Generating Music using Markov Chains

Rushikesh Nalla (ID:201401106)*
IT-461 Stochastic Simulation course, DA-IICT

We have generated music using markov chain in which we took an input file containing chords of a song and obtained the transition matrix. We were interested in comparing first, second, third order and combination of all three markov chains and see its effect on the song generation. There were some simplifications made in our implementation. Finally we thought of some extensions which could be incorporated.

INTRODUCTION

From a mathematical viewpoint music is nothing but a sequence of notes. Notes are musical sounds played at a particular pitch. A chord is a collection of notes played simultaneously. A song is actually sequence of chords and there is a transition from one chord to another. Set of chords defines the possible states which our song can be in at any instance of time. It is important to note what distinguishes music from noise. There are certain notes which sound good only when they occur after certain notes and for certain duration.

BACKGROUND, IMPORTANCE AND RELEVANCE

Music is pleasant because musicians understand which notes sound good together. Musical theory lays down certain rules which dictate which notes should be used, how can they be combined and in what sequence should they be used in order to play good music. This means that the transitions are not random (that would be noisy) but rather every state has some definite probabilities of transitioning to others governed by musical aesthetics. Markov chains are a very natural way to model such a process. The problem not only has the required structure but Markov chains will allow us capture the musically pleasant transitions. Remarkably, we were able to mimic subjective musical aesthetics using a mathematical construct.

TECHNIQUE

- The most important thing is to obtain the transition matrix.
- We parse the input file and examine the chords of the song. At every step we obtain the notes of previous chord and present chord. We assume that every note in previous chord transitions to every note in the present chord.
- Accordingly, we keep track of the count of transitions to estimate the probability of state transitions.
- If we use a first order markov chain, we consider only one note from the past as the state.
- Similarly for second and third order markov chain, we consider two and three notes from the past respectively as the state.
- Finally, we use a combination of the three chains (backoff method). Here we start with random notes. We first look at the third order markov chain. If the required state is not present as a state, then we consider second order markov chain. Similarly, if state is not present in second order transition matrix, we consider first order markov chain.

IMPLEMENTATION AND OBSERVATIONS

- We have used Smooth Criminal song by Michael Jackson in our implementation.
- A text file containing the chords of the song was read to generate the transition matrix for different order markov chains.
- There are 12 notes in an octave. The notes are stored as numbers which is from 0-127.
- We have considered only single note transitions because we used only one song to obtain the transition matrix and it becomes very sparse if chord transition is used.
- The duration for which each note was played was set as random between 1-4 sec.
- The velocity(volume) was set to be 255 constant throughout the song.

First order Markov Chain

- In FIG. 1 we obtain a 24*24 transition matrix. Though the total size possible is 128*24, the song only contains some transitions.

- The left column is the previous note and the top row is the present note.
- Each entry in the matrix is the number of times that a note(in column) has transitioned to a particular note(in row). For example the note 59(in column) has transitioned to 57(in row) 52 times in the song we have used.
- There are a lot of transitions between different notes which suggests variations in the music generated.
- There are some values in the matrix which are very high compared to others which implies that such note transitions sound good together.

Creating 1st order markov chain

	61	49	68	57	54
49	1	0	0	0	0
50	0	1	0	0	0
54	0	0	5	2	2
55	0	0	0	16	0
56	0	0	0	0	7
57	5	0	0	117	0
59	8	0	0	52	0
60	0	0	0	2	0
61	27	0	0	1	0
62	5	0	0	26	0

FIG. 1. Transition Matrix of first order Markov Chain

Second order Markov Chain

- Here in FIG. 2 you can see that the left column has two notes which indicates previous two notes and the top row is the present note.
- On comparing to first order matrix, this matrix has smaller values as the probability of transition of particular notes reduces because we need 3 notes to occur together as opposed to 2 notes in earlier case.
- The transition matrix has size 79*24 with maximum possible size of 576*24. This happens because of the song wont contain all possible transitions.
- The next state would contain one previous and one current note but only current note is shown in the matrix so the matrix size is 79*24 and not 79*79 (actual size).

