PROBABILISTIC PREDICTION OF RHYTHMIC CHARACTERISTICS IN MARKOV CHAIN-BASED MELODIC SEQUENCES

Bongjun Kim and Woon Seung Yeo
Audio & Interactive Media (AIM) Lab
Graduate School of Culture Technology, KAIST
Deajeon, 305-701, Korea Republic
iambongjun@gmail.com, woony@kaist.edu

ABSTRACT

Markov chain models have been widely used for algorithmic composition and machine improvisation. In this paper, we introduce a probabilistic prediction model of rhythmic characteristics of Markov chain-based note sequences. For this purpose, we propose an algorithm to generate a revised Markov chain model and calculate the onset probabilities of notes at each onset position in one measure. As an application of this algorithm, we present an interactive improvisation system which uses a customized syncopation index as an input parameter and allows the user to control the level of syncopation and rhythmic tension in real-time.

1. INTRODUCTION

Auto-generation of music with mathematical algorithms such as neural networks, genetic algorithms, generative grammars and cellular automata, has been researched for several decades. Markov chains are also widely used in algorithmic composition and machine improvisation system because it is computationally cheap to learn the style of existing music and imitate the music with simple probabilistic calculation [5, 7]. Markov chains imitate a style of sequence of musical events such as notes and chords with transition probabilities between events. This probability-based learning and creation enable us to generate more creative musical outcomes [1, 10].

Despite of the advantages of Markov chains, they are not suited for interactive control. Overcoming this drawback, Pachet et al. suggested methods to control the generation of event sequences from Markov chain models for interactive applications considering constraints for user inputs. But they focused on only pitch, not rhythmic factors [8, 9].

This paper addresses the issue of controlling rhythm of note sequences generated from a first-order Markov chain which is the simplest type. Our approach is to predict the onset probabilities of musical notes and to select the initial state of the Markov chains depending on the probabilities. As an application of the algorithm, we present an interactive improvisation system built in Max/MSP where users can control the amount of syncopation of the rhythm in real-time.

2. RHYTHM GENERATION AND ANALYSIS WITH MARKOV MODELS

Figure 1 illustrates an example of a simple first-order Markov chain for rhythm generation, which can be derived from the user's input melodies or sample pieces. Each node represents the duration of a note, and transition probabilities between nodes show their mutual dependencies. For example, a quarter note is followed by an eighth note with the probability 0.5 and, in turn, an eighth note is followed by an eighth rest with the probability 0.3. The outgoing probabilities from each state must sum to 1.

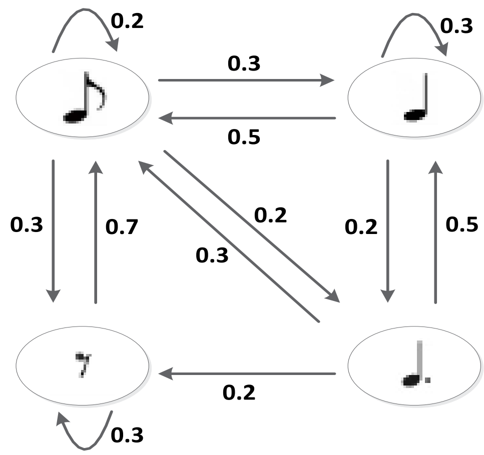

Figure 1. An example of a first-order Markov chain for rhythm generation

The Markov chain model is used not only to imitate a style of existing pieces and generate melodies, but also to calculate probabilities of future events using transition matrices, which means that we can predict the possibility of occurrence of the *n* th note from the initial note [4]. For example, if each state (or node) of a Markov chain denote pitch, we can calculate the probability that the pitch of the third note will be E or C. However, if it is a rhythm model involving the duration of notes shown in Figure 1, it is hard to predict the rhythmic characteristics per bar. This is because the onset position of each note is affected by the durations of their preceding notes. Figure 2 illustrates the problem. Depending on the combination of the first two events (either notes or rests) the third event is in a different position. With the simple Markov chain model in Figure 1, we can only calculate the probability that the third event is an eighth note, not the probability that it occurs at a specific position. But we need the latter to predict rhythmic characteristics in one measure.

