Multiple Viewpoint Modeling of North Indian Classical Vocal Compositions

Ajay Srinivasamurthy, Parag Chordia

Georgia Tech Center for Music Technology,
840 McMillan St., Atlanta, USA
{ajays,ppc}@gatech.edu
http://www.gtcmt.gatech.edu

Abstract. Previous research has shown that ensembles of variable length Markov models (VLMMs), known as Multiple Viewpoint Models (MVMs), can be used to predict the continuation of Western tonal melodies, and outperform simpler, fixed-order Markov models. Here we show that this technique can be effectively applied to predicting melodic continuation in North Indian classical music, providing further evidence that MVMs are an effective means for modeling temporal structure in a wide variety of musical systems.

Keywords: Multiple viewpoint modeling, Indian Classical Music, Variable Length Markov Modeling

1 Introduction and Motivation

Melody is an important component in almost all of the world's musical traditions. In North Indian classical music (NICM), it is paramount, and is the basis of a highly sophisticated system of melodic improvisation. Previous work [25] has demonstrated that melodies can be effectively modeled using Multiple Viewpoint Models (MVMs), which are ensembles of variable-length Markov models (VLMMs). Moreover, these models have been shown to accurately reflect listeners expectations about melodic continuation [27]. Chordia [8, 6] generalized this work to tabla sequence prediction; MVMs were shown to be highly effective at modeling the temporal structure of tabla compositions, a percussive tradition based on linear sequences of timbres, suggesting the generality of MVMs for modeling discrete temporal sequences in music. The current work examines whether such models applicable to melodic compositions of NICM. Specifically, we attempt to predict the next note in a symbolically notated melody, given the past context.

2 Background and Related Work

Many aspects of music, such as melodies and chord sequences, can be represented as temporally-ordered sequences of discrete symbols. It is intuitive that how a

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

sequence proceeds will depend most on recent events. For example, melodies are more likely to proceed by small intervals, rather than large jumps [29], which means that the current note constrains the next note. This idea of local dependency can be formalized using Markov models, which are discussed further in Section 5.

A defining characteristic of music is repetition [20]. Subsequences from early in a piece, such as a brief melodic motive, are often repeated later in a piece, sometimes many times. Such repetition also often occurs across pieces; for example, many songs contain common chord sequences. These patterns, which can be short or quite long, present a challenge for a fixed-order Markov models. Although high-order Markov models can be constructed to capture such long-range dependencies, for a length N pattern with K distinct states, there are K^N possible patterns. This means that when learning on a finite set of training sequences, most long patterns will be unseen, leading to extremely sparse transition matrices. A common solution to this problem is to only store sequences that have been seen, replacing a table of counts with a tree structure, called a Prediction Suffix Tree (PST) [28], which is the basis of variable-length Markov models (VLMMs). We describe VLMMs in further detail in Sec. 5. VLMMs form the basis for many music prediction and generation systems [19, 16, 1–3, 22, 18, 17, 15, 24].

In music, there are often multiple ways of representing the musical surface. For example, a melody can be thought of in terms of chromatic pitches, or more abstractly in terms of contour, a sequence of ups and downs. In some cases, patterns may be present when looking at one representation, but not in another. By combining information from multiple viewpoints, it may be possible to capture more of the temporal structure of the sequence. This is the essential idea of MVMs. Each representation, or viewpoint, is modeled using a VLMM and the predictions of each individual model are then combined to compute an overall predictive distribution (Sec. 5). MVMs were introduced by Conklin and Witten [13, 10, 14, 12, 11, 31], and developed by others such as Pearce and Wiggins [26, 25].

3 Indian Classical Music

North Indian Classical Music (NICM) is a centuries-old tradition that is based on melodic and rhythmic improvisation, typically featuring a main melodic instrument or voice and a percussionist. It is organized around *raag*, a melodic abstraction that lies somewhere between a scale, and a fixed melody. A *raag* is most easily explained as a collection of melodic motives, and a technique for developing them. The motives are sequences of notes that are often inflected with various micro-pitch alterations and articulated with an expressive sense of timing. Longer phrases are built by joining these melodic atoms together. Because of this generative process, the musical surface, contains many repeated melodic patterns, making it a natural candidate for modeling with VLMMs.

