

Evaluating Multiple Viewpoint Models of Tabla Sequences

Parag Chordia
Georgia Institute of
Technology
840 McMillan St
Atlanta, GA
ppc@gatech.edu

Avinash Sastry
Georgia Institute of
Technology
840 McMillan St
Atlanta, GA
asastry3@gatech.edu

Aaron Albin
Georgia Institute of
Technology
840 McMillan St
Atlanta, GA
aalbin3@mail.gatech.edu

ABSTRACT

We describe a realtime tabla generation system based on a variable-length n -gram model trained on a large symbolic tabla database. A novel, parametric smoothing algorithm based on a family of exponential curves is introduced to control the relative weight of high- and low-order models. This technique is shown to lead to improvements over a back-off smoothing for our tabla database. We find that cross-entropy is lowest when the coefficient of the exponential curve is between 1 and 2 and increases for values outside of this optimal range. The basic n -gram model is extended to model dependencies between duration, stroke-type, and meter using cross-products in a Multiple Viewpoints (MV) framework, leading to improvements in most cases when compared with independent stroke and duration models.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

General Terms

algorithms, experimentation

Keywords

Markov modeling, n -grams, context-dependent, variable-length, tabla, multiple viewpoints, music

1. INTRODUCTION AND BACKGROUND

Anticipation is an essential component of listening. Regularities in the structure of music lead to expectations that focus attention and guide perception [12]. For musicians, anticipation is essential for synchronization. Markov and n -gram models have been extensively used to model temporal structure in music [1]. They have been extensively used in algorithmic composition, timbral analysis [2] [11], and music cognition [14].

In this paper, we extend a previous predictive model for tabla based in an ensemble of n -gram models [6] by introducing new

smoothing techniques, and by modeling dependencies between rhythm and timbre in our short-term model (STM). The immediate motivation for this work is to create a realtime generative tabla system based on n -gram modeling of tabla sequences.

A significant problem that arises with fixed-order models is that, as the order n increases, the number of total n -grams increases as v^n , where v is the number of symbols. In music applications, such as melody prediction, where the past ten events could easily influence the next event, and where there might be a dozen or more symbols, we are left attempting to assess the relative frequency of greater than 12^{10} n -grams. Even for large databases, most n -grams will be unseen, leading to the so-called zero frequency problem [7]. This sparsity problem leads to a fundamental tradeoff between using the predictive power of longer context and the increasing unreliability of higher order n -gram counts. Variable-length n -gram models attempt to overcome this problem in two ways: 1) by building many fixed-order models and integrating information across orders (smoothing), 2) and by reserving a certain amount of probability mass for unseen n -grams (escape probabilities). We describe these techniques in Section 3.

Variable length n -gram modeling is an ensemble method in which the predictions of many fixed-order models are integrated. Ensemble methods such as boosting have been shown to be effective for classification tasks [9]. Multiple Viewpoint systems, introduced by Conklin and Witten [8], and developed by others such as Pearce [16], generalizes the idea of integrating an ensemble of predictive models. It is based on the fact that music can be simultaneously represented in many ways. For example, a melody can be thought of in terms of chromatic pitches, intervals, scale degrees, or contour. A rhythmic pattern can be thought of in terms of onset times, durations or position-in-bar. If we are trying to predict the next note in a melody, having multiple representations is useful in capturing structure that is obvious given one representation, but less so in another. A scale-degree representation of a melody might make it obvious that the chromatic pitch, say B, is actually the leading tone, making it very likely that the next note is C. However, if the training database contains many melodies in many different keys, this might not be obvious from the chromatic pitch representation. We describe the multiple viewpoints framework in Section 3.1.

Tabla is the most widely used percussion instrument in Indian music, both as an accompanying and solo instrument. Its two component drums are played with the fingers and hands and produce a wide variety of timbres, each of which has been named. A sophisticated repertoire of compositions and theme-based improvisations has developed over hundreds of years. Tabla is a natural candidate of n -gram modeling because it consists, at a basic level, of sequence of discrete states that are temporally structured. Tabla is particularly interesting because of the complex patterns and depen-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MML'10, October 25, 2010, Firenze, Italy.

Copyright 2010 ACM 978-1-4503-0161-9/10/10 ...\$10.00.

dencies that are present in typical compositions. Although tabla is primarily learned as part of an oral tradition, it is also notated using a system that indicates strokes and their durations. Little work to date has been done on statistical modeling of tabla. Gillet [11] and Chordia [5] both used an HMM framework for tabla transcription, while Bel and Kippen [3] created a model of tabla improvisation based on a context-free grammar, one of the earliest computational tabla models. The current work builds on the generative tabla system of Rae and Chordia [17].