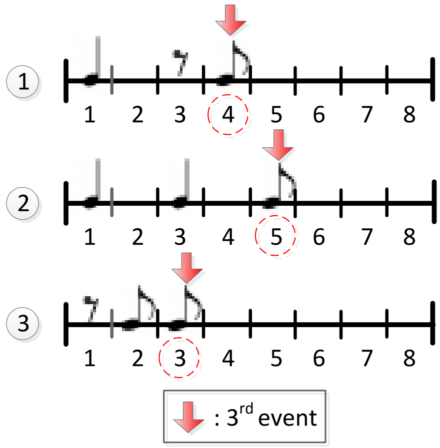

Figure 2. The onset positions of the third event vary with the durations of preceding events.

3. THE ALGORITHM

3.1. Revised Markov Chain Model

In order to calculate the probability of each state at a specific position in a bar, the simple Markov chain above needs to be revised. Figure 3 illustrates the structure of a new Markov chain model modified from the original one in Figure 1. To generate this model, the number of "unit pulses" in one bar needs to be defined. Here, we divide one measure into 8 beat pulses and, assuming 4/4 meter, the duration of a unit pulse corresponds to an eighth note. Compared to Figure 1, nodes whose duration is longer than one unit pulse are divided into multiple nodes so that each state can take only the unit duration, an eighth note.

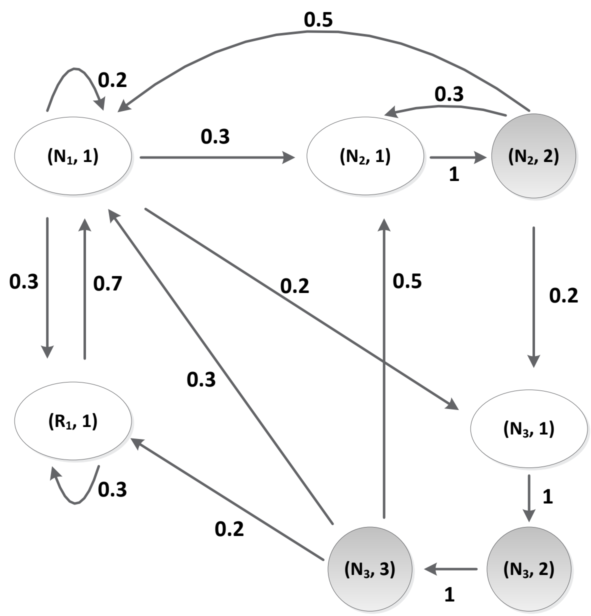

Figure 3. Revised Markov chain model (modified from Figure 1).

Each state is denoted as (a, b): the first element a shows the type and duration of the event (e.g., N_d or R_d, where N: note, R: rest, and d: duration) and the second element b means the "counter" parameter ranging from 1 to the duration of the event. For example, the dotted quarter note has three states, $(N_3, 1)$, $(N_3, 2)$, and $(N_3, 3)$. The transition probabilities to newly added nodes (gray-colored in Figure 3) are set to 1 because these events always occur after the first state of each event.

3.2. Onset Probability

The purpose of the revised Markov chain model is to derive state probabilities of each onset position in one measure from a simple Markov chain model. The state probabilities $\pi(k)$ are calculated using equation (1).

$$\pi(k) = \pi(0) \times T^{k-1} \quad (1)$$

In the equation, k denotes time step indicating onset positions from 1 to 8 and $\pi(0)$ means initial state. T is a transition matrix modified from the transition matrix of simple Markov chains in Figure 1 by a simple procedure. Firstly, the new transition matrix is initialized as a zero-matrix. If the size of previous transition matrix is $(N \times N)$, the size of revised transition matrix is $(d_1 + \cdots + d_N) \times (d_1 + \cdots + d_N)$, where d_N is the duration of note or rest N. Their transition probabilities, S' are derived using equation (2) and (3), where S denotes transition probabilities of the simple Markov model.

