600$ position states, with $N = 15$ tempo states and $R = 4$ patterns, the size of the state space is $\mathfrak{S} = 96000$ states. The computational complexity of the Viterbi algorithm is $O(K \cdot |\mathfrak{S}|^2)$. Even though the state transition matrix is sparse due to lesser number of allowed transitions leading to a complexity of $O(K \cdot M \cdot R)$, the inference with HMM can become computationally prohibitive and does not scale well with increasing number of states. This problem can be overcome, for instance, by using approximate inference methods such as particle filters.

Particle Filter (PF)

Particle filters (or Sequential Monte Carlo methods) are a class of approximate inference algorithms to estimate the posterior density of a state space. They overcome two main problems of the HMM: discretization of the state space and the quadratic scaling up of the size of state space with additional hidden variables. In addition, they can incorporate long term relationships between hidden variables.

In the continuous state space of $\mathbf{x}_{1:K}$, the exact computation of the posterior $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ is often intractable, but it can be evaluated pointwise. In particle filters, the posterior is approximated using a weighted set of points (known as particles) in the state space as,

$$P(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K}) \approx \sum_{i=1}^{N_p} w_K^{(i)} \delta(\mathbf{x}_{1:K} - \mathbf{x}_{1:K}^{(i)}) \tag{5.12}$$

Here, $\{\mathbf{x}_{1:K}^{(i)}\}$ is a set of points (particles) with associated weights $\{w_K^{(i)}\}$, $i = 1, \ldots, N_p$, and $\mathbf{x}_{1:K}$ is the set of all state trajectories until frame K, while $\delta(x)$ is the Dirac delta function. N_p is the number of particles.

Starting with $P(\mathbf{x}_0)$, to approximate the posterior pointwise, we need a suitable method to draw samples $\mathbf{x}_k^{(i)}$ and compute appropriate weights $w_k^{(i)}$ recursively at each time step. It is further nontrivial to sample from an arbitrary posterior distribution. A simple approach is Sequential Importance Sampling (SIS) (Doucet & Johansen, 2009), where we sample from a *proposal* distribution $Q(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ that has the same support and is as similar to the true (target) distribution $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ as possible. To account for the fact that we sampled from a proposal and not the target, we attach an importance weight $w_K^{(i)}$ to each particle, computed as,

$$w_K^{(i)} = \frac{P(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K})}{Q(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K})} \tag{5.13}$$

With a suitable proposal density, these weights can be computed recursively as,

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{P(\mathbf{y}_k \mid \mathbf{x}_k^{(i)})P(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{Q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \tag{5.14}$$

Following Krebs, Holzapfel, et al. (2015), we choose to sample from the transition probability $Q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k) = P(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})$, which reduces Eq. 5.14 to

$$w_k^{(i)} \propto w_{k-1}^{(i)} P(\mathbf{y}_k \mid \mathbf{x}_k^{(i)}) \tag{5.15}$$

The SIS algorithm derives samples by first sampling from proposal, in this case the transition probability and then computes weights according to Eq. 5.15. Once we determine the particle trajectories $\{\mathbf{x}_{1:K}^{(i)}\}$, we then select the trajectory $\mathbf{x}_{1:K}^{(i^*)}$ with the highest weight $w_K^{(i^*)}$ as the MAP state sequence.

Many extensions have been proposed to the basic SIS filter (Doucet and Johansen (2009) provide a comprehensive overview) to address several problems with it. Some of the relevant extensions are briefly mentioned, emphasizing their key aspects. A more detailed description of the algorithms has been presented by Krebs, Holzapfel, et al. (2015). The most challenging problem in particle filtering is the degeneracy problem, where within a short time, most of the particles have a weight close to zero, representing unlikely regions of state space. This is contrary to the ideal case when we want the proposal to match well with the target distribution leading to a uniform weight distribution with low variance. To reduce the variance of the particle weights, resampling steps are necessary, which replaces low weight particles with higher weight particles by selecting particles with a probability proportional to their weights. Several resampling methods have been proposed, but we use systematic resampling in this dissertation as recommended by Doucet and Johansen (2009). With resampling as the essential difference, the SIS filter with resampling is called as Sequential Importance Sampling/Resampling (SISR) filter.

In meter analysis problems, due to metrical ambiguities, the posterior distribution $P(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ is highly multimodal. Resampling tends to lead to a concentration of particles in one mode of the posterior, while the remaining modes are not covered. One way to alleviate this problem is to compress the weights $\mathbf{w}_k = w_k^{(i)}$, $i = 1, \ldots, N_p$ by a monotonically increasing function to increase the weights of particles in low probability regions so that they can survive resampling. After resampling, the weights have to be uncompressed to give a valid probability distribution. This can be formulated as an Auxiliary Particle Filter (APF) (Johansen & Doucet, 2008).

A particle system that is capable of handling metrical ambiguities must maintain the multimodality of posterior distribution and be able to track several hypotheses together, which SISR and APF cannot do explicitly. A system called the Mixture Particle Filter (MPF) was proposed by Vermaak, Doucet, and Pérez (2003) to track multiple hypotheses, and was adapted to meter inference by Krebs, Holzapfel, et al. (2015).

