TRACK EXTENSION AND GENERATION USING MARKOV CHAINS

Colson Demedeiros
University of Victoria
cdemedeiros@uvic.ca

Jae Park
University of Victoria
jaehyunpark@uvic.ca

Oscar Sandford
University of Victoria
oscarsandford@uvic.ca

ABSTRACT

Write this last.

1. INTRODUCTION

Music is typically constructed by humans, for humans. However, machines are becoming more adept at revealing patterns in the way musicians craft their chords. This enables humans to create their own music, and then let the machine take over the task of composer. Our project aims to understand and implement a framework for simple 8-bit music generation through temporal inference techniques. In this paper, we explore previous implementations and existing literature regarding music generation, and present techniques for designing generative Markov models to create derivative musical works.

2. RELATED WORK

2.1 Markov Models

Yanchenko and Mukherjee explored the use of Hidden Markov Models (HMMs) to compose classical music, finding proficiency in generating consonant harmonies, but lacking melodic progression [10]. Indeed, the models were found to learn the harmonic hidden states quite well, in some cases leading to overfitting. HMMs have also found use in chorale harmonization, where a given observable melody uses inference to derive hidden harmonies to complement it [2].

Walter and Van Der Merwe's methods involve representing the chord duration, chord progression, and rhythm progression with first or higher-order Markov chains, whereas the overlaying melodic arc is represented by a HMM [9]. This separation works well to reduce the processing power needed for music generation, but the independent learning of each component leads to less cohesive compositions. Generating music is generally done by sampling a statistical model [4]. However, we want to create music that does not only simply replicate the training data, but also creates cohesive pieces in a more natural way.

Shapiro and Huber's approach to music generation simply uses Markov chains, no hidden states [8]. In their work, the states represent sound objects with attributes such as pitch, octave, and duration. Their results show that human-composed pieces can be closely replicated using the simpler Markov chains. Further, they have attached their implementation in their paper. We will consider this work when constructing our own implementation.

Corrêa and Jungling suggest using Markov chains of different orders to predict classical music. By using stocahsic models to analyze different classcal songs and styles, the computer can even captures subtle and intuitive features such as style of composer. Although this paper focuses on classcial music and its prediction, the authors stress that its applicaition to other mucis genres should be straightforward [5]. The only requirement is a MIDI file with a good quality, as it should be used for markov chains can properly estimate the music's pattern.

2.2 Data Format

The papers previously mentioned use the MIDI file format to write digital music. This format appears to be the standard for digital music creation [6]. One of MIDI's drawbacks is that it cannot store vocals [3]. This is of no concern to us, as we will only be attempting to generate instrumental compositions. Additionally, successful approaches to melody extraction from MIDI files [7] make assurances that this will be an adequate medium for the music our models will generate. MusicXML is a standard file format for storing sheet music, just as MP3 is for recordings [1]. As both are commonly used standards, we plan to use MusicXML for input data and write our output to MIDI files.

3. TIMELINE

Text here, probably a figure too.

4. TASK DELEGATION

In order to figure out the best approach and gather a plethora of sources, we are each looking at various sources related to music data parsing and music generation, from theoretical papers to Python libraries. Colson and Jae are finding classic 8-bit tracks that we will use to train our models on. Oscar has set up a GitHub repository to include written work as well as source code, and made outlines for the final report.

Text

5. RESOURCES

5.1 Tools

Any tools we'll use (e.g. Jupyter Notebooks, MIDI, MusicXML, Python libraries).

Attribution: Colson Demedeiros, Jae Park, Oscar Sandford. "Track Extension and Generation using Markov chains", 21st International Society for Music Information Retrieval Conference, Montréal, Canada, 2020.

5.2 Data Sets

Data sets we'll use (e.g. 8-bit tracks to train/generate from).

6. REFERENCES

[1] Musicxml for exchanging digital sheet music, Jan 2022.

[2] Moray Allan and Christopher KI Williams. Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17:25–32, 2005.

[3] Zehra Cataltepe, Yusuf Yaslan, and Abdullah Sonmez. Music genre classification using midi and audio features. *EURASIP Journal on Advances in Signal Processing*, 2007(1), 2007.

[4] Darrell Conklin. Music generation from statistical models. *Journal of New Music Research*, 45, 06 2003.

[5] Débora C. Correa, Thomas Jungling, and Michael Small. Quantifying the generalization capacity of markov models for melody prediction. *Physica A: Statistical Mechanics and its Applications*, 549:124351, 2020.

[6] Gareth Loy. Musicians make a standard: The midi phenomenon. *Computer Music Journal*, 9(4):8–26, 1985.

[7] G. Ozcan, C. Isikhan, and A. Alpkocak. Melody extraction on midi music files. In *Seventh IEEE International Symposium on Multimedia (ISM'05)*, pages 8 pp.–, 2005.

[8] Ilana Shapiro and Mark Huber. Markov chains for computer music generation. *Journal of Humanistic Mathematics*, 11(2):167–195, Jul 2021.

[9] Andries Van Der Merwe and Walter Schulze. Music generation with markov models. *IEEE MultiMedia*, 18(3):78–85, 2011.

[10] Anna K. Yanchenko and Sayan Mukherjee. Classical music composition using state space models. *CoRR*, abs/1708.03822, 2017.