NICM uses approximately one to two hundred *raags*, of which perhaps fifty are quite common. Although the concept of a note is somewhat different from that in Western classical, often including subtle pitch motions that are essential rather than ornamental, it is accurate to say that the notes in any given *raag* conform to one of the twelve chromatic pitches of a standard just-intoned scale. It is rare to hear sustained tone that intentionally deviates from one of the twelve chromatic pitches. A given *raag* will use between five and twelve tones.

A typical performance will feature an unmetered elaboration of the *raag* called the *alap* followed by several compositions set to a rhythmic cycle, during which the tabla provides the rhythmic framework. During the rhythmic section there is an alternation between singing the composition and its elaboration, and free improvisations within the context of the *raag*.

Although NICM is largely an oral tradition, in the 20th century there was a push to systematize and notate traditional compositions. The notation that was adopted represented melodies as sequences of discrete notes, having a certain pitch and duration. Additionally certain important ornaments, such as grace notes (*kan swara*) and turns (*khatka*) were indicated as well. Figure 1 gives an example of this notation. Just as with Western music, the notation was not meant to be complete, but to be interpreted within a performance context. In

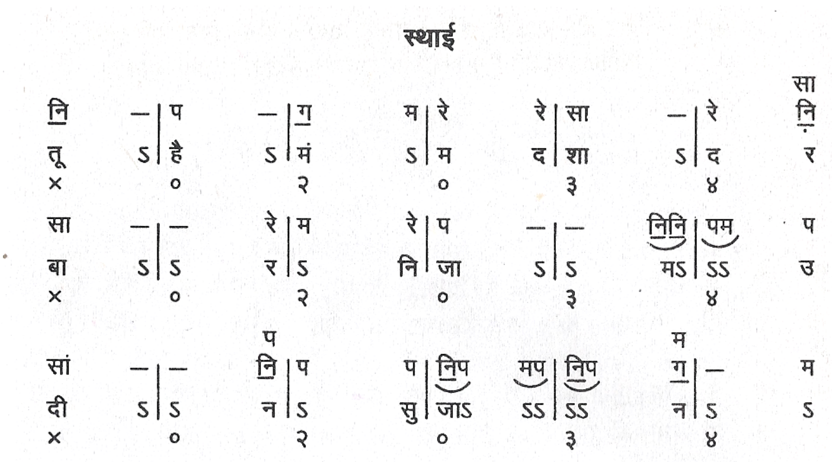

Fig. 1. An example composition of *Raag Suha* in Bhatkhande notation (Vol. II, Page 11 of [21])

this study, we decided to focus on notated compositions because of the difficulty of manual or automatic transcription from audio, which remains for future work.

4 The Indian Classical Music Database

The database used for the study is a part of the NICM symbolic database (bandishDB) being built using *Hindustani Sangeet Paddhati* by Pandit Vishnu

Narayan Bhatkhande [4] and *Abhinav Geetanjai* by Pandit Ramashray Jha [21] which are authoritative works of NICM. The database consists of *bandishes*, vocal compositions with accompanying lyrics. Each *bandish* consists of up to four sections, the *sthayee*, *antara*, *sanchari* and *abhog*. The latter two are relatively rare and are present only in a few compositions. It is worth emphasizing that NICM is largely improvised with the *bandish* providing an initial theme that is heavily elaborated according to the *raag*, within which it is set, and the artist's creativity and virtuosity.

