Automatic Music Generator
Progress Report 2

Sam Fleckenstein (sef44)
Ross Nanopoulos (rdn21)

March 21, 2014

Contents

1 Abstract **3**

2 Introduction **4**

3 Application **4**
3.1 The Echo Nest Interface 4
3.2 Learning Agent 4
3.3 Composition Agent 5

4 Methodology **5**
4.1 The Echo Nest Interface 5
4.2 Learning Agent 5

5 Software Design **5**
5.1 User Interface 5
5.2 The Echo Nest Interface 5
5.3 User Feedback 5
5.4 Learning Agent 6
5.5 Composition Agent 6

6 Project Management **6**
6.1 Communication 6
6.2 Source Control 6
6.3 Work Division 6
6.4 Management Plan Effectiveness 7
6.5 Now to the End 7

7 User Interface **7**

8 Testing and Evaluation **9**

9 Project Progress **9**
9.1 The Echo Nest Interface 9
9.2 Learning Agent 9
9.3 Composition Agent 9
9.4 User Interface 9

10 Discussion **9**

11 Appendices **10**
11.1 Database Design 10
11.2 User Manual 10
11.3 Programmer Manual 10

1 Abstract

The purpose of this project is 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 Introduction

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 composition including hidden Markov models [4], genetic algorithms [3], and neural networks [7]. Additionally, a field known as "combination theory" has combined these methods to create more advanced learning and composition algorithms [5]. Hidden Markov Models utilize an element of probability and uncertainty that can lead to much more interesting compositions The argument can be made that innate creativity plays a large role in being able to compose interest- ing music. However, a goal of artificial intelligence is to eventually develop systems that can think and have personalities of their own. Thus, this innate creativity when composing music will develop with more advanced artificially intelligent systems that can think for themselves and exhibit such behavior.

3 Application

3.1 The Echo Nest Interface

The backbone of the Echo Nest interface will be The Echo Nests large database of music intelligence. The song parser will utilize The Echo Nests 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 fingerprint, timbre, pitch, and loudness. Additionally, The Echo Nest provides 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.

3.2 Learning Agent

The job of the learning agent will be to take the raw music data gathered by Echo Nest and discover the relevant patterns in the music. There are a number of different algorithms that could be used to achieve this goal, but this project uses a hierarchical hidden Markov model to extract these patterns. A hierarchical hidden Markov model is a generalization of hidden Markov models, where each state is itself a hidden Markov Model. [2] This allows the learning agent to utilize the inherent structure of music to provide a more informed model.

Additionally, hierarchical hidden Markov models have been used to look at pitch structure in music. [6] In relation to this project, the learning agent looks at the patterns within the higher structure of the music in the Echo Nest's sections (i.e. patterns of choruses, verses, bridges, etc.). Then it moves deeper into the next level of musical events, bars, and continues this process all the way down to the level of notes, where the respective hidden Markov model will learn the actual sequence of pitches and duration of these notes.

This model was chosen because it can be used to represent processes where not all of the information about a state is known. 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 [4]. The complexity of a hidden Markov model is also very easy to expand. This can be done by looking at data that is farther in the past from the current observation, or by adding in more variables to the states being considered. Finally, the hierarchical structure of hidden Markov models allows the learning agent to best utilize the structure of the Echo Nest's data, which has the many levels of structure inherent to music.

3.3 Composition Agent

The composition agent will take the information that the learning agent provides and decide which notes, patterns, rhythms, etc. to incorporate 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 and classification.

4 Methodology

4.1 The Echo Nest Interface

This interface utilizes pyechonest, a python wrapper for The Echo Nest's Main API, in order to collect an audio summary, which contains the basic information for a song such as the key, mode, tempo, time signature, and an analysis URL (i.e. where the sequenced data for each song lives). The sequenced datea can be easily accessed via The Echo Nest's Remix API; therefore, this interface only needs to pass a list of track IDs to the learning agent.

4.2 Learning Agent

The learning agent utilizes the GHMM library to perform the required machine learning on the patterns of sequenced data for which the user rates highly. A hidden Markov model was created for each level of musical events, as specified by The Echo Nest's API. These musical events include, from largest to smallest: sections, bars, beats, tatums, and segments. Currently, the learning agent learns from all of the musical events separately; however, the next phase of development will focus on recursively moving down the structure of the music, and then reconstructing the song from the bottom upwards.