In a MPF, each particle is assigned to a cluster that (ideally) represents a mode of the posterior. During resampling, the particles of a cluster interact only with particles of the same cluster. Resampling is done independently in each cluster, while maintaining the probability distribution intact. This way, all the modes of the posterior can be tracked through the whole audio piece, and the best hypothesis can be chosen at the end. In this work, we use an identical clustering scheme using a cyclic distance measure as described by Krebs, Holzapfel, et al. (2015) to track several different possible metrical positions at a given time. We use a cyclic distance measure that can take into account the cyclic nature of the bar position ϕ. By representing the bar position as a complex phasor on the unit circle, we can compute the corresponding angle $\varphi(\phi_k) = {}^{2\pi\phi_k}/_{M}$. A distance between two particles indexed by i and j can then be computed as,

$$d(i,j) = \lambda_\phi \left[\left(\cos(\varphi^{(i)}) - \cos(\varphi^{(j)})\right)^2 + \left(\sin(\varphi^{(i)}) - \sin(\varphi^{(j)})\right)^2 \right] + \lambda_{\dot{\phi}} \left(\dot{\phi}^{(i)} - \dot{\phi}^{(j)} \right)^2 + \lambda_r (r^{(i)} - r^{(j)})^2 \quad (5.16)$$

where, the parameters $[\lambda_\phi,\ \lambda_{\dot{\phi}},\ \lambda_r]$ control the relative weights in the distance.

In the MPF, after an initial cluster assignment, we perform a reclustering before every resampling step, merging or splitting clusters based on the average distance between cluster centroids. The clustering, merging and splitting of clusters is necessary to control the number of clusters, which ideally represents the number of modes in the posterior. The mixture particle filter can be combined with the Auxiliary resampling to give the Auxiliary Mixture Particle Filter (AMPF). As recommended by Krebs, Holzapfel, et al. (2015), we resample at a fixed interval T_s.

It has been clearly shown by Krebs, Holzapfel, et al. that AMPF can be effectively used for the task of meter inference and tracking. In this dissertation, the AMPF algorithm, as outlined in Algorithm 1 is used for all meter analysis tasks that need approaximate inference. The AMPF algorithm with the bar pointer model as described in this section will be denoted as $\mathtt{AMPF}_0$in the dissertation.

The complexity of the PF schemes scale linearly with N_p irrespective of the size of state space, leading to an efficient inference in large state spaces. Further, compared to the HMM using Viterbi decoding that has a space complexity of $O(K \cdot |\mathfrak{S}|)$, the PF needs to store just N_p state trajectories and weights, significantly reducing the memory requirements. An additional advantage is that the number of particles can be chosen based on the computational power we can afford, and we can make the state space larger with no or only a marginal increase in the computational requirements.

To conclude, the bar pointer model is a state of the art model useful in all the meter analysis that are addressed in the thesis. The performance of meter analysis with bar pointer model will be a baseline for all the datasets and music cultures under study. Though a state of the art model explored before, the dissertation presents a further exploration of the model with the following novelties compared to the state of the art:

Algorithm 1 An outline of the AMPF$_0$ algorithm (Inference in BP-model using AMPF

```
for i = 1 to $N_p$ do
    Sample $\mathbf{x}_0^{(i)} \sim P(\mathbf{x}_0)$    ▷ $\mathbf{x}_k = [\phi_k, \dot{\phi}_k, r_k]$
    Set $w_0^{(i)} = {}^1/_{N_p}$
Cluster $\{\mathbf{x}_0^{(i)} | i = 1, 2, \cdots, N_p\}$, get cluster assignments $\{c_0^{(i)}\}$
for k = 1 to $K$ do
    for i = 1 to $N_p$ do    ▷ $\phi$, $r$: Proposal and weights
        Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
        if $\phi_k^{(i)} < \phi_{k-1}^{(i)}$ then    ▷ Bar crossed
            $r_k^{(i)} \sim P(r_k^{(i)} \mid r_{k-1}^{(i)})$    ▷ Sample patterns
        else
            $r_k^{(i)} = r_{k-1}^{(i)}$
        $\tilde{w}_k^{(i)} = w_k^{(i)} \cdot P(\mathbf{y}_k \mid \phi_k^{(i)}, r_k^{(i)})$
    for i = 1 to $N_p$ do    ▷ Normalize weights
        $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N_p} \tilde{w}_k^{(i)}}$
    if mod $(k, T_s) = 0$ then    ▷ Cluster, resample, reassign
        Cluster and resample $\{\mathbf{x}_k^{(i)}, w_k^{(i)}, c_k^{(i)} | i = 1, 2, \cdots, N_p\}$
        to obtain $\{\hat{\mathbf{x}}_k^{(i)}, \hat{w}_k^{(i)} = {}^1/_{N_p}, \hat{c}_k^{(i)}\}$
        for i = 1 to $N_p$ do
            $\mathbf{x}_k^{(i)} = \hat{\mathbf{x}}_k^{(i)}$, $w_k^{(i)} = \hat{w}_k^{(i)}$, $c_k^{(i)} = \hat{c}_k^{(i)}$
    Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)} \mid \dot{\phi}_{k-1}^{(i)})$    ▷ Sample tempo
Compute $\mathbf{x}_{1:K}^* = \mathbf{x}_{1:K}^{(i^*)} \mid i^* = \text{argmax}_i\, w_K^{(i)}$    ▷ MAP sequence
```

1. The bar pointer model has been extended and evaluated on Indian art music, showing its utility and discussing its limitations with the kinds of metrical structures that occur in Indian music. These learnings and insights will help improve the components of the model, pushing the state of the art ahead.
2. Even though the bar pointer model can handle multiple rhythmic patterns per rhythm class (or meter type), no previous study has applied it to include more than one rhythmic pattern per rhythm class. The dissertation for the first time applies the bar pointer model to multiple rhythm patterns per rhythm class and presents

a comprehensive evaluation.