Creating 2nd order markov chain

	57	69	83	81	61
(59, 55)	8	0	0	0	0
(69, 71)	0	6	0	0	0
(83, 83)	0	0	16	10	0
(57, 61)	0	0	0	0	5
(67, 64)	0	0	0	0	0
(84, 84)	0	0	8	0	0
(57, 57)	71	0	0	0	0
(66, 65)	0	0	0	0	0
(66, 64)	0	0	0	0	0
(54, 50)	0	0	0	0	0

FIG. 2. Transition Matrix of second order Markov Chain

Third order Markov Chain

- In FIG. 3 there are 3 entries in the left column containing previous 3 notes and the top row has current note entry.
- The transition matrix size is 137*24 with maximum possible size of 13824*24.
- The next state would contain two previous and one current note but only current note is shown in the matrix so the matrix size is 137*24 and not 137*137 (actual size).
- This matrix is mostly sparse in this case because the probability of occurrence of 4 notes together is very rare in a song.
- Hence when the generated music is played we hear a lot of repetitions.

Creating 3rd order markov chain

	69	57	64	67	62
(69, 68, 64)	3	0	0	0	0
(54, 57, 59)	0	2	0	0	0
(64, 67, 64)	0	0	2	2	0
(57, 57, 64)	0	0	0	0	26
(56, 54, 54)	0	0	0	0	0
(68, 66, 68)	0	0	0	0	0
(54, 68, 69)	0	0	0	0	0
(71, 66, 62)	1	0	0	0	0
(59, 57, 56)	0	0	0	0	0
(83, 81, 64)	0	0	0	0	0

FIG. 3. Transition Matrix of third order Markov Chain

Combination of First, Second and Third Order Markov Chains

- We have used all three transition matrices to generate the song.
- Here we start with 3 random notes. We first look at the third order markov chain. If the required notes is not present as a state, then we consider second order markov chain. Similarly, if the state is not present in second order transition matrix, we consider first order markov chain.
- The song generated becomes non deterministic because of the randomness involved in choosing the initial notes.

POSSIBLE EXTENSIONS

- Generating multi-note (Chord) based music sequences i.e. the transitions matrix is chord based and not note based. This is more realistic as in practice multiple notes are played at the same time.
- Training on more data (songs) and then trying to generate music of different genres like classical, pop etc.
- Classifying the midi file (song) based on the transition matrix we obtain (requires lot of training data).
- Nowadays people are using Neural Networks to generate songs by using training data (songs).

MIDI FILES

- Musical Instrument Digital Interface(MIDI) is a technical standard which allows electronic musical instruments and computers to connect with one another.
- They contain information on how to play the tune unlike audio files. This tune can be played by a variety of computer generated instruments.
- Consists of header chunks and track chunks as seen in FIG. 4
- MIDI files have one header chunk containing information about the instruments playing the sounds, followed by track chunks containing the notes played and their duration, velocities (volume), pitch etc.
- Libraries used were MidiUtils library (to read the MIDI file), wildmidi (for playing midi file).

```
Track 0:
<meta message set_tempo tempo=250000 time=0>
<meta message end_of_track time=0>
Track 1: Song Generated Using Markov Chain
<meta message track_name name=u'Song Generated Using Markov Chain' time=0>
note_on channel=0 note=68 velocity=127 time=0
note_off channel=0 note=68 velocity=127 time=1920
note_on channel=0 note=69 velocity=127 time=0
note_off channel=0 note=69 velocity=127 time=3840
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=3840
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=960
note_on channel=0 note=64 velocity=127 time=0
note_off channel=0 note=64 velocity=127 time=960
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=960
note_on channel=0 note=64 velocity=127 time=0
note_off channel=0 note=64 velocity=127 time=2880
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=2880
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=1920
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=960
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=960
note_on channel=0 note=66 velocity=127 time=0
note_off channel=0 note=66 velocity=127 time=1920
```

FIG. 4. Contents of the MIDI File

CONCLUSION

- The input file (song) contains multi-note chords.
- The file (song) generated have uni-note chords. Hence, the music we obtain is pretty simple compared to the input.
- First order music is very simple but has a lot of variations because it is based on only one previous note.
- In contrast, third order music is complex but is very repetitive because of sparsity of the 3rd order transition matrix.
- Backoff technique is used to remove deterministic behaviour in the system. With this method, any random starting point can be used and the music will still be generated.
- We can get close to the original song by implementing the extensions especially using chord based transitions as mentioned above.

* 201401106@daiict.ac.in

[1] https://medium.com/@omgimanerd/generating-music-using-markov-chains-40c3f3f46405
[2] https://en.wikipedia.org/wiki/MIDI
[3] https://github.com/CMasanto/melody-generator