221 P-Proposal: iMGM - Improved Music Generation with Magenta

Suraj Heereguppe[a], **Nathan Dalal**[b], **Akash Mahajan**[c]
[a]ICME, Stanford University `hrsuraj@stanford.edu`
[b]CS, Stanford University `nathanhd@stanford.edu`
[c]MS&E, Stanford University, `akashmjn@stanford.edu`

1 Introduction

1.1 Objective

We aim to attempt two problems in music generation. Our first milestone would be generating sequences of coherent melodies, in part drawing upon the work of the Magenta project [1]. The second, longer shot, is to try to extend current research techniques to conditionally generate melodies that play well with a sequence of chords. Our inspiration comes from singing together in an acapella group, and the MUSIC421n research seminar, with the eventual goal to be generating harmonies for existing melodies.

1.2 Baseline, and Oracle

- **Baseline**: Generate random notes and evaluate performance based on whether notes make sense in a series (whether the note intervals are common melodious intervals, whether it sounds good subjectively).
- **Oracle**: Develop a score generation program that takes in a solo MIDI line, an audio track, or the name of a song, and generates a 4-5 part a cappella score for the song. Train and evaluate by understanding structure of existing music, where our score outputs would be the outputs of some generative model.

2 Datasets

For our project we require both MIDI and chord labeled datasets in order to execute our music generation. Here's a list of datasets and their possible applications in our project.

- Lakh MIDI Dataset: The Lakh MIDI Dataset contains 176K+ unique MIDI files, with 45K+ MIDI files matched to specific songs in the Million Song Dataset. We can use this dataset to try to generate unique music based on MIDI patterns. MIDI files are simply a time series of notes in plain text, which makes them far easier to use as training examples over raw audio.
- Reddit Unique MIDI Dataset: Contains 133K+ unique music files scraped from the Internet and sorted by a user of Reddit.
- Piano MIDI: Classical piano MIDI dataset of songs by various composers.
- Nottingham Music Database: Contains 1K Folk Music MIDI files.
- MIREX Audio Chord Estimation: A starting point in terms of data for our stretch goal of chord prediction, a labeled dataset from audio files to chord and beat structures.

[1]Ian Simon and Sageev Oore. "Performance RNN: Generating Music with Expressive Timing and Dynamics." Magenta Blog, 2017. https://magenta.tensorflow.org/performance-rnn

3 Methodology

3.1 Plan of Action

These are a few steps that we see as helping us achieve our realistic goal for the project.

- Our baseline algorithm is a random note generator that picks a random note from a given set of notes. Its performance will be used to gauge the performance of all subsequent models.
- Our next step is to first trivially generate music based on Markov chains learned from our existing MIDI datasets. We'll develop an evaluation metric based whether generated chains have some structure based on the number of major and minor jumps between notes. Modeling this problem as a Markov Decision Process inherently biases the model to generate notes from a restricted search space, the search space being defined by the specific kind of music it is being trained on.
- To overcome the problems of Markov chains, we will then advance to a recurrent strategy to generate melodies, inspired by the Google Magenta Project's recent results on melody generation. There are existing huge datasets for MIDI files and strategies that have worked for Magenta, which we will research and try to replicate or improve upon.
- As a final stretch goal, we would like to implement chord mapping from a MIDI input. Specifically, can we take a series of notes and produce chords in a way that is sufficiently creative and musically correct? And if we have a MIDI solo line for a song, can we turn this into a basic score with chord-based harmonies?

3.2 Evaluation Metrics

Evaluation metrics for this problem are very hard to define objectively because any sequence of notes generated by the algorithm could be pleasing to hear to one person while it may not sound good to another person. However, we aim to partly overcome this issue by defining what is musically correct. Some examples are:

- Major and minor note progressions: Jumps of particular intervals are major while some jumps are minor. In Western music, songs are usually holistically major or minor, so we could define an evaluation that looks for a song with mostly major or mostly minor notes.
- Chord progressions: Chords in Western music appear in several sequences that are common in song construction (examples include (I, V, I) and (I, IV, I)). Evaluation metrics for chords would see which types of chord progressions are occurring (major, minor, suspension, diminished) and whether these line up with the mood of the song (major, minor).

4 Potential Challenges and Conclusion

The main challenge we see to developing our algorithm is defining an evaluation metric so that the algorithm may improve its music generation with every iteration of training. Since, technically, there are no objective measures to defining ”good music”, defining an evaluation metric is the toughest challenge. However, we have enumerated a couple of metrics in the previous section that define what good music should structurally sound like. We believe this is a good starting point to algorithmic music generation.