$$S'_{(a,b),(a,b+1)} = 1 \ \ for\ a = 1,\ldots,N, b = 1,\ldots,d_a - 1 \quad (2)$$

$$S'_{(i,d_a),(j,1)} = S_{ij} \ \ for\ i = 1,\ldots N,\ j = 1,\ldots N \quad (3)$$

Now, we can calculate onset probabilities at each position in one measure. The probability that note a occurs at position k can be calculated by following equation (4).

$$p_a(k) = \pi(k)_{a,1} \ \ for\ a \notin rest \quad (4)$$

Finally, the onset probability at position k, $P_{onset}(k)$ is derived from equation (5).

$$p_{onset}(k) = \sum_{a=N_1}^{N_t} \pi(k)_{a,1} \ \ for\ a \notin rest, \quad (5)$$

where t is the total number of notes.

Figure 4 is onset probability distribution which allows us to predict how the rhythm will be generated from the Markov chain in Figure 1. We also know that the probabilities change depending on the initial events.

4. APPLICATION

As an application of the algorithm, we developed an interactive improvisation system where users can control the amount of syncopation of the rhythm in real-time.

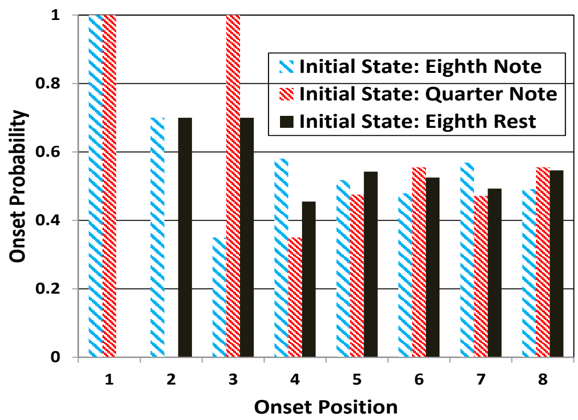

Figure 4. Onset probability distribution in one measure with different initial states.

4.1. Syncopation Index

Syncopation is one of the rhythmic characteristics which produce rhythmic tension. Many researchers have explored the method of measures and perception of syncopation [3], and it has been used as user's input parameter of interactive music system [11].

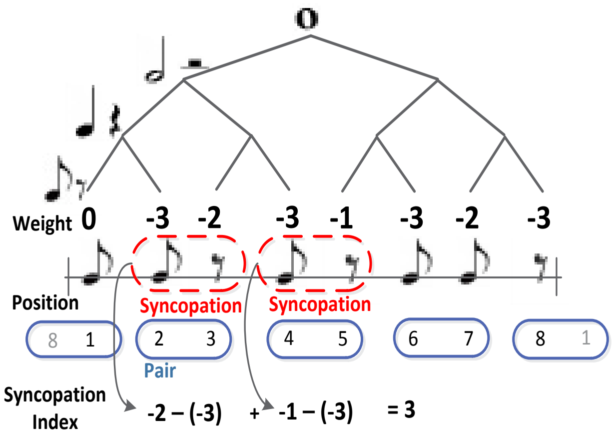

Figure 5. A rhythm tree and syncopation index in 4/4 meter [6].

In order to calculate syncopation value with our probabilistic model, we follow the Longuet-Higgins and Lee's definition of syncopation [2, 6]. They defined metrical tree having weight values of zero or less to calculate syncopation index (figure 5). The weights describe how much the metrical positions contribute to the measure's rhythmic feeling. According to their papers, syncopations occur when a rest (or tied note) is preceded by a note of lesser weight, and the difference in weights of rest and note means each syncopation value. Thus the sum of all syncopation values is the syncopation index for the rhythm. For example, in Figure 5, the rhythm has two syncopations. The first syncopation value is 1, and the second one is 2. So the syncopation index for the rhythm equals to 3 (more detailed algorithm can be found in [2]).

