

Generating Music Using Markov Chains

Stochastic Simulation (IT461) Project

Group Members-

Aditya Joglekar - 201401086

Rajdeep Pinge - 201401103

Rushikesh Nalla - 201401106

Omkar Damle - 201401114

Deep Raiya - 201401221

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.
- Song - Sequence of chords. 'Transition' of one chord into another.
- Set of chords defines the possible states which our song can be in at any instant of time.
- It is also important to note what distinguishes 'music' from 'noise'.



Motivation for use of Markov Chain

- 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.



Motivation for use of Markov Chain

- 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 will be able to mimic subjective musical aesthetics using a mathematical construct.



Technique

- The goal is to obtain a 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.



Technique

- 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.



Implementations

- 1st order markov chain
- 2nd order markov chain
- 3rd order markov chain
- Combination of markov chain using best of the 3 markov chains (Explain)
 - Random Initialization
 - Song generated even when random initialization
 - Different song generated each time. Possibility of seterministic behaviour removed.



Transition matrix: 1st order Markov Chain(24x24)

Creating 1st order markov chain										
	61	49	68	57	54	62	50	59	55	64
49	1	0	0	0	0	0	0	0	0	0
50	0	1	0	0	0	0	0	0	0	0
54	0	0	5	2	2	1	1	0	0	0
55	0	0	0	16	0	0	0	12	0	0
56	0	0	0	0	7	0	0	0	0	0
57	5	0	0	117	0	0	0	42	20	28
59	8	0	0	52	0	0	0	8	8	2
60	0	0	0	2	0	0	0	16	0	6
61	27	0	0	1	0	0	0	16	0	0
62	5	0	0	26	0	0	0	0	0	0

Transition matrix: 2nd order Markov Chain(79x24)

Creating 2nd order markov chain

[illegible]

Transition matrix: 3rd order Markov Chain(137x24)

Creating 3rd order markov chain

	69	57	64	67	62	54	66	61	68	49
(69, 68, 64)	3	0	0	0	0	0	0	0	0	0
(54, 57, 59)	0	2	0	0	0	0	0	0	0	0
(64, 67, 64)	0	0	2	2	0	0	0	0	0	0
(57, 57, 64)	0	0	0	0	26	0	0	0	0	0
(56, 54, 54)	0	0	0	0	0	1	0	0	0	0
(68, 66, 68)	0	0	0	0	0	0	2	0	0	0
(54, 68, 69)	0	0	0	0	0	0	5	0	0	0
(71, 66, 62)	1	0	0	0	0	0	0	4	0	0
(59, 57, 56)	0	0	0	0	0	6	0	0	0	0
(83, 81, 64)	0	0	0	0	0	0	2	0	0	0

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.
- Consist of header chunks and track chunks.
- MIDI files have one header chunk containing information about the “instruments” playing the sounds, followed by track chunks containing the notes played and their durations, velocities (volume), pitch etc.
- Libraries used : MidiUtils library, wildmidi (for playing midi file).



MIDI file example

```
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
```

Observations

- The input files contain multi note chords.
- The files generated are simple uni-note chords. The music is thus pretty simple compared to the input.
- First order music is very simple but has a lot of variations.
- 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.



References

- Theory:
<https://medium.com/@omgimanerd/generating-music-using-markov-chains-40c3f3f46405>

<https://en.wikipedia.org/wiki/MIDI>
- Code for reference (just for help):
<https://github.com/CMasanto/melody-generator>



Demo

We demonstrate the generation of a music MIDI file from input MIDI files using

1. First order MC
2. Second order MC
3. Third order MC
4. Backoff

Compare the music files generated.



Questions?