5 Software Design

5.1 User Interface

1. The UI will prompt the user for a musical genre (COMPLETED)
2. The UI will prompt the user for a song tempo (COMPLETED)
3. The UI will prompt the user for a time signature (COMPLETED)
4. The UI will prompt the user for a key signature (COMPLETED)
5. The UI will send user choices to the Echo Nest Interface (COMPLETED)

5.2 The Echo Nest Interface

1. The Echo Nest Interface will take as input the user input from the UI (COMPLETED)
2. The Echo Nest Interface will make a call to The Echo Nest API using the user input (COMPLETED)
3. The Echo nest will provide a list of track IDs to be utilized by the learning agent (COMPLETED)

5.3 User Feedback

1. The user will be prompted to rate a number of short song clips (IN PROGRESS)
2. The user ratings will be tabulated and used to update the learned model based on which musical patterns the user rated highest
3. The user will be presented with one longer song based on the updated musical model

4. The user feedback will be injected to the relevant stages of the hierarchical hidden Markov model to improve the learning

5.4 Learning Agent

1. The learning agent will use the GHMM library
2. The learning agent will take as input track IDs from the Echo Nest interface (COMPLETED)
3. The learning agent will utilize The Echo Nest's Remix API [1] to access sequenced data from track IDs (COMPLETED)
4. The learning agent will train a model using the GHMM library and the input from the Echo Nest Interface
5. The learning agent will output a trained model to the composition agent

5.5 Composition Agent

1. The composition agent will take as input a trained model from the learning agent (COMPLETED)
2. The composition agent will create a sequence of sections from the model
3. The composition agent will create a sequence of bars from the model as it relates to sections
4. The composition agent will create a sequence of beats from the model as it relates to bars
5. The composition agent will create a sequence of tatums from the model as it relates to beats
6. The composition agent will create a sequence of segments as it relates to tatums
7. The composition agent will write the resulting composition to disk as a .wav file (IN PROGRESS)

6 Project Management

6.1 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.

6.2 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.

6.3 Work Division

The work division will be as follows. Ross will be reponsible for the primary development and implementaiton of the HHMM in the learning agent. Sam will be responsible for the design and implementation of the composition agent. Both members will utilize the various APIs provided by The Echo Nest to complete their tasks. 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.

6.4 Management Plan Effectiveness

The largest issue with the management plan has been changes to the learning agent. These changes have necessitated a pushback of the timeline for the learning agent. This has not been too large of an issue because other work that does not rely on the learning agent has continued while the learning agent has been delayed. Given this, the project should still complete on time.

The communication method has been an effective way of keeping on track to complete assigned tasks. It has also allowed questions to be asked more frequently, so that getting stuck on one particular challenge for too long is not an issue.

In regards to version control, when the project started, merging commits were a problem due to constantly changing and moving parts. This has been resolved, however, and merging has become a much smoother process.

As such, the project management plan will be kept largely the same, except for shifting around of when various parts of the project will occur.