The compositions were first manually encoded into an intermediate text-based representation. Each composition in the symbolic database was then encoded using Humdrum-based syntax called **kern, which was used to encode pitch and duration information. Additionally the following meta-data was stored for each composition: *raag* (melodic framework), *taal* (rhythmic cycle), tempo category (slow, medium, fast). Details of encoding and representation can be found in [5]. An example of the intermediate representation is shown in below.

```
id: jha2011
vol: 2
page: 11
sthayee: Tu hain mammadshah
raag: suha
taal: ektaal
tempo: madhyalaya
//  n  -/  P  -/  g  M/  R  R/  S  -/  R  >Sn,/
//  S  -/  -  R/  M  R/  P  -/  -  nn/  PM  P/
//  S'  -/  -  >Pn/  P  P/  nP  MP/  nP  >Mg/  -  M/
```

Although the artist is free to choose the actual pitch of the tonic in a rendition, all the compositions here are notated with C_4 as the middle tonic. The notes are represented using MIDI note numbers assuming a single key for all the pieces. The notes range from C_3 to B_5 but are folded into one octave from C_4-B_4. The true octave number is stored as an additional parameter. For this study, grace notes and other ornaments such as *meends*(glissandos) were ignored. These ornaments are essential for a complete experience of Indian classical music and the MIDI representation is an approximation, and it does not capture the nuances of singing in its entirety. However, for the present study, a MIDI based representation is sufficient for the analysis of symbolic music scores.

Currently the database consists of 128 compositions in *raags Bageshri, Bihag, Khamaj, Yaman, Yaman Kalyan*, totaling 12,816 notes. When completed, it is expected to be the largest machine readable symbolic NICM database. The data can be freely downloaded at `http://paragchordia.com/data.html`. Table 1 summarizes the dataset used for the experiments in this paper. For this study, compositions from *raags Yaman* and *Yaman Kalyan*, two nearly identical *raags*, are pooled together.

Table 1. Dataset

Raag	***Bandishes***	**Notes**
Yaman	44	4422
Bageshri	36	3720
Khamaj	30	2965
Bihag	18	1709
Total	**128**	**12816**

5 Predictive Modeling

The basic prediction problem can be stated as follows: given a sequence of discretely valued observations, $\{x_1, \ldots, x_{t-1}\}$, compute the next-symbol distribution $P(x_t|x_1, \ldots, x_{t-1})$. In the present case of melody prediction, given the set of symbols (note labels) S, each $x_i \in S$. Given what has occurred so far till time step $t-1$, we wish to predict the next event at time t.

Markov models can be effectively used to model these sequences of symbols, which are often referred to as strings. An n^{th} order Markov model assumes that the next state (associated with a symbol from the alphabet S) depends only on the past n states, i.e. $P(x_t|x_{t-1}, \ldots, x_1) = P(x_t|x_{t-1} \ldots x_{t-n})$. This conditional probability can be calculated by counting how often the symbol x_t follows the context $e_{t-n}^{t-1} = (x_{t-n}, \ldots, x_{t-1})$. Strings of length n are often referred to as n-grams.

Increasing order n, we can model longer strings. However, the number of possible strings ($|S|^n$) increases exponentially. So, even in large databases, most of these n-grams will never be seen, leading to the zero-frequency problem [30, 9]. Variable-length Markov models (VLMMs) address this problem by using an ensemble of fixed-order models, up to order n, to smooth probability estimates. Rather than naively storing counts for all n-grams in a table, to avoid space complexity that increases exponentially with model order, and to make it easy to search for a sequence, n-grams and counts are stored in a partial k-ary tree called a prediction suffix tree (PST) [28]. In the PST, branches represent the succession of certain symbols after others, and a node at a certain level of the PST holds a symbol from the sequence, along with information about the symbol such as the number of times it was seen in the sequence following the symbols above it, and the corresponding probability of occurrence. With this efficient representation of the VLMMs, and with a suitable smoothing method for unseen sequences, we can effectively model melodic sequences.

There are two basic approaches to smoothing: backoff and interpolated. In backoff smoothing, the probability of an unseen sequence is computed by recursively backing off to scaled versions of lower orders. The scale factor applied during each backoff serves as a penalty factor. A backoff smoothing method which adjusts the counts of unseen sequences by adding one, termed as Backoff-A (Method-A in [25]) is explored in this paper. We also explored a simple backoff approach Backoff-B, where there is no penalty for backing off. This allows the model to back off to lower orders without penalty; the model always chooses a context length for which there is a seen example sequence in the training data.

