Markov Models in Python

Abhishek Choubey
Andrea Sofia Vallejo Budziszewski
Computational Music Creativity
February 2023

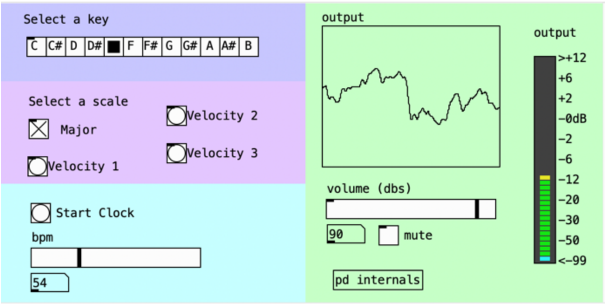
The project aimed to create a tool for generating chord progressions based on any tonic of a major scale. The generated progressions are played via the serum plugin and the user can choose any tonic from the chromatic scale. The tool also allows for the creation of chord velocity, which is controlled by assigning values ranging from one to three.

To create the chord progressions, the team used chord data from two John Mayer songs, "Gravity" and "Who Says." They compiled a list of chord progressions and created a dictionary of all the chords used in these two songs, with each chord mapped to an integer. Since the chord data lacked velocity information, a random velocity value between 1 and 3 was created and paired with each chord. The resulting tuple had the chord integer as the first element and the velocity value as the second. A Markov model was then trained using this data to produce a dictionary of Probability Density Function (PDF) values and unique chords that are played after each chord.

The final output is a list of eight elements, the first of which is the chord value and the second of which is the velocity value. This pattern is repeated to create the complete chord progression. The steering function is then used to change the generation based on the velocity value that has been assigned.

In Pure Data, the data is gathered through the pyext object, which unpacks 8 values representing each chord degree and velocity. The MIDI numbers for each chord are then calculated based on the key chosen by the user. For example, if the key is C, the 1st degree of the scale will get the MIDI numbers 60 (C4), 64 (E4), and 67 (G), which represent the notes used in the C major chord. If the user chooses G as the key, it will add 7 to the MIDI numbers, resulting in 67 (G), 71 (B), and 74 (D), which are the notes used in the G major chord.
The velocities are controlled with a random object to add variation to the chord velocity. Velocity number 1 generates a random number between 60 and 80, velocity number 2 generates a random number between 80 and 100, and velocity number 3 generates a random number between 100 and 127.

Finally, the chord progression and velocity data are sent to the serum plugin to generate sound. To ensure that each chord is played in a steady tempo, the system is designed to play each chord every quarter time.



GUI Instructions:
1. Star the clock and set a BPM.
2. Select a tonic
3. Click major
4. Select a velocity
5. Increase the gain

*if the user doesnt have Serum, they can replace it with a phasor object in pd internals → serum~