Automatic Music Generator

Project Proposal

Sam Fleckenstein (sef44)
Ross Nanopoulos (rdn21)

February 7, 2014

Contents

1 Abstract

To develop an intelligent music composer that will analyze common and popular patterns in music, reason about those patterns, and generate a new piece of music that is significantly different than the analyzed pieces, while still being interesting.

2 Background

What is the process by which humans make music? They study the fundamentals: beats, measures, time and key signatures, tempo, rhythm. They listen to great composers: Bach, Tchaichovsky, Mahler, Debussy, Chopin. Somehow this knowledge combined with creativity yields additional, masterful compositions. How then, does one enable a computer to exhibit this thoughtful creativity?
A variety of methods have been proposed for algorithmic compositon including hidden markov models, genetic algorithms, neural networks, or a combination thereof.

3 Intended Audience

Researchers who would like to incorporate different learning techniques for algorithmic composition will be able to easily integrate with this system.

4 Requirements

4.1 Song Parser

The backbone of the song parser will be The Echo Nest's large database of music intelligence. The song parser will utilize The Echo Nest's API to extract useful song information from the database, which includes a plethora of aspects including time signature, key, mode, tempo, loudness, duration, end of fade in, start of fade out, audio fingerprints, timbre, pitch, and loudness. Additionally, The Echo Nest provies sequenced data as ”musically relevant elements” that include segments, tatums, beats, bars, and sections. This information will allow the learning agent to discern the myriad dynamics of songs and learn about the ways in which different songs are composed.

4.2 Learning Agent

The job of the learning agent will be to take the raw music data gathered by the song parser and discover the relevant patterns in the music. There are a number of different algorithms that could be used to achieve this goal, but the current plan is to use a hidden Markov model to extract these patterns. This model was chosen because it can be used to represent processes where not all of the information about a state is know. This is useful because music is very complex and it is very difficult to determine every variable that goes into determining what should come next in a song. Another reason that hidden Markov models were chosen for this project is because they have been successfully applied the automatic generation of music.

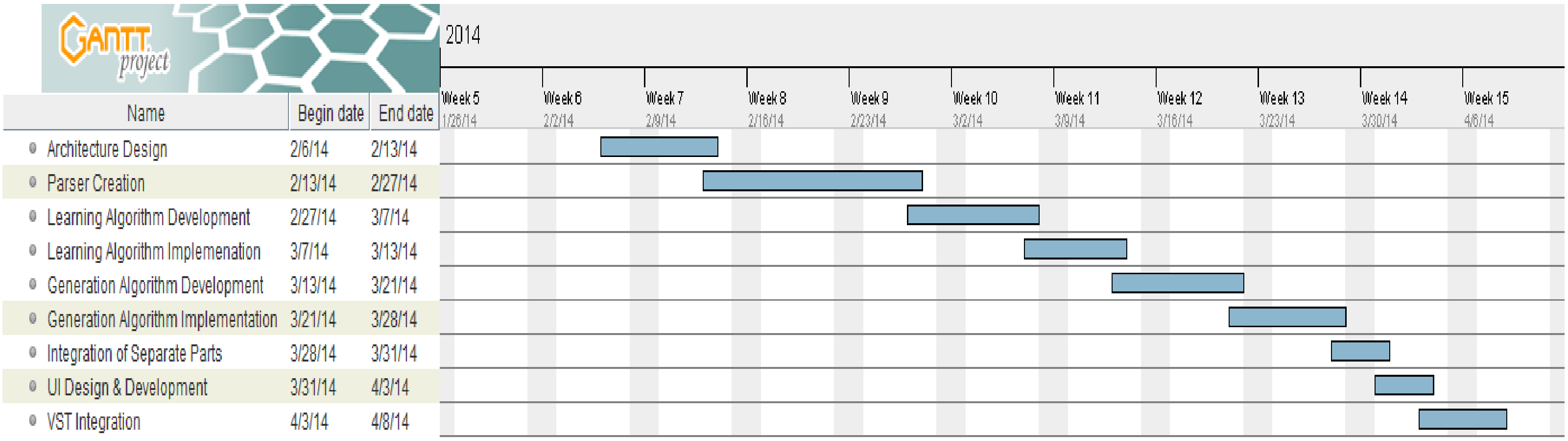
The learning agent is to be designed and implemented in such a way as to allow it to be easily swapped out for a different algorithm. This has several benefits. First, it will be easy to implement a completely different learning algorithm if one does not produce high enough quality results. Second, this will allow for easier testing of the various components of the project. It will be easy to set up dummy functions in the learning agent to accept data from the song parser in order to verify that the parsing is happening correctly. Dummy functions can also be used to pass specific patterns to the composition agent in order to verify that the patterns are applied correctly.

4.3 Composition Agent

The composition agent will take the information that the learning agent provides and decide which notes, patterns, rhthyms, etc. to incoproate into its own piece of music. It will then be responsible for outputting this generated music to a .wav file, which can be used later for further analysis.

