Markov Melodies

Santiago Ruiz de Aguirre

April 6, 2020

1 Markov Chains

1.1 Basic Definition

Markov Chains are the discrete version of a Markov process, a common type of *stochastic process*. A sequence of random variables $\{X_t\}_{t \in I}$ that satisfies the Markov property:

Definition: Let $(\Omega, \mathcal{F}, P)$ be a probability space with a filtration $(\mathcal{F}_s, s \in I)$ (with I being a totally ordered index set). Let $(S, \mathcal{S})$ be a measurable space. A $(S, \mathcal{S})$ valued stochastic process $X = \{X_t : \Omega \to S\}_{t \in I}$ adapted to the filtration $\mathcal{F}$ is said to satisfy the Markov Property if for each $A \in \mathcal{S}$ and $s, t \in I$ with $s < t$.

$$P(X_t \in A | \mathcal{F}_s) = P(X_t \in A | X_s)$$

The definition can be simplified in a discrete scenario, having:

$$P(X_n | X_{n-1}, ..., X_0) = P(X_n | X_{n-1})$$

I.e. The Markov property implies that the next state of a process that satisfies it only depends on the current state of that process.

1.2 Transition Matrices

For a continuous Markov Process, transition probability is defined as

$$P(X_t = i | \mathcal{F}_s) = P(X_t | X_s = j) = p_{ij}(t - s)$$

Where p_{ij} is called a transition matrix.

In a finite state space and discrete time, matrix representation is more intuitive, having

$$P(X_{n+1} = j | X_n = i) = p_{ij}$$

1.2.1 Properties

- Since each row represents the probability of moving from state i into state j, it holds that

$$\sum_{j=1}^{S} p_{ij} = 1$$

- Since values represent probabilities, $p_{ij} > 0 \forall i, j$.
- For multi-step transitions, the transition matrix for $P(X_{n+k} = j | X_n = i)$ is P^k.
- Stationary distributions will not be deeply discussed, but are essentially row covectors for P. A stationary distribution π satisfies:

$$\pi P = \pi$$

2 Applications to Music

2.1 Defining a State Space

Let $|\alpha\rangle$ be the Dirac representation of the state of a system. A monophonic melody at some point in time t is in a state

$$|n \in \mathscr{N}, o \in \mathscr{O}, d \in \mathscr{D}\rangle$$

Where n is a note (e.g. C, C#/Db, D, etc.), o is an octave number (e.g. -1, 0, 1, 2, 3, 4, etc.) and d is a duration (e.g. 1, 1/2, 1/4, etc.). For instance if the system is in a $|E, 5, 1/4\rangle$ state, it is playing a quarter-note E5.

Additionally one can define a transformation $T : \mathscr{N} \times \mathscr{O} \to \mathbb{R}$ (if using frequency values, e.g. $(A, 4) \mapsto_T 440$) or another different transformation $T : \mathscr{N} \times \mathscr{O} \to \mathbb{P}, \mathbb{P} \subset \mathbb{N}$ if using MIDI pitches; therefore, states can be expressed as

$$|p, d\rangle$$

With p being a pitch value and d a duration. While it is possible to transform $\mathbb{P} \times \mathscr{D}$, it won't be done and we will assume the system can be separated into two subsystems, one for duration and one for pitch, i.e.

$$|p, d\rangle = |p\rangle_{\mathbb{P}} |d\rangle_{\mathscr{D}}$$

2.2 The Pitch Space

A different pitch space can be set depending on the circumstances. For instant, the state space of the entire MIDI pitch list is the sequence from 0 to 127. This would represent a chromatic scale on all octaves. A C major on one octave with root on C4 would be $\{60, 62, 64, 65, 67, 69, 71, 72\}$. For this particular R library, states were defined as the scale modes for each possible root note from C4 to B7.

2.3 The Duration Space

Durations considered were half notes, quarter notes, eighth notes, sixteenth notes, thirty-second notes, all their dotted versions and round notes and sixty-fourth notes.

2.4 Transition Matrices

Transition matrices in the pitch space in the current model are 22x22 square stochastic matrices. While there are no proper rules on how to define the matrices, it is suggested to give priorities for certain intervals. For the pitch space, the current matrix in the library contemplates standard random sampling (i.e. a rank 1 matrix that has the same distribution in all rows.

3 Scalability

All the code used is open source and hosted on my GitHub page at https://github.com/chnnxyz/markov.melodies. Feel free to extend the code's capabilities.