We use this concept to probabilistically predict syncopation index of note sequences to be generated from Markov chains. In our model, comparing onset probabilities for each pair, we assign a rest at the position which has less onset probability, and another position is assigned by a note. For example, given the onset probabilities in figure 4, when an initial event is an eighth note, a note at onset position 2 and a rest at position 3 are assigned. Thus, using the same calculation method shown in figure 5, its syncopation index can be predicted as 3.

As shown in this example, we can predict probabilistic syncopation index with different initial states, which means that syncopation indices can be user's input parameters in Markov chain-based improvisation system.

4.2. Max/MSP External Object

The algorithm was implemented in one Max/MSP external object named *MC_OnsetProb*. The object receives a list of sets of the transition matrix of simple Markov chain model, and it converts them to the transition matrix for revised Markov chain model. It also calculates the onset probabilities at each pulse and syncopation index. The output of this external object is a list of sets of onset probability, syncopation index, and related initial state.

4.3. Real-time Improvisation System

The system overview of Markov chain-based improvisation system is shown in Figure 6, which consists of three modules, (1) Markov Chain Analysis, (2) Syncopation Index Calculation, and (3) Melody Generation.

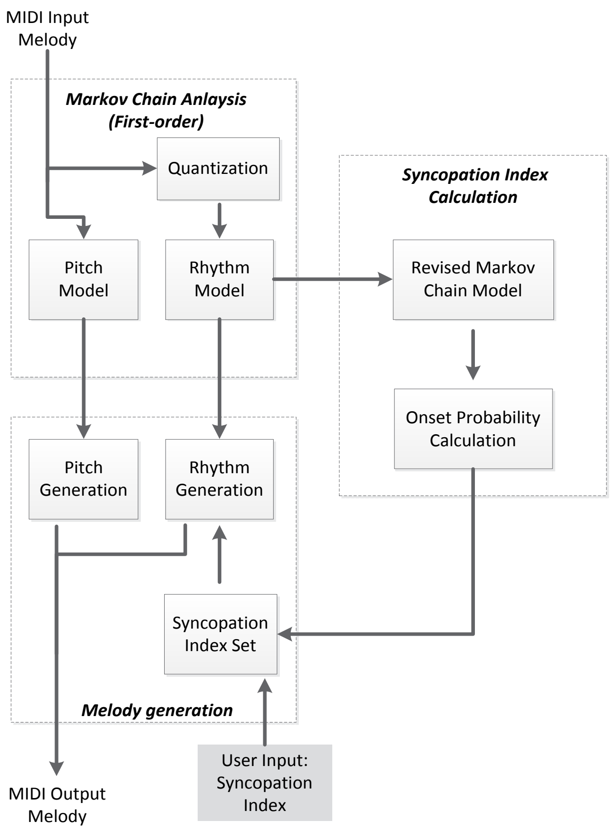

Figure 6. System overview of improvisation system

Firstly, a user inputs monophonic melody with an external midi device. The rhythm of the melody is quantized to eighth-note level and first-order Markov chains for pitch and rhythm are constructed. Secondly, the simple Markov rhythm model is converted into a revised Markov chain model, and the onset probabilities and syncopation index are calculated. Lastly, a user selects one syncopation index among all possible candidates of the syncopation indices which are derived from the second module. The chosen syncopation index determines the initial state in the Markov chain model. After all these procedures, new melody can be played. While music is playing, a user can change syncopation index, which means that the amount of syncopation can be controlled in real-time. When a user selects one of syncopation indices, the changed rhythm starts from the next measure. For the real-time controllable feature, we made the restrictions: first event (note or rest) at onset position 1 in every measure is always the initial state and the last event should fit in the bar even if it is not finished.

