Creating Harmony Using Human Reinforcement and Machine Learning

Thomas Klimek & Ben Machlin

1 Project Overview

Our project looks to apply the key concepts and algorithms from reinforcement learning to a musical setting. The main goal is to create an agent that is capable of generating harmonies and accompaniment for a solo musician independent of style and genre. Our agent will function by loading in a series of preset chords and harmonies, which are customizable by the musician. Using these harmonies the agent will undergo training by the musician to learn the style and song structure with the goal of ultimately being able to provide dynamic accompaniment in a live setting.

We have three key goals that we hope to achieve with this project:

- Aim 1: Create a working prototype for human reinforced music accompaniment.
  - This goal is not concerned with the quality of the music generated. We simply want to create a modular prototype for generating music, which can later be modified using different state spaces, reward structures, harmonies, instruments, etc…
- Aim 2: Implement an efficient representation for making intelligent musical decisions.
  - How the agent is able to learn is dependent on the representation and model we use for processing the music played the soloist. This goal is concerned with teaching the agent to listen to a musician and make intelligent decisions based on the context of what is being played. What defines a “good” choice in music is highly subjective, however there are a series of rules provided in music theory that explain why certain harmonies produce a pleasant quality, and others not so much. Our goal is to create a model such that we can replicate some of these rules within our agents policies, without directly exposing our agent to the rules of music theory.
- Aim 3: Experiment with different methods of reward signal, and use our agent in a live performance.
  - Our initial goal is to solely use human based rewards, however we may find that our agent can produce better results if we encode some rules for the choices it is allowed to make. We aim to experiment with the effects of different reward structures, as well as the differences made when different musicians train the

system for different genres. Our ultimate goal is to make the system intelligent enough for live performance.

2 Background and Related Work

Reinforcement Learning for Live Musical Agents by Nick Collins

http://composerprogrammer.com/research/rlforlivemusicalagents.pdf

This paper contains load of relevant information for our project. It addresses factors regarding musical agents the use RL. Particularly it addresses the difficulty of creating a reward signal. Unfortunately, it doesn't supply a valuable solution but helps with consideration of what a good reward signal might entail. It also touches on the difficulty of online RL performance. While our project is currently an offline process, we have hopes to turn it into an online, adaptive accompaniment agent. In that case we would follow the examples provided in this paper.

Interactively Shaping Agents via Human Reinforcement: The TAMER Framework by W. Bradley Knox & Peter Stone

http://www.cs.utexas.edu/~sniekum/classes/RLFD-F16/papers/Knox09.pdf

The TAMER framework is a way of replacing environmental reward completely with a human reward signal. It uses the concept of shaping to train an agent by providing it was solely positive and negative feedback from a human. Because of the lack of a correct environmental reward signal in our project, using TAMER allows us to teach the agent with a non-trivial reward signal. It also allows us to combine human reward signal with other experimental reward signals or even other human signals (imagine two humans simultaneously rewarding a single agent).

3 Problem Formulation and Technical Approach

In basic terms, the RL agent will receive melodies and will output chords. A single input will be 16 notes (or silence/rest) representing one bar of music. However, to account for every possible bar of music would require billions and billions of possible inputs and states. To simplify the input space, we will encode each bar by counting the occurrence of each note in the bar. Figure 1 would be encoded as A3B2C2D2E2F2G2R1.

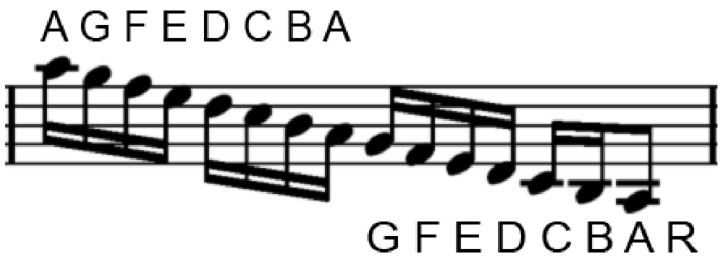

Figure 1

We will use this encoding as our state. The action taken by the agent is selecting from a bank of possible chords. This bank will consist of major and minor chords for each chromatic note. That is: C-major, C-minor, C#-major, C#-minor, D-major, etc. This bank can easily be extended to include augmented, diminished, inverted, user-specified, and many other types of chords, but we will use the 24 major and minor chords to start with a limited action space.The reward will be distributed according to a TAMER [2] mechanism that allows a human to listen to the agent's chord selection played along with the bar it corresponds to and provide positive or negative feedback.

To facilitate ease of use of our project, the input bars will be in Musical Instrument Digital Interface (MIDI) format. This is a common output format for most digital instruments and is relatively easy to generate both algorithmically and manually. Our program will then interpret each bar and encode it into with our note counting method. One cycle of learning consists of the following steps:

1. Decode a bar of MIDI input (state)
2. Agent selects a chord (action) for the given input
3. Playback the chord along with the original MIDI input bar
4. Human gives feedback on how well the chord fits the input melody
5. Feedback is summed and provided to the agent as a reward for picking the chord given the input (state)
6. The agent updates its internal values

This algorithm then repeats using the next bar of the MIDI input as the next state. The learn algorithm we plan to use is Q-Learning as described in Sutton and Barto [1].

Our state space is still significantly large and it would be infeasible to use a tabular RL method. Additionally, because we want similar melodies to reinforce similar chord choices, we want to use function approximation to generalize reinforcement of specific actions in specific states to similar states. For example if choosing C-major given the state/input C6E4G5R1 yields very high reward, we want C-major to be an appealing action given the new state/input C3E3G3R7. In this case, the count of each note in the bar is our feature set.

4 Evaluation and Expected Outcomes

We plan to use three main metrics to evaluate the performance of our agent. The first is a theoretical evaluation of the chords produced by our agent. We will use rules of music theory to deduce how correct or incorrect the choices were and introduce a metric for scoring a decision. The score will be related to how closely the chord correlates to the notes found in the melody played by the musician.

The second method of evaluation will be to see if our agent can learn an “optimal policy” for a well known song, essentially asking whether it can learn the real chords to Hotel California, or a number of well known songs across different genres. We will perform training using preset melodies for these songs and experiment with different representations and learning algorithms to see if we can learn the correct chords to the song and how long this will actually take.

The final metric will be a satisfaction metric of the user. Where the first two metrics may vary with regards to the musical expertise of the musician training the agent, satisfaction will be independent of musical knowledge. We plan to conduct the training of our agent with a few musicians of varying skill level and gauge there satisfaction with its performance. This evaluates the human reinforcement portion of our agent, and also provides insight into the different policies our agent is capable of learning dependant on the person using the algorithm.

References

[1] Richard S. Sutton and Andrew G. Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[2] W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.