Interpolation methods compute the probability of a symbol given a context by a weighted interpolation of predicted probabilities at all the orders. A 1/N weighting scheme, as described in [8] is used. For computing probabilities at each order, each node of the PST has some probability mass reserved for unseen sequences, called the escape probability which is added through an extra escape character with a finite escape count. When an unseen sequence is seen in the test sequence at a particular order, the escape probability at that order is returned. We explored interpolated smoothing using an escape count of one at each node (termed as Interp-A) and a very low escape count of 10^{-6} at each node (termed as method Interp-B). Interp-A provides higher escape probabilities while Interp-B method assigns negligible probability mass on unseen sequences. Interp-B is very similar to backing off to lower orders without a penalty, or the Backoff-B method, because of the low escape counts. The smoothing methods and the escape counts define the performance of models, especially at higher orders and with limited training sequence data.

MVMs generalize the idea of combining an ensemble of predictive models. A multiple viewpoints system maintains an ensemble of predictive models based on various viewpoints with varying degrees of specificity. The viewpoints could be basic, derived, or linked viewpoints. Basic viewpoints often refer to the variable being predicted, and are usually observed. Derived viewpoints are obtained from basic viewpoints. Linked viewpoints are cross-type viewpoints which are the Cartesian products of simple and/or derived viewpoints. MVMs finally merge the predictions of these models according to each model's uncertainty at a given time step, using a weighted average as described in [26]. Each viewpoint model is assigned a weight depending on its cross-entropy at each time step. The weight for each model m at time-step t is given by $w_{\mathrm{m}}(t) = H(p_{\mathrm{max}})/H(p_{\mathrm{m}}(t))$, where $H(p_{\mathrm{m}}(t))$ is the entropy of the probability distribution and $H_{\mathrm{max}}(p_{\mathrm{m}})$ is the maximum entropy for a prediction in the distribution. The distributions are then combined by a convex combination, $p(t) = \frac{\sum_m w_m p_m(t)}{\sum_m w_m}$. Higher entropy values result in lower weights. In this way, models that are uncertain (i.e., have higher entropy) make a smaller contribution to the final predictive distribution.

There are two fundamental types of VLMMs which we refer to as long term models (LTMs) and short term models (STMs). LTMs are built on a corpus of songs, while STMs by reading in the symbols, one at a time, from a single composition. The goal of LTMs is to capture patterns that are common across all compositions, while STMs model song-specific patterns. Because songs often contain internal repetition, STMs are often highly predictive. On the other hand, if there is little or no repetition in a song, or the song contains few symbols, LTMs will be more predictive since these have seen much more data. It is also possible to combine the predictions of of the LTM and STM, in a manner analogous to merging viewpoint predictions.

A common domain-independent approach for evaluating the quality of model predictions is cross-entropy [23]. If the true distribution is unknown, the cross entropy of a test sequence of length T can be approximated by $H_c = -\frac{1}{T}\sum_{i=1}^{T}\log_2(p_i)$, which is the mean of the probabilities of true symbols evaluated from predictive distribution at each time step, measured in bits. A closely related concept, often used in natural language modeling is perplexity per symbol [23], defined to be $P = 2^{H_c}$, where H_c is the cross-entropy as described above. Perplexity has a simple interpretation, it is the number of choices that the model is confused between and would be equivalent to the model choosing uniformly between P choices.

6 Experiments

An LTM was built for each *raag*. This was done because the patterns utilized in a bandish are highly dependent on the raag. In future work, it would be straightforward to automatically classify the raag of each bandish, eliminating the need for manually dividing the compositions into *raags* [7].

The viewpoints used in the experiments are listed in Table 2. The viewpoints are obtained from the **kern scores. The note numbers are folded back to lie within the range of a single octave as the absolute notes range over three octaves. The note durations are quantized to the set of Dur={0.125, 0.25, 0.5, 1, 4/3, 2, 8/3, 3, 4, 6, 8}, (where $Dur = 1$ represents the quarter note) in order to limit the total number of duration classes. The melodic interval refers to the interval in semitone between two consecutive notes. This viewpoint, together with the Note viewpoint prevents any loss of information due to the octave folding of notes. The Note⊗Duration viewpoint is a cross-type viewpoint which models the inter-play between pitch and duration.