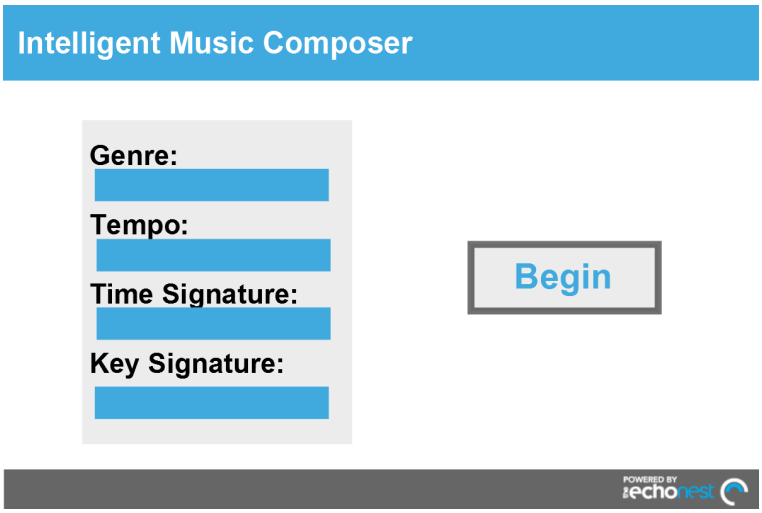
6.5 Now to the End

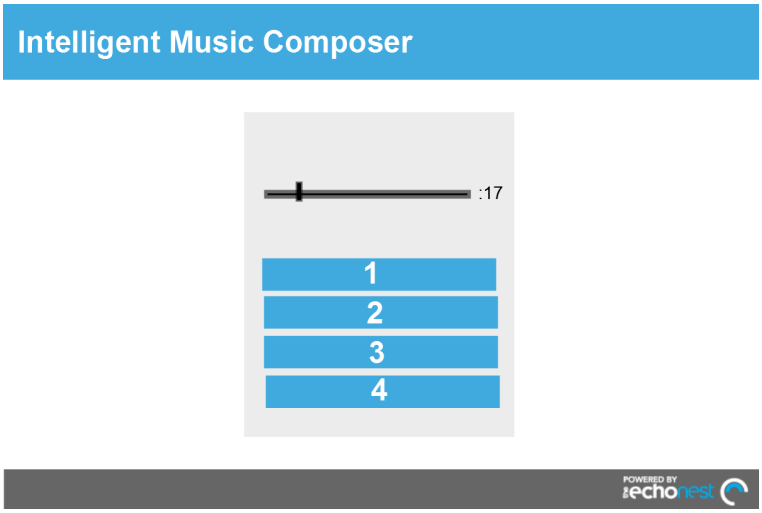
Given the changes that have been made to the learning agent, timelines have been pushed back. The project should still finish on time, but it may not be as well polished as is desirable.
During the next stage of the project, the focus will be shifting towards completing the learning agent, and towards implementing the composition agent. Since various components of the project are getting finished up, they will also be hooked together to form a more functional product.

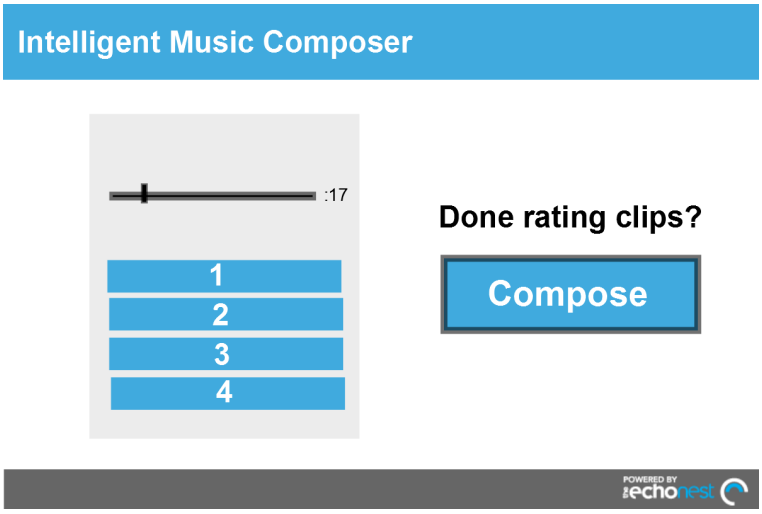
Lastly, much of the code is currently repetitive between the different musical event learners. There is discussion about creating a hidden Markov model state factory class that will be able to be passed the type of musical event (section, beat, segment, etc.), the alphabet for that type (chorus verse bridge, a e a d, etc.), as well as the respective transmission and emission matrices. This factory will create the hidden Markov model states that will be utilized by the learning agent.

7 User Interface

The user interface will be implemented as a web application. First the user will be prompted to choose a genre, tempo, time signature, and key signature. The user will click begin, and will be played short sound clips, which can be rated 1 to 4 (this will be updated to words, once it is decided which will provide a better user experience). After the default number of music clips, the user can choose to rate more music clips or compose a song. Below are screen shots of the application flow:


Intelligent Music Composer
Genre:
Tempo:
Time Signature:
Key Signature:
Begin
POWERED BY echonest


Intelligent Music Composer
:17
1
2
3
4
POWERED BY echonest


Intelligent Music Composer
:17
1
2
3
4
Done rating clips?
Compose
POWERED BY echonest

8 Testing and Evaluation

The testing is currently all manual. Each of the developers is responsible for testing all of their components to make sure that they work properly, but the present focus of the project is the implementation of all required features. Once all of the desired features are implemented, a more thorough testing procedure will be implemented to ensure the music generator works as designed.