3. Several novel extensions to the bar pointer model are explored and presented in the dissertation to address several shortcomings of the model, and to extend the functionality of the model.

Several extensions and enhancements to the bar pointer model can be proposed. For better organization, these extensions are grouped into two categories: model extensions that explore changes to the model structure of the bar pointer model, either by adding additional hidden variables or using different conditional independence relationships, and inference extensions that explore different inference schemes in the bar pointer model, for better and faster inference.

5.3.2 Model extensions

The model extensions proposed to the bar pointer model improve upon the model structure. Two different model extensions are proposed in the dissertation: a mixture observation model, and the section pointer model.

Bar pointer model with a mixture observation model (MO-model)

We propose a simplification to the bar pointer model that uses a diverse mixture observation model incorporating observations from multiple rhythmic patterns. The bar pointer model uses multiple rhythmic patterns for meter analysis. When the task is only to track the beats and downbeats in meter tracking (assuming the meter type is known *a priori*), tracking pattern transitions is superfluous. However, to capture the diversity of patterns, a diverse mixture observation model can be used to incorporate observations from multiple rhythmic patterns. Since all the rhythmic patterns belong to the same type of meter, we can simplify BP-model to track only the ϕ and $\dot{\phi}$ variables while using an observation model that computes the likelihood of an observation by marginalizing over all the patterns. The motivation for this simplification is two-fold: the inference is simplified with only two hidden variables, and we can increase the influence of diverse patterns that occur throughout a metrical cycle in the inference. This simplication of the BP-model that uses a mixture observation model is referred to as MO-model and is shown in Figure 5.4b.

With this simplification in the model structure in Figure 5.4b, the transition model in Eq. 5.4 now changes to,

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = P(\boldsymbol{\beta}_k \mid \boldsymbol{\beta}_{k-1}) = P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}) P(\dot{\phi}_k \mid \dot{\phi}_{k-1}) \tag{5.17}$$

Here, $\boldsymbol{\beta} = [\phi, \dot{\phi}]$ is defined as the subset of the hidden variables tracked using the MO-model . The tempo transition term of the above equation remains identical to the BP-model , as in Eq. 5.6. The term for ϕ also remains similar to Eq. 5.5 in the BP-model , apart from the removal of the dependence on r_{k-1} as,

$$P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}) = \mathbb{1}_{\phi} \tag{5.18}$$

where $\mathbb{1}_{\phi}$ is an indicator function that takes a value of one if $\phi_k = (\phi_{k-1} + \dot{\phi}_{k-1}) \bmod (M)$ and zero otherwise, noting that the length of all rhythmic patterns are equal, $M_r = M$, for all values of r.

The observation model aims to utilize information from multiple rhythmic patterns. The MO-model uses a mixture observation model computed from Eq. 5.8 by marginalizing over the patterns, assuming equal priors.

$$P(\mathbf{y} \mid \boldsymbol{\beta}) \propto \sum_{j=1}^{R} P(\mathbf{y} \mid \phi, r = j) \tag{5.19}$$

This observation model makes the MO-model simpler, while giving a computational advantage. Since the observation likehood can be precomputed, inference with MO-model requires much lower computational resources, with only a marginal increase in cost during inference with increase in number of patterns.

Inference in MO-model

The inference in MO-model is similar to that using BP-model , by discretizing the state space to lead to an HMM and applying Viterbi algorithm, or using particle filters. The inference in HMM can be performed with pre-computed likelihood from different rhythmic patterns from the mixture observation model, denoted to as HMM$_{\mathrm{m}}$ in this dissertation. Similarly, the AMPF with the mixture observation model extension is outlined in Algorithm 2 and is denoted as AMPF$_{\mathrm{m}}$ in the rest of the chapter.

Algorithm 2 Outline of the $\texttt{AMPF}_{\mathrm{m}}$ algorithm (AMPF for inference in the simplified bar pointer model with a mixture observation model: MO-model)

```
for i = 1 to $N_p$ do
    Sample $\boldsymbol{\beta}_0^{(i)} \sim P(\phi_0)P(\dot{\phi}_0)$, $w_0^{(i)} = {}^1\!/\!{}_{N_p}$    ▷ $\boldsymbol{\beta}_k = [\phi_k, \dot{\phi}_k]$
Cluster $\{\boldsymbol{\beta}_0^{(i)} | i = 1, 2, \cdots, N_p\}$, get cluster assignments $\{c_0^{(i)}\}$
for k = 1 to $K$ do
    for i = 1 to $N_p$ do    ▷ $\phi$: Proposal and weights
        Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)} \mid \boldsymbol{\beta}_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
        $\tilde{w}_k^{(i)} = w_k^{(i)} \times \sum_{j=1}^{R} P(\mathbf{y}_k \mid \phi_k^{(i)}, r = j)$
    for i = 1 to $N_p$ do    ▷ Normalize weights
        $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N_p} \tilde{w}_k^{(i)}}$
    if  mod $(k, T_s) = 0$ then    ▷ Cluster, resample, reassign
        Cluster and resample $\{\boldsymbol{\beta}_k^{(i)}, w_k^{(i)}, c_k^{(i)} | i = 1, 2, \cdots, N_p\}$
         to obtain $\{\hat{\boldsymbol{\beta}}_k^{(i)}, \hat{w}_k^{(i)} = {}^1\!/\!{}_{N_p}, \hat{c}_k^{(i)}\}$
        for i = 1 to $N_p$ do
            $\boldsymbol{\beta}_k^{(i)} = \hat{\boldsymbol{\beta}}_k^{(i)}$, $w_k^{(i)} = \hat{w}_k^{(i)}$, $c_k^{(i)} = \hat{c}_k^{(i)}$
    Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)} \mid \dot{\phi}_{k-1}^{(i)})$
Compute $\boldsymbol{\beta}_{1:K}^{*} = \boldsymbol{\beta}_{1:K}^{(i^*)} \mid i^* = \mathrm{argmax}_i\, w_K^{(i)}$    ▷ MAP sequence
```