Table 2. Viewpoints used in the experiments

Viewpoint	**Description**	**Range**
Note (N)	The MIDI number of the Note	60, 61,. . ., 71
Contour (C)	A derived viewpoint indicating if the current note is increasing(+1) up, decreasing down(-1) the scale or unchanged(0) from the previous note	-1,0,+1
Interval Change (I)	A derived viewpoint indicating the number of semitones change from the previous note	-11, -10,. . ., 0,. . ., 10, 11
Note⊗Duration(N×D)	A cross-type viewpoint which is 2-tuple of Note and Quantized Duration	$\{(x, y) \mid x \in Note,\, y \in Dur\}$

For each *Raag*, a leave-one-out cross validation is performed using the compositions from that *Raag*. In each experiment, one composition is chosen as the test composition. The STM is built on the test composition for each viewpoint.

The LTM is trained on the rest of the compositions for each viewpoint. The predictive distribution at each time-step for each viewpoint of LTMs and STMs is computed. The predictions from each viewpoint are merged to obtain combinations such as NI(Note and Interval Change), NC(Note and Contour), NCI(Note, Contour and Interval Change), and NCI+N×D(Note, Contour, Interval Change, and the cross-type N×D). The LTM and STM are also combined into a single predictive distribution as discussed in Sec. 5 and for each case, the perplexity is computed from cross entropy of model predictions. The experiment is also repeated for various different maximum orders of VLMM which correspond to the maximum lengths of symbols modeled at each order.

7 Results and Discussion

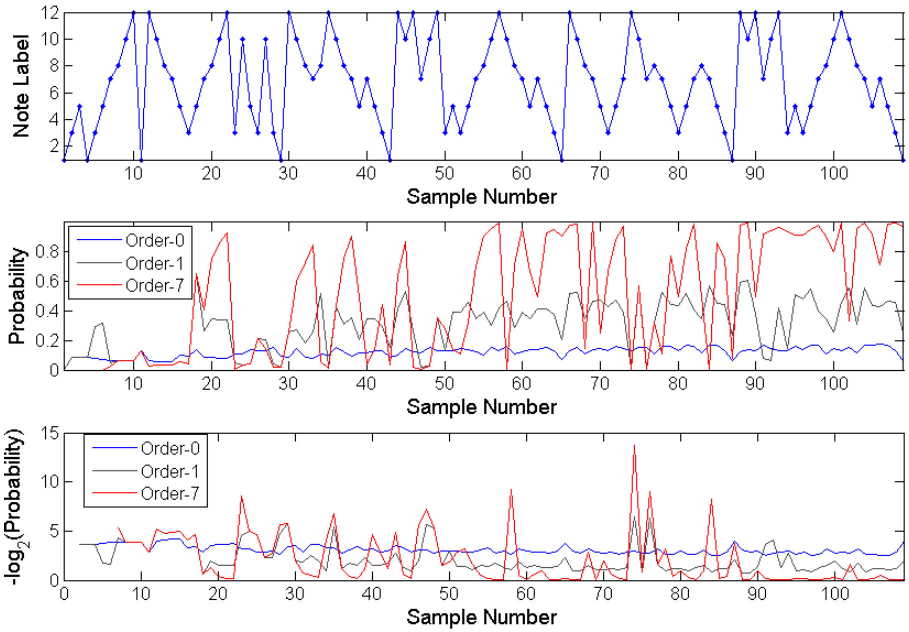

Fig. 2. (a) A single Yaman composition with 109 notes showing distinct repetitive note patterns. (b) The probability p and (c) the log probability, $-\log_2(p)$, of the true symbol as predicted by an STM for different VLMM orders 0 (priors), 1 and 7.