9 Project Progress

9.1 The Echo Nest Interface

The Echo Nest interface is completely finished. It is capabale of making a call to the Echo Nest's API using the user input and returning a list of song IDs, which the learning agent can utilize when it gathers user preferences in the music clip rating portion of the application.

9.2 Learning Agent

Given the changes that have been made to the plan for the learning agent, it is now behind schedule. It takes in song IDs from the EN interface, and gathers relevant information about the songs.
Because we are now utilizing the hierarchical hidden Markov models, the following classes have been created: SectionLearner, BarLearner, BeatLearner, TatumLearner, and SegmentLearner. These will be able to be treated as states for our top-level LearningAgent, thus producing a hidden Markov model at each level of structure within the song.

9.3 Composition Agent

The composition agent has not been started, and will be the primary focus of the next phase in development.

9.4 User Interface

A basic, but functional user interface has been created that takes in genre, tempo, key signature, and time signature from the user and passes those parameters to the Echo Nest interface. The user is then redirected to a page where she will provide ratings of songs that the learning agent can utilize in its learning. Some basic templating has been performed, in order to prepare for CSS styling. Basic CSS has also been implemented. Views still need to be created for the composition and results pages.

10 Discussion

Given that the project was called unoriginal and uninnovative during its last review, changes have been made to the learning model that will be used. The model has been expanded from a single hidden Markov model to a hidden Markov model of hidden Markov models, a so-called hierarchical hidden Markov model. This expansion allows for better utilization of the structure of The Echo Nest's data, as well as the inherent structure of music in general. Now, instead of just piecing together separate patterns that may come from random parts of songs, the model will be able to take a pattern of higher level patterns such as choruses and verses and then explore deeper into the patterns that make up those patterns such as bars, beats, and notes. Therefore each level of patterns will be dependent on the patterns that came before it. For example, segments will depend on tatums, which will depend on beats, and so on. This will provide a more cohesive song structure that will combine better patterns that relate to each other.

11 Appendices

11.1 Database Design

As there is not a large quantity of data to be saved for this project, there is no associated database.

11.2 User Manual

Given the incomplete nature of the project at this point, a user manual is not appropriate.

11.3 Programmer Manual

Given the incomplete nature of the project at this point, a programmer manual is not appropriate.

References

[1] Remix api. http://echonest.github.io/remix/apidocs/.

[2] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Mach. Learn.*, 32(1):41–62, July 1998.

[3] B. Matityaho and M. Furst. Neural network based model for classification of music type. In *Electrical and Electronics Engineers in Israel, 1995., Eighteenth Convention of*, pages 4.3.4/1–4.3.4/5, March 1995.

[4] W. Schulze and B. van der Merwe. Music generation with markov models. *MultiMedia, IEEE*, 18(3):78–85, March 2011.

[5] P. Sheikholharam and M. Teshnehlab. Music composition using combination of genetic algorithms and recurrent neural networks. In *HIS '08. Eighth International Conference on Hybrid Intelligent Systems*, pages 350–355, Sept 2008.

[6] Michele Weiland, Alan Smaill, and Peter Nelson. Learning musical pitch structures with hierarchical hidden markov models.

[7] K. Youssef and Peng-Yung Woo. Music note recognition based on neural networks. In *ICNC '08. Fourth International Conference on Natural Computation*, volume 2, pages 474–478, Oct 2008.

Automatic Music Generator

Progress Report 1

Sam Fleckenstein (sef44)
Ross Nanopoulos (rdn21)

March 3, 2014

Contents

1 Abstract **3**

2 Introduction **4**

3 Application **4**
3.1 The Echo Nest Interface 4
3.2 Learning Agent 4
3.3 Composition Agent 4

4 Methodology **4**
4.1 The Echo Nest Interface 4
4.2 Learning Agent 5

5 Software Design **5**
5.1 User Interface 5
5.2 The Echo Nest Interface 5
5.3 User Feedback 5
5.4 Learning Agent 5
5.5 Composition Agent 6

6 Project Management **6**
6.1 Communication 6
6.2 Source Control 6
6.3 Work Division 6
6.4 Management Plan Effectiveness 6
6.5 Now to the End 6

