# CS221 Project Proposal

Han Lin Aung, Justin Xu, Sicheng Zeng

October 26, 2017

## 1 Input-Output Behavior, Scope, Evaluation Metric

We want to create a music generator, specifically focusing on hip hop and rap beats. Instead of taking on the task of generating whole songs, we'd like to create an AI that will take in famous beats such as "The Real Slim Shady", parse it into different voices and beat lines, and then use those voices to create new, equally as catchy beats.

Our inputs are .midi files and our outputs will be new .midi files. Our target scope is restricted to generation to short, repeatable beats rather than lyrical melodies. To evaluate the beats, we'll listen to them ourselves. A successful generator would be one that fits the different beat lines together well in a way that produces a complete, catchy beat with varied voice lines.

## 2 Preliminary Data

As our preliminary data, we've collected rap .midi files from midiworld.com, such as the file "The Real Slim Shady (Eminem)", saved at http://www.midiworld.com/search/2/?q=rap. Using music21, a toolkit for computer-aided musicology developed at MIT, we split the .midi files into several tracks, each a different instrument or voice part, then randomly combined the tracks to produce output .midi files.

## 3 Baseline and Oracle

Our baseline beat generator uses music21 to separate the beat into different instrument tracks, then takes a random sample of the tracks and combines those back into a new beat. music21 has a function named partitionByInstrument(), and our baseline uses that function to parse the midi file without questioning the results. To combine the tracks, we use another music21 function, insert(). Then we used music21's write() function to output the result to a playable .midi file.

We are taking the role of oracle in our project, since we have the ability to hear the separate tracks easily in our heads, and the know-how to combine them in ways so that the rhythms match up and instruments work together.

The issues with our baseline are that it is combining the different beats randomly, so rhythms aren't matching up, even in a good, syncopated way, and the baseline algorithm could randomly choose to use only base tracks and nothing else, which would just sound muddy and uninteresting. Additionally, music21's partitioning function isn't perfect, and sometimes does not parse the midi tracks correctly.

In order to solve the issues with a baseline and bring it closer to the oracle, we can use, instead of a random selection of tracks, a selection of tracks chosen by a deep neural network, which can 'build intuition' by analyzing how the tracks fit together in the original songs to make smarter choices fitting together different tracks, and we can implement a k-means algorithm to sort the notes into groups by part.

# 4 Related Work

There have been several previous 221 projects with music generation from existing music as their goal. For instance, Kai-Chieh Huang, Quinlan Jung, and Jennifer Lu's "Algorithmic Music Composition using Recurrent Neural Networking" looks at generation of music having learned on a specific composer's music in one genre, and attempting to create music indistinguishable from music that the original composer created.

Chris Pesto's 221 final project was simply titled "Music Generation", and had a more general objective, to generate new music using a convolutional neural network trained on a giant database of music. Pesto's paper states that he also uses music21 to parse his data and uses Google's trained Project Magenta AI to judge whether or not the resulting music is successful.

Yet another 221 final project by Allen Huang and Raymond Wu is titled "Deep Learning for Music", and is focused on creating melodies and harmonies from using only deep neural nets trained on classical music.

Other, non-221 examples of generating music include the previously mentioned Project Magenta, which uses TensorFlow to create art and music, and deepjazz, which is trained on a eight minute long jazz piece to generate similar sounding endless jazz.