2. TABLA DATABASE

The database used for training the model is a set of traditional tabla compositions compiled by tabla maestro Alok Dutta [10]. The compositions were encoded in a Humdrum-based syntax called `**bol` that encoded the stroke name and duration [5]. The online database consists of 35 compositions in a variety of forms. Altogether there are 27,189 strokes in the dataset composed of 42 unique symbols.

3. N-GRAM MODELING

N -gram modeling is a commonly used technique to probabilistically model sequences of elements such as phonemes in speech, letters in a word or musical notes in a phrase. [13] N -grams can be efficiently stored in a tree-shaped data structure, commonly referred to as a trie or prefix tree.

Branches represent the succession of certain symbols after others, and a node at a certain level of the trie 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. After the trie has been built, it can be used to predict the next symbol given a test sequence. The modeling and evaluation framework was implemented in C++ as an external object in Max/MSP along with supporting patches. For more information about the trie data structure used in this experiment see [6].

As noted above, the zero frequency problem occurs because most n -grams in high-order models will never have been observed [7]. This problem is resolved by reserving an escape probability for each level of the trie. Whenever an event returns zero probability, it returns the escape probability instead. Based on the results of Bell and Witten [18], we have implemented the Poisson distribution method. The escape probability for each level is assigned by $e(n) = \frac{T_1(n)}{N(n)}$, where T_1 is the number of tokens that have occurred exactly once and N is the total number of tokens seen by the model so far.

Smoothing addresses the tradeoff between the specificity of higher-order models and the reliability of the n -gram counts for lower-order models. Since higher order models are much sparser, many n -grams will be assigned zero probability, and counts for n -grams that have been observed will tend to vary greatly based on the particular training database. This variance can be reduced by incorporating information from lower order models. There are two basic types of smoothing algorithms: back-off models and interpolation models. Given a test sequence, a back-off model will search for the entire sequence, and if no match is found in the trie, the process continues recursively after dropping the first element of the sequence, stopping once a positive match is found and the count for that n -gram count is greater than some threshold. Interpolated smoothing, by contrast, always incorporates lower order information even if the n -gram count in question is non-zero.

In our previous work [6], two smoothing methods were studied, Kneser-Ney (KN) and an averaging method we termed $1/N$.

□

Figure 1: Family of exponential curves used for parametric smoothing model. Each curve represents a given exponent coefficient and shows relative weight assigned to each order while taking weighted average.

KN was adopted directly from language processing because earlier work showed it to be a superior smoothing method in the context of natural language processing [4]. In the $1/N$ smoothing method, weights for the n -th order model are given by $\frac{1}{(\maxOrder-n+1)}$, giving greater relative weight to higher-order models. We found that the $1/N$ model outperformed KN.

We introduce a back-off model and a novel, parametric approach based on generalizing the $1/N$ technique. For the back-off model, the match threshold was set to two; if the pattern was observed only once at a given level, we continued to back-off. The motivation for setting the threshold greater than one was to ensure that, particularly for higher-order models, that a match was more likely to be a genuine pattern. The parametric model is based on a family of exponential curves given by

$$w(n) = a \left\{ \left(1 - \frac{c}{a} \right) \left(\frac{n}{\maxOrder} \right)^x \right\} + c \quad (1)$$

$n = 0, 1, 2, 3 \dots \maxOrder$

As x increases, the curve rises more steeply, giving more importance to higher-order models as can be seen in Figure 1. The parameters a and c are used to set the minimum and maximum allowable weights.

3.1 Multiple Viewpoints

A multiple viewpoints system tracks variables such as pitch and rhythm independently, maintaining many predictive models simultaneously. The final prediction is obtained by combining these predictions into a meaningful set of basic parameters. Such a system incorporates information from different variables and can also model complex relationships between two or more of these variables, making use of that information to strengthen its prediction. For a more detailed explanation of the multiple viewpoints system, see [8]. We use 4 viewpoints in our system - Strokes (tabla notation), Durations (Interval between two strokes), Stroke x Duration (a cross-type consisting of a stroke and its duration) and Stroke x PositionInCycle (PositionInCycle is the occurrence of a stroke at a point in the rhythmic cycle).