Figure 2 shows a single composition in *Raag Yaman* that contains some repeated motives. We build an STM on the composition and show the probability the model assigned to the true symbol at each timestep for different order STMs. Initially, the probability values are low, but as the composition progresses, the

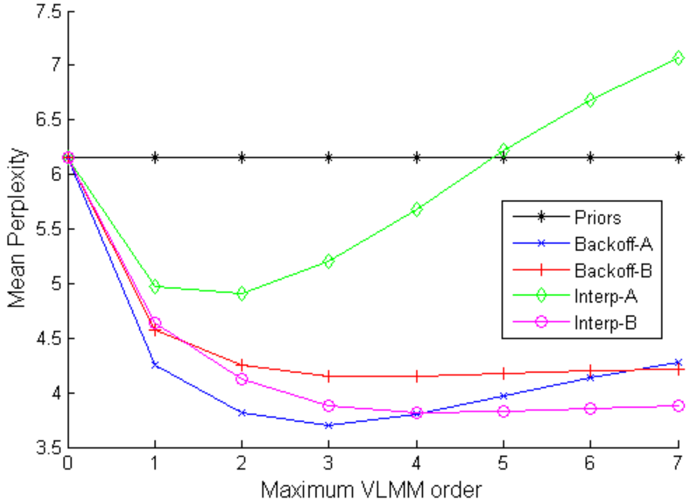

Fig. 3. A comparison of smoothing methods for LTM with the combined viewpoints NCI+N×D

STM is quick to learn the patterns and predicts the true symbol with a high probability. The negative log probability is also shown (in panel (c)), as it is more related to the information in musical events. The peaks in the curve indicate events which are unpredictable. Initially, we see that the negative log probability is higher, and as the piece progresses, the value decreases indicating higher predictability. But there are peaks in the later parts of the piece, which correspond to unexpected changes in note progression. We can also see that there are long term patterns in the piece, which leads to better performance when using higher order models. This demonstrates the effectiveness of STMs in modeling local repetitions. However, these kind of patterns were uncommon in the database – in general, the notated bandishes contained little internal repetition.

In the following discussion, we report results that were found to be statistically significant using a Tukey-Kramer multiple comparison test with confidence bounds at 99%. Figure 3 shows a comparison of smoothing methods for the LTM using the combined viewpoint NCI+N×D. The predictive performance is reported as the mean perplexity of the cross validation experiments, averaged across all the *Raag* models. The priors correspond to probability of true symbols computed through a zeroth order prior distribution of notes through a symbol count. When Interp-A method is used for smoothing, high escape counts lead to flatter predictive distributions with high entropy and hence the perplexity of LTM increases at higher model orders. For reporting further results, we choose the Backoff-A smoothing method which provides a good balance between the probabilities of seen sequences and those of unseen sequences.

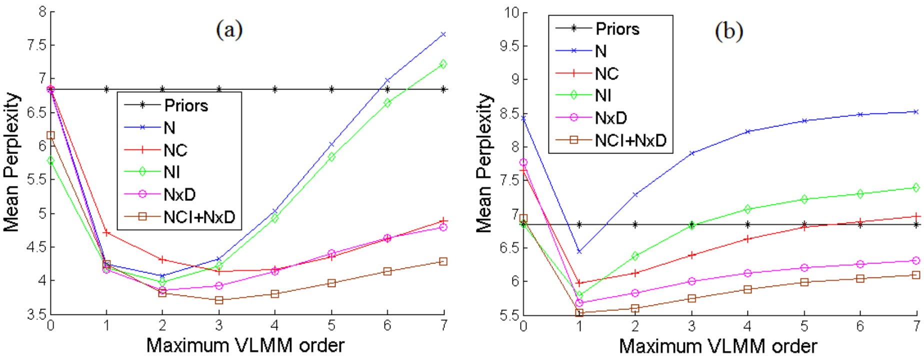

Fig. 4. A comparison of viewpoints with the smoothing method Backoff-A. Panel (a) shows the LTM performance and (b) shows the STM performance

Table 3. The mean perplexity at maximum order of 3 for different *raags* and different viewpoints with LTM and STM. The perplexity of LTM and STM combined using the combination NCI+N×D is also shown. The smoothing method used is Backoff-A.