5 Project Management

Below is a Gantt Chart, which will drive our development process:



Name	Begin date	End date
Architecture Design	2/6/14	2/13/14
Parser Creation	2/13/14	2/27/14
Learning Algorithm Development	2/27/14	3/7/14
Learning Algorithm Implemenation	3/7/14	3/13/14
Generation Algorithm Development	3/13/14	3/21/14
Generation Algorithm Implementation	3/21/14	3/28/14
Integration of Separate Parts	3/28/14	3/31/14
UI Design & Development	3/31/14	4/3/14
VST Integration	4/3/14	4/8/14

5.1 Communication

In order to facilitate on-time delivery of the Intelligent Music Generator, in-person meetings will be held at least once at week on Thursdays at 4:30. In addition to this, meetings will be held as necessary to discuss upcoming deadlines as well as any issues that have come up. Communication will also happen during the rest of the week primarily via email.

5.2 Source Control

Github will be used for feature tracking and reporting bugs. Pull requests will be utilized to ensure that each member has reviewed the code before it enters the master branch. Branches will be utilized for implementing different components and features.

5.3 Work Division

The primary responsibilities of each of the project members are as follows. Sam will be responsible for the primary development and implementation of the learning algorithms. Ross will be in charge of the implementation of the rest of the project. This is not a hard division of the work as each of the group members will also be working a great deal on the parts of the project they are not in charge of. This division of work fits well with the strengths and experience of each of the project members.

6 Completed Work

The work that is completed so far has been mostly research based. Various learning techniques for the learning agent have been examined before settling on using a Hidden Markov model. This model was chosen for several reasons. First, because it has been applied to various other projects with similar goals to this one. Second, because it appears to complex enough to produce interesting music, but simple enough that it will be possible to complete the project in the given timeline. In addition, pseudo-code has been written for the learning agent in order to more fully understand what will be required to implement the agent.

7 Wishlist Features

There are a number of features that will be included in the project if the required components are completed in time. First is some sort of user friendly UI. Currently, the only planned way of interacting with the program is through the command line. Since this method is not particularly user friendly, a UI with user controls may be added.

Another feature that may be added is a different learning algorithm. If there is time, something like a neural network may be added to do the learning, instead of just using a hidden Markov model.

A third feature that will be added if there is time is the ability to output the music to a Virtual Studio Technology. This would allow songs to be created that have more than one instrument. It would also allow the user or program to select what instrument they would like to use.

8 Software Requirements

8.1 Song Parser

The song parser will contain the interface for interacting with The Echo Nest's API and database. It will make a call using the API determined by user input. It will be fed a seed note, chord, key, or other attributes from which the parser can locate and return relevant songs. For example, the song parser could be passed in a genre of rock, with a seed note of "A", a time signature of 4/4, and a tempo of 180-200 beats per minute. This will allow the calls to The Echo Nest's API to be more specific, so that the other parts of the automatic music generator will have more specific data to reason with. The Echo Nest's API returns JSON objects, that can be used by the learning agent in its reasoning.

8.2 Learning Agent

8.2.1 Internal Architecture

The learning agent will have two main components. The first part will be the algorithm responsible for extracting patterns from the music. It will work by constructing a model of how it perceives a model of the music to be structured. This includes a representation of the observed outputs (pitches, note durations, chord progressions, key changes, etc) of the hidden states of the musical model. After receiving these outputs from the song parser, this algorithm will find the most useful and interesting patterns and output them.

In order for the learning algorithm to work properly, it needs to be provided with estimated values of the hidden parameters in the music (i.e. the probabilities of transitioning between the hidden states). These values will come from a second algorithm, namely the Baum-Welch algorithm, which works by finding the maximum likelihood estimate of the hidden parameters, given the observed data. That is, it finds the values for the transition parameters that make the observed data most likely.

8.2.2 Parser to Learning Agent Interface

In order to make the learning agent as modular as possible, the interface between it and the song parser will be kept very small, and will consist of a single call to the parser to get the raw song data. This will make it easier to swap out the learning algorithm if the need arises.

8.2.3 Learning Agent to Composition Agent Interface

Similar to the interface between the parser and the learning agent, this interface between the learning agent and the composition agent will be kept as small as possible. The only interaction between these two components will be when the learning agent passes the composition agent a file containing a representation of all of the relevant patterns that it found in the music.

8.3 Composition Agent

The composition agent will take the relevant patterns provided by the learning agent and decide which patterns should be utilized in its composition. It will consist of an algorithm to decide which patterns will be chosen, as well as an actual component to write a .wav file to disk.

Automatic Music Generator

Project Concept

Sam Fleckenstein and Ross Nanopoulos

January 18, 2014

1 Background

Musicians have been composing music for centuries by first studying music and incorporating certain identifying aspects into their own pieces. One current problem in the field of artificial intelligence is to mimic human creativity in subjects including, but not limited to linguistics, visual fields, and music. Because computers are inherently deterministic, generated music tends to be repetitive and, in many cases, rather boring. This poses an especially challenging problem.

2 Scope

The goal of this project is to create a music creation application, which will intelligently analyze an existing body of music, in order to create new music based on what it has learned. The music generated is not expected to be of extremely high quality, but it should at the very least be interesting. This new music should contain enough attributes found in the body of music it is based on, in order to be identified as belonging to the same genre.

3 Methodology

The music generator will compile statistics from a large volume of music. It will look at a variety of attributes including:

- Beats and Rhythms
- Key/Time Signatures
- Pitches
- Chord progressions
- Note patterns
- Dynamics, especially changes during a single song

The two main components of the program include: 1.) the learning agent that will be responsible for listening for, and identifying relevant attributes in the music, and 2.) the generating agent that will be responsible for taking the attributes learned and applying them to create new music.