Section pointer model

To the best of our knowledge, the methods for meter tracking and inference so far, including the bar pointer model, have been applied and evaluated on metrical cycles of short durations. E.g., the typical duration of a 4/4 measure in popular Eurogenetic music would last from a bit less than 2s to little more than 4s. Longer metrical cycles were reported to cause problems in existing approaches (Holzapfel et al., 2014). Interestingly, this upper duration coincides with the limit of a perceptual phenomenon referred to as *perceptual present* (Clarke, 1999), and it has been argued that longer metrical cycles might not be perceived as a single rhythmic entity (Clayton, 2000). In tracking such long metrical cycles, listeners often track shorter, but musically meaningful sections of the cycle. This motivates the use of sub-bar or sub-cycle length rhythmic patterns in meter analysis tasks. Compared to longer cycle length patterns, shorter pattern have lower variability and hence might provide better cues for meter tracking.

A similar idea was applied by Böck et al. (2014), where rhythmic patterns of beat length are learned in order to perform beat tracking. However, Böck et al. assume beats to form an isochronous sequence - an assumption that does not hold for many musics of the world, such as Indian, Turkish, Balkan, or Korean musics. Furthermore, they do not attempt to infer higher metrical levels, i.e. downbeat positions. By proposing a generalization to the bar pointer model, we address for the first time, the two basic limitations of the existing meter tracking approaches including the bar pointer model: the restrictions to short cycles and isochronous beat sequences. The generalization of the bar pointer model, called the section pointer model (SP-model), uses musically meaningful and possibly unequal section length rhythmic patterns in the task of meter tracking. With the new model, it is further possible to evaluate if using shorter section length rhythmic patterns can improve meter tracking compared to bar (cycle) length rhythmic patterns, in the presence of long metrical cycles.

The idea behind the section pointer model is to track sections instead of the whole bar (cycle). The rhythmic patterns are now one section in length, and hence possibly unequal in length. A pointer tracks the progresion through each section, and a over-arching section identifier handles the progression through the sections of a cycle. The structure of the SP-model is shown in Figure 5.4c, and is a generalization to the bar pointer model, with the bar pointer model being a special sub-case. Hence the SP-model can be applied to arbitrary music styles in a straight forward way, just like the bar pointer model.

Both Carnatic and Hindustani music have sections within the tāḷa (a□ga and vibhāg, respectively), which are musically well defined and hence the use of section length rhythmic patterns in the task of meter analysis can be explored with meaningful cycle divisions. Hindustani music further has tāl cycles that last over a minute (Clayton, 2000) and hence is a good test case for the section pointer model. **improve!** The large tempo range and the filler strokes can provide a dense surface rhythm than what is expected from the underlying metrical structure. This surface rhythm can confuse the meter trackers and bias it towards the higher values of tempo, something that can be mitigated by tracking shorter section length patterns. Further, tracking large mātrā periods in vila□bit pieces causes an unstable local tempo estimate that leads to a drifting of the tracking algorithms, which also is expected to be mitigated by tracking shorter length patterns.

In the section pointer model, a hypothetical pointer traverses each section of a metrical cycle. Hence, in addition to the variables ϕ, $\dot{\phi}$, $r$of the bar pointer model, we now additionally introduce a section indicator variable. In reference to the SP-model , at each audio frame, we redefine and denote the hidden (latent) variable vector $\mathbf{x}_k = [\phi_k, \dot{\phi}_k, r_k, v_k]$, where:

- *Section indicator*: The section indicator variable $v \in \{1, \ldots, V\}$ is an indicator variable that identifies the section (vibhāg in Hindustani music or a□ga in Carnatic music) of a bar (tāl/a), and selects one of the V observation models corresponding to each section length rhythmic pattern learned from data. A rhythm class (tāl/a) might have many sections of different lengths. We denote the number of mātrās/beats in a section v by B_v.

- *Rhythmic pattern indicator*: For each section v, there are one or more associated rhythm patterns denoted by r. The rhythm pattern indicator r, along with the section indicator v select the appropriate observation model to be used. For convenience, we assume each section to be modeled by an equal number of patterns, with a total of R distributed across all the sections equally. Hence, the number of rhythmic patterns per section is given as, R/V patterns, with the assumption that R is an integer multiple of V.

- *Position in section*: The position variable ϕ in the SP-model tracks the position within a section as $\phi \in [0, M_v)$, where M_v is the length of section v. ϕ increases from 0 to M_v and then resets to 0 to start tracking the next section. We set the length of the longest section as M, and then scale the lengths of other sections accordingly.