Figure 7 shows a screenshot of the Max/MSP patch. This also consists of three modules same as Figure 6. Users can select syncopation index by pressing *bang* button while playing (A video excerpt of the system is available at http://www.bongjunkim.com/work/markov-chain).

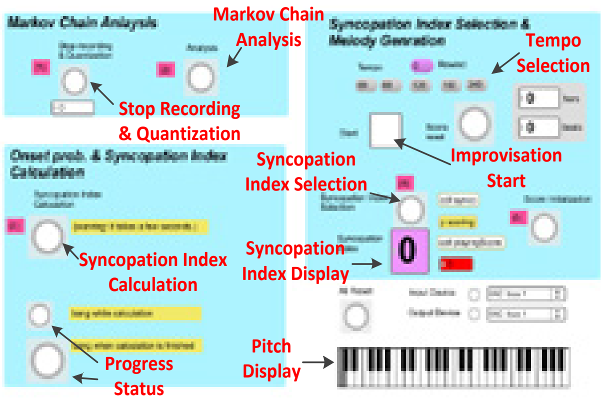

Figure 7. The screenshot of improvisation system built in Max/MSP patch.

5. CONCLUSIONS

This paper presented a probabilistic model for rhythms in Markov chain-based note generation, and an interactive improvisation system was built using Max/MSP. In order to calculate the onset probabilities, a revised Markov model was suggested and the syncopation index in one measure was derived from the onset probabilities at each metrical position. Our model enables users to probabilistically control rhythmic tension when a simple Markov chain generates melodic sequences.

As future works, we need to evaluate our model through users' response to the rhythmic changes and compare with other models. This paper also can be extended for higher-order Markov chain model, and we will continue our research on probabilistic rhythm analysis for algorithmic composition and machine improvisation.

6. REFERENCES

[1] Bischof, M., Conradi, B., Lachenmaier, P., Linde, K., Meier, M., Pötzl, P. and André, El. "Xenakis: combining tangible interaction with probability-based musical composition", *Proceedings of the International Conference on Tangible and embedded interaction,* New York, USA, 2008.

[2] Fitch, W. T. and Rosenfeld, A. J. "Perception and Production of Syncopated Rhythms", *Music Perception*, 25(1): pp. 43-58, 2007.

[3] Gómez, F., Melvin, A., Rappaport, D. and Toussaint, G. "Mathematical Measures of Syncopation", *BRIDGES: Mathematical Connections in Art, Music and Science*, Banff, Canada, 2005.

[4] Kitahara, T., Totani, N., Tokuami, R. and Katayose, H. "BayesianBand: Jam Session System based on Mutual Prediction by user and System", *Proceedings of International Conference on Entertainment Computing,* Pairs, France, 2009.

[5] Klinger, R. and Rudolph, G. "Automatic Composition of Music with Methods of Computational Intelligence", *WSEAS Trans. Information Science and Application*, pp. 508-515, 2007.

[6] Longuet-Higgins, C. and Lee, C. S. "The rhythmic interpretation of monophonic music", *Music Perception*, 1(4): pp. 424-441, 1984.

[7] Nierhaus, G. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer-Verlag, 2009.

[8] Pachet, F. and Roy, P. "Markov constraints: steerable generation of Markov sequences", *Constraints*, 16(2): pp.148–172, 2011.

[9] Pachet, F., Roy, P. and Barbieri, G. "Finite-length Markov processes with constraints", *Proceedings of International Joint Conference on Artificial Intelligence,* Barcelona, Spain, 2011.

[10] Schulze, W. and Van der Merwe, B. "Music generation with Markov models", *IEEE Multimedia*, vol. 18, no. 3, pp. 78-85, 2011.

[11] Sioros, G. and Guedes, C. "Automatic Rhythmic Performance in Max/MSP: the kin.rhythmicator",*Proceedings of International. Conference on New Interfaces for Musical Expression*, Oslo, Norway, 2011.