A common limitation of predictive models built on large databases is that the model is usually unaware of any patterns specific to a particular song. The model becomes too general to be effective; patterns and predictions which seem obvious to humans are missed because they are infrequent in the training database. To solve this problem, we used two models: a long-term model (LTM) built on the entire training database, and a short-term model that starts out empty and is built up as a particular composition is processed. Each of these viewpoints has a long term model (LTM), which is fed with a database of tabla compositions, and a short term model (STM), which is trained only upon the current song being evaluated.

3.2 Merging Model Predictions

An important point here is the process of merging the predictions of each of the models. Though there are many different ways to do this, we use a weighted average as described in [16]. Each viewpoint model is assigned a weight depending on its cross-entropy at each time step. The weight for each model is given by $w_m =$

Model	Strokes		Strokes MV		Durations		Durations MV		Stroke-Duration		Stroke-PIC	
	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
BO	1.437	0.105	1.270	0.095	1.081	0.023	0.870	0.004	1.940	0.374	1.370	0.486
1/N	1.055	0.454			0.530	0.147			1.432	0.655	1.350	0.452
P (x=1)	1.003	0.407	1.017	0.341	0.626	0.130	0.505	0.102	1.375	0.524	1.363	0.452

Table 1: Summary of cross-entropy results for for back-off, 1/N, and parametric (with exponent coefficient equal to 1) smoothing methods. Results for combined models using a maximum order of 10. MV refers to the multiple-viewpoints model in which the Stroke-Duration and Stroke-PIC viewpoints have been incorporated.

$H(p_m)/H_{\max}(p_m)$, where $H(p_m)$ is the entropy of the probability distribution and $H_{\max}(p_m)$ is the maximum entropy for a prediction in the distribution. Higher entropy values result in lower weights. In this way, models that are uncertain (i.e., have higher entropy) make a lesser contribution to the final distribution. The distributions are then combined by taking their weighted average.

When a prediction is to be made at a given time-step, the LTM and STM are combined into a single distribution for each of three tries (Strokes, Durations, and Stroke-Durations). Additionally, the Strokes-Durations cross-type is combined with the basic stroke and duration types by marginalizing the Strokes-Durations predictive distribution. The Strokes-PositionInCycle viewpoint is only used in in the STM. The LTM cannot be used for this viewpoint because there are many different rhythmic cycles in the corpus leading to very different relations between strokes and metric position. At each time-step, the model checks the current position in the cycle (the length of the cycle is defined by the length of the theme for that song – if the theme is not complete yet, then this trie is not used) and calculates the probabilities for strokes at this position. The resultant distribution is then combined with Strokes distribution. The Strokes and Durations models are then normalized leading to a single predictive distribution for each that can be used to calculate the cross-entropy.

4. REALTIME GENERATION

Chordia [17] introduced a generative tabla system capable of improvising in realtime. New phrases were generated based on recombining thematic material from a database of compositions. The system produced improvisations that were judged to be musical and stylistically appropriate when compared to the performance of an expert tabla player [17]. Here we extended the system so that new phrases could be generated based on the learned STM for a given composition. The program generates phrases continuing from the Position-In-Cycle at which it was stopped. At each time-step the predictive distribution is calculated for strokes and durations as described in Section 3.2. The next stroke and its duration is chosen by sampling the predictive distributions. The process continues until a phrase containing a specified number of beats is generated.

5. RESULTS AND FUTURE WORK

Cross-validation was performed using a leave-one-out design. For each of the 35 compositions, training of the LTM was performed on the remaining 34. The STM was trained on the remaining song. Reported results were averaged over all 35 trials. A common domain-independent approach for evaluating the quality of the models’ predictions is cross-entropy [15]. If the true distribution is unknown, the cross entropy can be approximated by $-\frac{1}{n} \sum_{i=1}^n \log_2(p_i)$, which is the mean of the entropy values for a given set of predictions. To illustrate, at a given step t , we note the true symbol. We then look at the predictive distribution for sym-

bols at step $t - 1$ and calculate the entropy for the true symbol at step t . After running through all the symbols in the test set, these entropies are averaged, giving a cross-entropy result for that particular test set. An informal qualitative evaluation was also performed on the generated output. We are currently preparing a more detailed qualitative evaluation using a web-based survey similar to Chordia [17].