- *Instantaneous tempo*: Instantaneous tempo variable $\dot{\phi}$ (measured in positions per time frame) is similar to the instantaneous tempo variable of the BP-model and denotes the rate at which the position variable ϕ progresses through a section at each time frame.

The allowed range of the variable $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ depends on the frame hop size ($h = 0.02$ second used here as before), and can be preset or learned from data. In a given section v, a value of $\dot{\phi}_k$ corresponds to a section duration of $(h \cdot M_v/\dot{\phi}_k)$ seconds and $(60 \cdot {}^{B_v \cdot \dot{\phi}_k}/_{(M_v \cdot h)})$ mātrās/beats per minute.

Given the conditional dependence relations in Figure 5.4c, the transition probability in SP-model factorizes as,

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}, v_{k-1})\, P(\dot{\phi}_k \mid \dot{\phi}_{k-1}) \\ P(v_k \mid v_{k-1}, \phi_k, \phi_{k-1})\, P(r_k \mid r_{k-1}, v_k, v_{k-1}) \quad (5.20)$$

Each of the terms in Eq. 5.20 can be expanded as,

$$P(\phi_k \mid \phi_{k-1}, \dot{\phi}_{k-1}, v_{k-1}) = \mathbb{1}_{\phi} \quad (5.21)$$

where $\mathbb{1}_{\phi}$ is an indicator function that takes a value of one if $\phi_k = (\phi_{k-1} + \dot{\phi}_{k-1})\,\text{mod}(M_{v_{k-1}})$ and zero otherwise. The tempo transition is given by,

$$P(\dot{\phi}_k \mid \dot{\phi}_{k-1}) \propto \mathcal{N}(\dot{\phi}_{k-1}, \sigma^2_{\dot{\phi}}) \times \mathbb{1}_{\dot{\phi}} \quad (5.22)$$

where $\mathbb{1}_{\dot{\phi}}$ is an indicator function that equals one if $\dot{\phi}_k \in [\dot{\phi}_{\text{min}}, \dot{\phi}_{\text{max}}]$ and zero otherwise, restricting the tempo to be between a predefined range. $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. The value of $\sigma_{\dot{\phi}}$ depends on the value of tempo and the length of the section. As before with the BP-model , we set $\sigma_{\dot{\phi}} = \sigma_n \cdot \dot{\phi}_{k-1} \cdot ({}^{M_{v_{k-1}}}/_{M})$, where σ_n is a user parameter that controls the amount of local tempo variations we allow in the music piece. The section transition probability is given by,

$$P(v_k \mid v_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbb{B}(v_{k-1}, v_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_v & \text{else} \end{cases} \quad (5.23)$$

where, $\mathbb{B}$ is the $V \times V$ time-homogeneous transition matrix with $\mathbb{B}(i, j)$ being the transition probability from v_i to v_j, and $\mathbb{1}_r$ is an indicator function that equals one when $v_k = v_{k-1}$ and zero otherwise. The pattern transitions are governed by,

$$P(r_k \mid r_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbb{A}(r_{k-1}, r_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_r & \text{else} \end{cases} \quad (5.24)$$

$$\mathbb{B} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbb{A} = \begin{bmatrix} 0 & 0 & p_1 & 1-p_1 & 0 & 0 \\ 0 & 0 & p_2 & 1-p_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_3 & 1-p_3 \\ 0 & 0 & 0 & 0 & p_4 & 1-p_4 \\ p_5 & 1-p_5 & 0 & 0 & 0 & 0 \\ p_6 & 1-p_6 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 5.6: An illustration of the form of section and rhythmic pattern transition matrices for tracking rūpak tāl with the SP-model. The patterns with index $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$ correspond to sections $1, 2$, and 3, respectively. The values p_1 to p_6 are learnt from training data.

where, $\mathbb{A}$ is the $R \times R$ time-homogeneous transition matrix with $\mathbb{A}(i, j)$ being the transition probability from r_i to r_j, and $\mathbb{1}_r$ is an indicator function that equals one when $r_k = r_{k-1}$ and zero otherwise.

Section changes are permitted only at the end of the section. Since the rhythmic patterns are also one section in length, pattern transitions are allowed only at the end of a section. The matrix $\mathbb{B}$ is used to determine the order of the sections as defined in the tāl/a by allowing only those defined transitions. Further, $\mathbb{B}$ can be set to do meter tracking by including only the section transitions of a specific tāl/a. A larger $\mathbb{B}$ including all the sections from all the rhythm classes can be used for meter inference as well. $\mathbb{A}$ closely follows $\mathbb{B}$ and has non-zero probabilities only for allowed pattern transitions. As an illustration, consider tracking rūpak tāl (which has three vibhāg $V = 3$) with the SP-model and two rhythmic patterns per section (hence, $R = 6$. The canonical forms of the section transition matrix $\mathbb{B}$ and $\mathbb{A}$ can be illustrated as in Figure 5.6.