7 User Interface **7**

8 Testing and Evaluation **8**

9 Project Progress **8**
9.1 The Echo Nest Interface 8
9.2 Learning Agent 8
9.3 Composition Agent 8
9.4 User Interface 9

10 Discussion **9**

11 Appendices **10**
11.1 Database Design 10
11.2 User Manual 10
11.3 Programmer Manual 10

1 Abstract

The purpose of this project is 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 Introduction

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 composition including hidden Markov models [4], genetic algorithms [3], and neural networks [6]. Additionally, a field known as "combination theory" has combined these methods to create more advanced learning and composition algorithms [5]. Hidden Markov Models utilize an element of probability and uncertainty that can lead to much more interesting compositions The argument can be made that innate creativity plays a large role in being able to compose interest- ing music. However, a goal of artificial intelligence is to eventually develop systems that can think and have personalities of their own. Thus, this innate creativity when composing music will develop with more advanced artificially intelligent systems that can think for themselves and exhibit such behavior.

3 Application

3.1 The Echo Nest Interface

The backbone of the Echo Nest interface will be The Echo Nests large database of music intelligence. The song parser will utilize The Echo Nests 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 fingerprint, timbre, pitch, and loudness. Additionally, The Echo Nest provides 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.

3.2 Learning Agent

The job of the learning agent will be to take the raw music data gathered by Echo Nest and discover the relevant patterns in the music. There are a number of different algorithms that could be used to achieve this goal, but this project 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 known. 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 [4]. The complexity of a hidden Markov model is also very easy to expand. This can be done by looking at data that is farther in the past from the current observation, or by adding in more variables to the states you are considering

3.3 Composition Agent

The composition agent will take the information that the learning agent provides and decide which notes, patterns, rhythms, etc. to incorporate 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 and classification.

4 Methodology

4.1 The Echo Nest Interface

This interface utilizes pyechonest, a python wrapper for The Echo Nest's Main API, in order to collect an audio summary, which contains the basic information for a song such as the key, mode, tempo, time

signature, and an analysis URL (i.e. where the sequenced data for each song lives). The sequenced datea can be easily accessed via The Echo Nest's Remix API; therefore, this interface only needs to pass a list of track IDs to the learning agent.

4.2 Learning Agent

The learning agent will use the GHMM library [1] to perform the required machine learning on the patterns within the sequenced data for which a user rates highly. This library was chosen because it offers all of the features needed to perform the required learning task. It is also free, which is another important feature.

5 Software Design

5.1 User Interface

1. The UI will prompt the user for a musical genre (COMPLETED)
2. The UI will prompt the user for a song tempo (COMPLETED)
3. The UI will prompt the user for a time signature (COMPLETED)
4. The UI will prompt the user for a key signature (COMPLETED)
5. The UI will send user choices to the Echo Nest Interface (COMPLETED)

5.2 The Echo Nest Interface

1. The Echo Nest Interface will take as input the user input from the UI (COMPLETED)
2. The Echo Nest Interface will make a call to The Echo Nest API using the user input (COMPLETED)
3. The Echo nest will provide a list of track IDs to be utilized by the learning agent (COMPLETED)

5.3 User Feedback

1. The user will be prompted to rate a number of short song clips (IN PROGRESS)
2. The user ratings will be tabulated and used to update the learned model based on which musical patterns the user rated highest
3. The user will be presented with one longer song based on the updated musical model

5.4 Learning Agent

1. The learning agent will use the GHMM library (COMPLETED)
2. The learning agent will take as input track IDs from the Echo Nest interface
3. The learning agent will utilize The Echo Nest's Remix API [2] to access sequenced data from track IDs
4. The learning agent will train a model [1] using the GHMM library and the input from the Echo Nest Interface (COMPLETED)
5. The learning agent will output a trained model to the composition agent (COMPLETED)

5.5 Composition Agent

1. The composition agent will take as input a trained model from the learning agent (COMPLETED)
2. The composition agent will create an overarching chord progression
3. The composition agent will then fill in notes using the trained model
4. The composition agent will write the resulting composition to disk as a .wav file (IN PROGRESS)

6 Project Management

6.1 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.

6.2 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.

6.3 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 interface to The Echo Nest's API, as well as the development and implementation of the composition algorithms. 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.