Raag	LTM						STM					Combined
	Prior	N	NC	NI	N×D	NCI+N×D	N	NC	NI	N×D	NCI+N×D	NCI+N×D
Yaman	6.89	4.58	4.39	4.54	4.14	3.99	8.70	6.80	7.23	6.50	6.05	4.21
Bageshri	6.76	4.15	4.04	4.05	3.72	3.54	7.43	6.11	6.57	5.74	5.54	4.00
Khamaj	7.33	4.48	4.15	4.27	4.17	3.79	7.42	6.16	6.45	5.63	5.55	3.73
Bihag	6.12	3.83	3.67	3.72	3.36	3.18	7.72	6.31	6.98	5.92	5.79	3.47

Figure 4 shows the mean perplexity obtained in the cross validation experiment with different viewpoint combinations, averaged across the *Raags* for both LTM and the STM. Using just the Note viewpoint, the LTM has a lowest perplexity (4.06) when the maximum VLMM order is 2, while we see that combining viewpoints as in NCI+N×D provides the lowest perplexity (3.70) at order 3. Combining viewpoints for both LTM and STM is useful, as the perplexity for the combined decisions are lower than individual viewpoints. This is especially true in LTMs, where the use the multiple viewpoints brings down the perplexity significantly at higher orders. In the case of STMs, the minimum perplexity is consistently achieved at a maximum order of 2. Using the combination of viewpoints NCI+N×D, the perplexity at order-2 drops to 5.54 from the Note viewpoint perplexity of 6.45. The optimal order for LTMs is seen to be 3 while the optimal order for STMs is 1. The low optimal order for STMs show that the training compositions had unpredictable note progressions, even within the same *raag* and hence higher orders are not particularly useful in STMs.

Table 3 consolidates the LTM and STM performance with each *Raag* for different viewpoint combinations. It also tabulates the combined performance of LTM and STM. We see that the model performance is similar across different *Raags*, which indicates that the technique is generalizable to different *Raags*.

The combined performance of LTM and STM is intermediate between LTMs and STMs.

The *bandishes* used in the experiments only provide a basic framework for the actual rendition and lack the repetitions which we normally see in the actual renditions. Further, the *bandishes* are short with about 100 notes per composition. These qualities are reflected in the relatively poor performance of the STMs. Figure 4 shows that the best case perplexity of STM was 1.84 higher than that of the LTM. This is quite different than what was reported in [8], where STMs significantly outperformed LTMs. However the tabla compositions contained, on average 1000 symbols, an order of magnitude more data than the average *bandish*.

The cross entropy of the STM predictions is computed over the entire composition. However, when the STM performance was evaluated only in the second half of the compositions, after the STM has evolved, there was a considerable decrease in perplexity. The smoothing method used contributes to the increasing trend at higher orders. Multiple viewpoints help to reduce the perplexity of melodic prediction and the combination NCI+N$\times$D gives the lowest perplexity.

8 Conclusions and Future Work

MVMs have been shown to effective at predicting melodies in NICM, outperforming fixed and low-order Markov models. This work provides further evidence that MVMs are general tools that can be used to model temporal sequences in a variety of musical genres.

We plan to extend the experiment to include more *raags* once the database is complete. We also plan to extend the work to synthesized audio and develop intermediate-term models based on unsupervised clustering of *bandishes*.

References

1. G. Assayag and S. Dubnov. Universal prediction applied to stylistic music generation. In *Mathematics and Music*, pages 147–160. Springer-Verlag, 2002.
2. G. Assayag and S. Dubnov. Using factor oracles for machine improvisation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 8(9):604–610, 2004.
3. G. Assayag, S. Dubnov, and O. Delerue. Guessing the composer's mind: applying universal prediction to musical style. In *Proceedings of the 1999 International Computer Music Conference*, pages 496–499. San Francisco: ICMA, 1999.
4. V. N. Bhatkhande. *Hindustani Sangeet Paddhati: Kramik Pustak Maalika Vol. I–VI.* Sangeet Karyalaya, 1990.
5. P. Chordia. A system for the analysis and representation of bandishes and gats using humdrum syntax. In *Frontiers of Research in Speech and Music Conference*, 2007.
6. P. Chordia, A. Albin, and A. Sastry. Evaluating multiple viewpoints models of tabla sequences. In *ACM Multimedia Workshop on Music and Machine Learning*, 2010.