The observation model with the SP-model is similar to that of the BP-model, with the assumption that the audio features depend on the position in section, the rhythmic pattern, and the section indicator variables. The annotated data has mātrās/beats numbered with their position in the bar (cycle) and hence they can used to extract section length rhythmic patterns from audio recordings. Section length patterns from each section are then clustered into ${}^{R}\!/_{V}$ pattern clusters using a k-means algorithm. Each section is further discretized into 64th note cells, all features within the cell are accumulated and a two component GMM is fit to each cell. The observation likelihood with the SP-model can hence be computed as,

$$P(\mathbf{y} \mid \mathbf{x}) = P(\mathbf{y} \mid \phi, r, v) = \sum_{i=1}^{2} \pi_{\phi,r,v,i} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\phi,r,v,i}, \boldsymbol{\Sigma}_{\phi,r,v,i}) \tag{5.25}$$

where, $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution and for the mixture component i, $\pi_{\phi,r,v,i}$, $\boldsymbol{\mu}_{\phi,r,v,i}$ and $\boldsymbol{\Sigma}_{\phi,r,v,i}$ are the component weight, mean (2-dimensional) and the covariance matrix (2×2), respectively. Hence, there is an observation GMM for each section, rhythmic pattern, and tied section position states.

A special case of the SP-model is when the number of sections equals the number of rhythmic patterns, $V = R$, with each section being modeled with just one rhythmic pattern. In such a case, the matrices $\mathbb{A} = \mathbb{B}$ rendering the additional r variable superfluous. In such a case, the SP-model can be simplified (Srinivasamurthy et al., 2016) to the form shown in Figure 5.4d.

Inference in SP-model

Both exact and approximate inference schemes can be used for inference in SP-model, similar to those for BP-model. The Viterbi algorithm inference on a dicretized SP-model state space is denoted as $\texttt{HMM}_\text{s}$. The AMPF inference in SP-model will be referred to in the rest of the chapter as $\texttt{AMPF}_\text{s}$ and the algorithm is outlined in Algorithm 3.

5.3.3 Inference extensions

The inference extensions proposed aim for better approximate inference with the bar pointer model, either by making it faster, or by improving approximate inference.

End-of-bar pattern sampling

The use of bar (cycle) length rhythmic patterns for meter analysis in BP-model is well motivated. When there are multiple rhythmic patterns being tracked, we can theoretically infer the rhythmic

Algorithm 3 Outline of the `AMPF`$_\text{s}$ algorithm (AMPF inference in SP-model)

```
for i = 1 to $N_p$ do
    Sample $\mathbf{x}_0^{(i)} \sim P(\mathbf{x}_0)$                    ▷ $\mathbf{x}_k = [\phi_k, \dot{\phi}_k, r_k, v_k]$
    Set $w_0^{(i)} = {}^1/_{N_p}$                                 ▷ $\boldsymbol{\alpha}_k = [\phi_k, \dot{\phi}_k, v_k]$
Cluster $\{\mathbf{x}_0^{(i)} \mid i = 1, 2, \cdots, N_p\}$, get cluster assignments $\{c_0^{(i)}\}$
for k = 1 to $K$ do
    for i = 1 to $N_p$ do                             ▷ $\phi$, $r$, $v$: Proposal and weights
        Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)} \mid \boldsymbol{\alpha}_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
        if $\phi_k^{(i)} < \phi_{k-1}^{(i)}$ then                                  ▷ Bar crossed
            $r_k^{(i)} \sim P(r_k^{(i)} \mid r_{k-1}^{(i)})$                      ▷ Sample from $\mathbb{A}$
            $v_k^{(i)} \sim P(v_k^{(i)} \mid v_{k-1}^{(i)})$                      ▷ Sample from $\mathbb{B}$
        else
            $r_k^{(i)} = r_{k-1}^{(i)}, v_k^{(i)} = v_{k-1}^{(i)}$
        $\tilde{w}_k^{(i)} = w_k^{(i)} \cdot P(\mathbf{y}_k \mid \phi_k^{(i)}, v_k^{(i)}, r_k^{(i)})$
    for i = 1 to $N_p$ do                                     ▷ Normalize weights
        $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N_p} \tilde{w}_k^{(i)}}$
    if  mod $(k, T_s) = 0$ then                  ▷ Cluster, resample, reassign
        Cluster and resample $\{\mathbf{x}_k^{(i)}, w_k^{(i)}, c_k^{(i)} | i = 1, 2, \cdots, N_p\}$
         to obtain $\{\hat{\mathbf{x}}_k^{(i)}, \hat{w}_k^{(i)} = {}^1/_{N_p}, \hat{c}_k^{(i)}\}$
        for i = 1 to $N_p$ do
            $\mathbf{x}_k^{(i)} = \hat{\mathbf{x}}_k^{(i)}, w_k^{(i)} = \hat{w}_k^{(i)}, c_k^{(i)} = \hat{c}_k^{(i)}$
    Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)} \mid \dot{\phi}_{k-1}^{(i)})$                       ▷ Sample tempo
Compute $\mathbf{x}_{1:K}^{*} = \mathbf{x}_{1:K}^{(i^*)} \mid i^* = \text{argmax}_i\, w_K^{(i)}$         ▷ MAP sequence
```

pattern that occurred in the current bar only after observing the features corresponding to the whole bar. However, in the `AMPF`$_0$ algorithm with the BP-model , at the beginning of every bar, the pattern transition matrix $\mathbb{A}$ is used to sample a pattern for the current bar. The rhythmic pattern so sampled is fixed for the whole bar, which is suboptimal. This is contrary to intuition, in which we need the whole bar to see and infer which pattern occurred, a decision that can only be made at the end of the bar, not the beginning. An extension to $\mathtt{AMPF}_0$ algorithm is proposed to address this limitation.