Table 1 summarizes the cross-entropy results for the different smoothing models and different prediction types. For all tasks the interpolation-based smoothing methods outperform the back-off method. Cross-entropy for stroke prediction using the multiple viewpoints (MV) is 1.017 for the parametric model (using an exponent of 1) and 1.270 for the back-off model, with similar differences for Durations and Stroke-Duration. This compares favorably to a baseline cross-entropy of 3.614 when using only using prior probabilities for strokes. When predicting Stroke-PIC there are no significant differences between smoothing methods.

Incorporating the Stroke-Duration and Stroke-PIC viewpoints improves performance when predicting strokes (1.237 vs 1.437) and durations (1.081 vs .870) for the back-off model. For the parametric model, these additional viewpoints do not seem to significantly improve performance.

Median entropy is less than average entropy for all models, with a greater difference for back-off models. The distribution of entropy values showed a larger peak near zero and "fatter tail" for back-off vs. interpolation methods. This is perhaps due to the fact that back-off methods do well when a high-order match is made but do not degrade as well because lower-order information is not fully integrated.

Figure 2 shows results for different values of the exponent for the parametric model. It seems that there is an optimum range for the exponent between .5 and 2, with values outside of this range leading to worse performance. While these results are not statistically significant, they suggest that for interpolation methods the relative weights of models is important. More specifically, it seems that while higher-order models should receive greater weight, increasing their weight beyond a certain point decreases performance.

Initial qualitative results were promising. Phrases generated using the STM were both novel and musical, essential components of an artificially creative system. It should be emphasized that generation is significantly more challenging than evaluation based on cross-entropy, where only a single note is predicted after all previous strokes have been revealed; during active generation there is no reference sequence and sequences can quickly diverge towards unacceptable results. Interestingly, although the Stroke-PIC viewpoint was not useful for reducing cross-entropy when using parametric smoothing, it proved to be crucial when generating novel sequences. This was due to the fact that phrases were better aligned with the underlying meter.

We plan to continue exploring the MV framework for automatic tabla composition. We are particularly interested in the qualitative



Figure 2: Cross-entropy as a function of exponent coefficient in parametric model. There seems to be an optimal range between 1 and 2.

effects of different smoothing techniques and the use of additional viewpoints to control musical structure.

6. REFERENCES

- [1] C. Ames. The markov process as a compositional model: A survey and tutorial. *Leonardo*, 22(2):175–187, 1989.
- [2] J.-J. Aucouturier, F. Pachet, and M. Sandler. The way it sounds: Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035, 2005.
- [3] B. Bell and J. Kippen. Bol processor grammars. *Understanding music with AI: perspectives on music cognition*, pages 366–400, 1992.
- [4] S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 310–318, 1996.
- [5] P. Chordia. *Automatic Transcription of Solo Tabla Music*. PhD thesis, Stanford University, Dec. 2005.
- [6] P. Chordia, A. Albin, and A. Sastry. Multiple viewpoints modeling of tabla sequences. In *Proceedings of International Conference on Music Information Retrieval*, 2010.
- [7] J. G. Cleary and W. J. Teahan. Experiments on the zero frequency problem. In *DCC: Proceedings of the Conference on Data Compression*, page 480. IEEE Computer Society, 1995.
- [8] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24:51–73, 1995.
- [9] T. G. Dietterich. Ensemble methods in machine learning. In *International Workshop On Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.
- [10] A. E. Dutta. *Tabla: Lessons and Practice*. Ali Akbar College, 1995.
- [11] O. Gillet and G. Richard. Supervised and unsupervised sequence modeling for drum transcription. In *Proceedings of International Conference on Music Information Retrieval*, pages 219–224, 2007.
- [12] D. Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.
- [13] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*, pages 60–78. MIT Press, 2002.
- [14] Pearce, H. Ruiz, Kapasi, Wiggins, and Bhattacharya. Unsupervised statistical learning underpins computational, behavioural and neural manifestations of musical expectation. *NeuroImage*, 50(1):302–313, 2010.
- [15] M. Pearce. *The construction and evaluation of statistical models of melodic structure in music perception and cognition*. PhD thesis, City University, London, 2005.
- [16] M. Pearce, D. Conklin, and G. Wiggins. *Methods for Combining Statistical Models of Music*, volume 3310, pages 295–312. Springer Berlin, 2005.
- [17] A. Rae and P. Chordia. Tabla gyan: An artificial tabla improviser. In *First International Conference on Computational Creativity (ICCCX)*, 2010.
- [18] 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*, 34(4):1085–1094, 1991.