6.4 Management Plan Effectiveness

The management plan set out in the project proposal has turned out to be very effective. The work of each of the team members has been neatly separated, allowing each member to progress at their own pace without having to rely on unfinished work from the other team member.

The communication method has been an effective way of keeping on track to complete assigned tasks. It has also allowed questions to be asked more frequently, so that getting stuck on one particular challenge for too long is not an issue.

In regards to version control, when the project started, merging commits were a problem due to constantly changing and moving parts. This has been resolved, however, and merging has become a much smoother process.

As such, the project management plan will be kept the same.

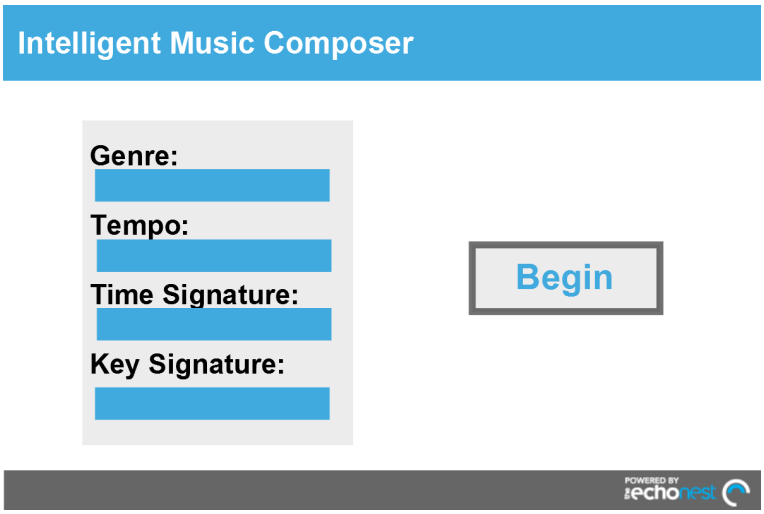
6.5 Now to the End

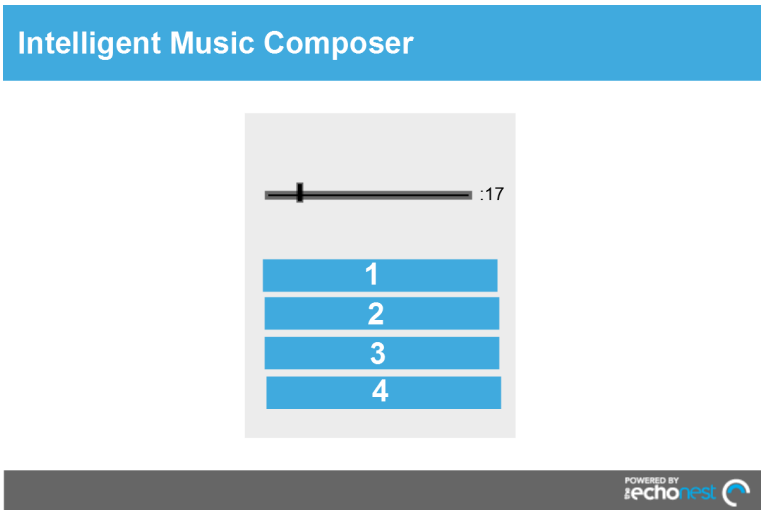
Given the speed of progression, all goals set out in the previous documents should be achievable by the end of the semester. The wishlist feature of implementing a different learning algorithm within the learning agent will likely not happen due to the amount of research that would be required in order to create and implement another algorithm to do the same sort of work as GHMM; however, clustering algorithms are currently being