The extension, called end-of-bar pattern sampling extension to AMPF (called $\mathtt{AMPF}_\mathrm{e}$ in short), defers a decision of sampling the pattern in the current bar to the end of the bar. In the current bar that is being tracked, the algorithm accumulates likelihood over all the patterns being tracked, and uses this likelihood to choose the most likely pattern at the end of the bar. The particle weights are updated at the end of the bar based on such an accumulated likelihood.

The proposed enhancement can be formulated in a particle system using two different clustering steps and resampling steps. In addition to AMPF clustering based on metrical position and tempo, an additional grouping is achieved with the rhythmic patterns for each particle, each of which interact within the groups during resampling. Hence within a single system of particles, we can defer the inference of patterns till the end of a bar, as outlined in detail below.

We first start by rewriting the particle system of Eq. 5.12 as,

$$P(\mathbf{x}_{1:K} \mid \mathbf{y}_{1:K}) \approx \sum_{i=1}^{N_p} \sum_{j=1}^{R} w_K^{(i,j)} \delta(\mathbf{x}_{1:K} - \mathbf{x}_{1:K}^{(i,j)}) \tag{5.26}$$

where $\mathbf{x}_{1:K}^{(i,j)}$ are particle trajectories with weights $w_K^{(i,j)}$, both indexed by i and j. Compared to the particle system in Eq. 5.12, the additional index j is used to index the rhythmic patterns for each particle. The weights are two dimensional, one dimension denotes the subset of hidden variables $\boldsymbol{\beta} = [\phi, \dot{\phi}]$, and the other dimension stores the weights of all patterns for each value of $\boldsymbol{\beta}$. With a suitable proposal density, these weights can be computed recursively as,

$$w_k^{(i,j)} \propto w_{k-1}^{(i,j)} \frac{P(\mathbf{y}_k \mid \mathbf{x}_k^{(i,j)}) P(\mathbf{x}_k^{(i,j)} \mid \mathbf{x}_{k-1}^{(i,j)})}{Q(\mathbf{x}_k^{(i,j)} \mid \mathbf{x}_{k-1}^{(i,j)}, \mathbf{y}_k)} \tag{5.27}$$

As before, we choose to sample from the transition probability $Q(\mathbf{x}_k^{(i,j)} \mid \mathbf{x}_{k-1}^{(i,j)}, \mathbf{y}_k) = P(\mathbf{x}_k^{(i,j)} \mid \mathbf{x}_{k-1}^{(i,j)})$, which reduces weight update to,

$$w_k^{(i,j)} \propto w_{k-1}^{(i,j)} P(\mathbf{y}_k \mid \mathbf{x}_k^{(i,j)}) = w_{k-1}^{(i,j)} P(\mathbf{y}_k \mid \boldsymbol{\beta}_k^{(i)}, r_k = j) \tag{5.28}$$

Let us define the following terms:

$$\mathbf{w}_k^{(i,:)} = [w_k^{(i,1)}, w_k^{(i,2)}, \cdots, w_k^{(i,R)}] \tag{5.29}$$

$$\Omega_k^{(i)} = \sum_{j=1}^{R} w_k^{(i,j)} \tag{5.30}$$

Here, $\mathbf{w}_k^{(i,:)}$ stores the weights of a particle for each rhythmic pattern and $\Omega_k^{(i)}$ denotes the marginal of a particle trajectory over all rhythmic patterns.

The AMPF$_\mathrm{e}$ is outlined in Algorithm 4. The algorithm can be interpreted to have two groups of particles in the particle system, one grouped based on $\boldsymbol{\beta}$ and the other group based on rhythm patterns. These two groups are sampled in two different sampling steps, one every T_s with the $\boldsymbol{\beta}$, and one at the end of the bar with the rhythm pattern group of particles for a specific value of $\boldsymbol{\beta}$. After each of the two resampling steps, the weights are redistributed to maintain a valid probability distribution over the particle system. Since all rhythmic patterns at a specific value of $\boldsymbol{\beta}$ are to be resampled together, it is necessary that all patterns be of equal size, and hence the AMPF$_\mathrm{e}$ algorithm can only be used in the task of meter tracking.

Faster Inference

The MO-model presented in Section 5.3.2 simplifies the BP-model and makes inference faster. Inference in BP-model can also be made faster by utilizing the time sparsity of onsets to make inference faster, using what we propose as hop inference. The idea of hop inference is that instead of performing inference at every time frame, we do inference only at specific frames that are associated with rhythmic events such as onsets. The motivation for such a hop inference is that the onsets might just be sufficient to infer metrical structures. This makes inference faster by skipping likelihood computation and sampling steps and can speed up by inference by a factor as large as 10.

Two different hop inference algorithms extensions are proposed for AMPF with BP-model in this work:

Peak Hop Inference (AMPF$_\mathrm{p}$) : The peaks of the spectral flux feature sequence is an indicator of events such as onsets. Using a