7. P. Chordia and A. Rae. Automatic raag classification using pitch–class and pitch–class dyad distributions. In *Proceedings of 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.
8. P. Chordia, A. Sastry, and S. Senturk. Predictive tabla modelling using variable length markov and hidden markov models. *Journal of New Music Research*, 40(2):105–118, 2011.
9. J. Cleary and W. Teahan. Experiments on the zero frequency problem. In *Proceedings of Data Compression Conference, DCC '95*, 1995.
10. D. Conklin. Prediction and entropy of music. Master's thesis, University of Calgary (Canada), 1990.
11. D. Conklin. Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.
12. D. Conklin and C. Anagnostopoulou. Representation and discovery of multiple viewpoint patterns. In *Proceedings of the 2001 International Computer Music Conference*, pages 479–485. International Computer Music Association, 2001.
13. D. Conklin and J. G. Cleary. Modelling and generating music using multiple viewpoints. In *Proceedings of the First Workshop on AI and Music*, pages 125–137, Menlo Park, CA, 1988. AAAI Press.
14. D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24:51–73, 1995.
15. A. Cont, S. Dubnov, and G. Assayag. Guidage: A fast audio query guided assemblage. In *Proceedings of International Computer Music Conference (ICMC)*. Copenhagen, September 2007.
16. S. Dubnov. Analysis of musical structure in audio and midi using information rate. In *Proceedings of International Computer Music Conference (ICMC)*, 2006.
17. S. Dubnov, G. Assayag, and A. Cont. Audio oracle: A new algorithm for fast learning of audio structures. In *Proceedings of International Computer Music Conference (ICMC)*, Copenhagen, September 2007.
18. S. Dubnov, G. Assayag, and R. El-Yaniv. Universal classification applied to musical sequences. In *Proceedings of the 1998 International Computer Music Conference*, pages 332–340. San Francisco: ICMA, 1998.
19. S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine-learning methods for musical style modeling. *IEEE Computers*, 36(10):73–80, 2003.
20. D. Huron. *Sweet Anticipation: Music and the Psychology of Expectation.* MIT Press, 2006.
21. R. Jha. *Abhinav Geetanjali Vol. I–V.* Sangeet Sadan Prakashan, 2001.
22. O. Lartillot, S. Dubnov, G. Assayag, and G. Bejerano. Automatic modelling of musical style. In *Proceedings of the 2001 International Computer Music Conference*, pages 447–454. San Francisco: ICMA, 2001.
23. C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*, pages 60–78. MIT Press, 2002.
24. F. Pachet. The continuator: Musical interaction with style. In *Proceedings of International Computer Music Conference, Gotheborg (Sweden), ICMA*, pages 211–218, 2002.
25. M. Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Cognition.* PhD thesis, City University, London, 2005.
26. M. Pearce, D. Conklin, and G. Wiggins. *Methods for Combining Statistical Models of Music*, volume 3310, pages 295–312. Springer Berlin, 2005.

27. M. Pearce, M. H. Ruiz, S. Kapasi, G. A. Wiggins, and J. Bhattacharya. Unsupervised statistical learning underpins computational, behavioural and neural manifestations of musical expectation. *NeuroImage*, 50(1):302–313, 2010.
28. J. L. Triviño-Rodriguez and R. Morales-Bueno. Using multiattribute prediction suffix graphs to predict and generate music. *Computer Music Journal*, 25(3):62–79, 2001.
29. P. von Hippel. Redefining pitch proximity: Tessitura and mobility as constraints on melodic intervals. *Music Perception*, 17(3):315–327, 2000.
30. I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.
31. I. H. Witten, L. C. Manzara, and D. Conklin. Comparing human and computational models of music prediction. *Computer Music Journal*, 18(1):70–80, 1994.