Algorithm 4 Outline of the $\mathtt{AMPF_e}$ algorithm (AMPF inference in BP-model with end-of-bar pattern sampling)

```
for i = 1 to $N_p$ do
    Sample $\boldsymbol{\beta}_0^{(i)} \sim P(\phi_0)P(\dot{\phi}_0)$, $(r_0^{(i)}) \sim P(r_0)$    ▷ $\boldsymbol{\beta}_k = [\phi_k, \dot{\phi}_k]$
    Set $\mathbf{w}_0^{(i,:)} = 1/(N_p \cdot R)$, $\Omega_k^{(i)} = 1/N_p$, $\psi^{(i)} = 0$
Cluster $\{\boldsymbol{\beta}_0^{(i)} | i = 1, 2, \cdots, N_p\}$, get cluster assignments $\{c_0^{(i)}\}$
for k = 1 to $K$ do
    for i = 1 to $N_p$ do                                   ▷ $\phi$: Proposal and weights
        Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)} \mid \phi_{k-1}^{(i)}, \dot{\phi}_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
        if $\phi_k^{(i)} < \phi_{k-1}^{(i)}$ then                          ▷ Bar crossed
            $j^* = \text{argmax}_j(\mathbf{w}_k^{(i,:)})$; Set $r_{\psi^{(i)}:k-1}^{(i)} = j^*$, $\psi^{(i)} = k$
            for j = 1 to $R$ do
                $w_k^{(i,j)} = \mathbb{A}(j^*, j) \cdot \Omega_k^{(i)}$   ▷ Weights redistributed
        else
            $r_k^{(i)} = r_{k-1}^{(i)}$
        for j = 1 to $R$ do
            $\tilde{w}_k^{(i,j)} = w_k^{(i,j)} \cdot P(\mathbf{y}_k \mid \phi_k^{(i)}, r = j)$
    for i = 1 to $N_p$ do                                   ▷ Normalize weights
        for j = 1 to $R$ do
            $w_k^{(i,j)} = \frac{\tilde{w}_k^{(i,j)}}{\sum_{i=1}^{N_p}\sum_{j=1}^{R}\tilde{w}_k^{(i,j)}}$
    if mod $(k, T_s) = 0$ then                     ▷ Cluster, resample, reassign
        Cluster and resample $\{\boldsymbol{\beta}_k^{(i)}, \Omega_k^{(i)}, c_k^{(i)} | i = 1, 2, \cdots, N_p\}$
        to obtain $\{\hat{\boldsymbol{\beta}}_k^{(i)}, \hat{\Omega}_k^{(i)} = 1/N_p, \hat{c}_k^{(i)}\}$
        for i = 1 to $N_p$ do
            Set $\boldsymbol{\beta}_k^{(i)} = \hat{\boldsymbol{\beta}}_k^{(i)}$
            for j = 1 to $R$ do                         ▷ Weights redistributed
                $w_k^{(i,j)} = w_k^{(i,j)} \cdot \frac{\hat{\Omega}_k^{(i)}}{\Omega_k^{(i)}}$
    Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)} \mid \dot{\phi}_{k-1}^{(i)})$
$\boldsymbol{\beta}_{1:K}^* = \boldsymbol{\beta}_{1:K}^{(i^*)} \mid i^* = \text{argmax}_i\, \Omega_K^{(i)}$         ▷ MAP sequence
```

In the algorithm, $\Omega_k^{(i)} = \sum_{j=1}^{R} w_k^{(i,j)}$

peak finding algorithm, the peak frames are estimated. The particles are sampled and their weights are updated only at these peak frames. The transition model updates Eq. **??**-**??** re to be redefined accordingly. In particular, the position variable update shown in Eq. 5.5 scales the instantaneous tempo by the number of frames hopped from the previous peak in order to maintain the same tempo even with a peak hop inference. Peak hop inference can speed up inference by up to a factor of 10.

Onset gated weight update (AMPF$_g$) : Despite the advantage of a faster inference, peak hop inference can lead to sharp discontinuities in ϕ and tempo values due to large jumps in their values since they are sampled after significant number of frames. An improvement to peak hop while maintaining continuity is the gated weight update, where $\dot{\phi}$ and ϕ are updated every frame to maintain continuity, while the observation model and weights of particles are updated only at frames where there is a peak in the feature, indicating an event. The basic premise is to maintain the continuity in tracking the ϕ and $\dot{\phi}$ variables, while retaining the principle of peak hop. Gated weight update needs an observation likelihood computation only at peak frames, and hence speeds up inference.

The different meter tracking models were presented in detail in this section can be summarized in Table 5.3. We now present the experiments and results of evaluation of these models and extensions on the annotated datasets.

5.4 Experiments and results

Acronym	Model	Inference algorithm	Meter Analysis	
			Inference	Tracking
§$\texttt{HMM}_0$	BP-model†	Viterbi algorithm	✓	✓
§$\texttt{AMPF}_0$	BP-model	AMPF	✓	✓
⋆$\texttt{HMM}_\text{m}$	MO-model†	Viterbi algorithm	×	✓
⋆$\texttt{AMPF}_\text{m}$	MO-model	AMPF	×	✓
⋆$\texttt{HMM}_\text{s}$	SP-model†	Viterbi algorithm	✓	✓
⋆$\texttt{AMPF}_\text{s}$	SP-model	AMPF	✓	✓
⋆$\texttt{AMPF}_\text{e}$	BP-model	AMPF with end-of-bar pattern sampling	✓	✓
⋆$\texttt{AMPF}_\text{p}$	BP-model	Peak hop inference with AMPF	✓	✓
⋆$\texttt{AMPF}_\text{g}$	BP-model	Onset gated weight update in AMPF	✓	✓

Table 5.3: A summary of the meter analysis models and inference algorithms presented in this section. The symbol § indicates an existing state of the art algorithm while the symbol ⋆ is used to denote an algorithm proposed in this dissertation. The symbol † indicates that a discretized counterpart of the model is used. The last two columns show the applicability of the algorithm in the meter analysis tasks of meter inference and meter tracking. ✓ indicates applicable, × indicates not applicable.