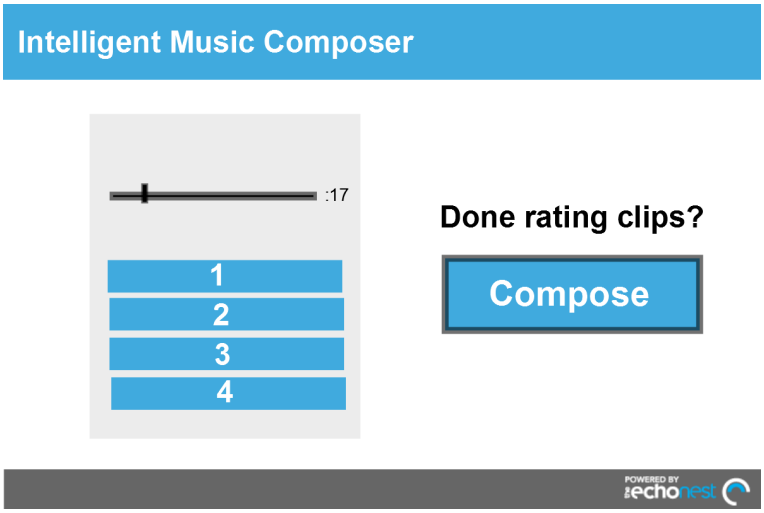
researched, in order to generate the sound clips for the user feedback. Once clusters are identified and classified, they can provide useful information to the learning agent by making the user feedback more meaningful. The status of the wishlist feature of adding in the ability to output music to a virtual studio technology is that it is currently unlikely to happen, but depending on the ease-of-use of pre-existing libraries to assist in this, could still be implemented.

7 User Interface

The user interface will be implemented as a basic web application. First the user will be prompted to choose a genre, tempo, time signature, and key signature. The user will click begin, and will be played short sound clips, which can be rated 1 to 4 (this will be updated to words, once it is decided which will provide a better user experience). After the default number of music clips, the user can choose to rate more music clips or compose a song. Below are screen shots of the application flow:







8 Testing and Evaluation

The testing is currently all manual. Each of the developers is responsible for testing all of their components to make sure that they work properly, but the present focus of the project is the implementation of all required features. Once all of the desired features are implemented, a more thorough testing procedure will be implemented to ensure the music generator works as designed.

9 Project Progress

9.1 The Echo Nest Interface

The Echo Nest interface is completely finished. It is capabale of making a call to the Echo Nest's API using the user input and returning a list of song IDs, which the learning agent can utilize when it gathers user preferences in the music clip rating portion of the application.

9.2 Learning Agent

The learning agent is largely finished at this point in time. It is capable of reading in a sequence of notes, applying the Baum-Welch algorithm to the data, and adjusting the parameters of it's data model to better fit that sequence of notes.

The main issue that is preventing the learning agent from being totally complete is the difficulty of getting GHMM to accept a sequence of objects to learn from. This problem is currently being handled by passing GHMM a carefully constructed string that represents each note object from a sequence. This solution could potentially work be used for the rest of the project; it is just uglier and somewhat more difficult to read that if note objects were passed around, rather than their string representations.

9.3 Composition Agent

The composition agent has not been started, and will be the primary focus of the next phase in development.

9.4 User Interface

A basic, but functional user interface has been created that takes in genre, tempo, key signature, and time signature from the user and passes those parameters to the Echo Nest interface. The user is then redirected to a page where she will provide ratings of songs that the learning agent can utilize in its learning.

10 Discussion

As previously discussed, the project is on pace to be completed by the end of the semester. There have been various minor issues, but workarounds have been found for all of them, and as such, no major changes have been made to the scope of the project.

11 Appendices

11.1 Database Design

As there is not a large quantity of data to be saved for this project, there is no associated database.

11.2 User Manual

Given the incomplete nature of the project at this point, a user manual is not appropriate.

11.3 Programmer Manual

Given the incomplete nature of the project at this point, a programmer manual is not appropriate.

References

[1] General hidden markov model. http://ghmm.org/.

[2] Remix api. http://echonest.github.io/remix/apidocs/.

[3] B. Matityaho and M. Furst. Neural network based model for classification of music type. In *Electrical and Electronics Engineers in Israel, 1995., Eighteenth Convention of*, pages 4.3.4/1–4.3.4/5, March 1995.

[4] W. Schulze and B. van der Merwe. Music generation with markov models. *MultiMedia, IEEE*, 18(3):78–85, March 2011.

[5] P. Sheikholharam and M. Teshnehlab. Music composition using combination of genetic algorithms and recurrent neural networks. In *HIS '08. Eighth International Conference on Hybrid Intelligent Systems*, pages 350–355, Sept 2008.

[6] K. Youssef and Peng-Yung Woo. Music note recognition based on neural networks. In *ICNC '08. Fourth International Conference on Natural Computation*, volume 2, pages 474–478, Oct 2008.

Automatic Music Generator

Project Proposal

Sam Fleckenstein (sef44)
Ross Nanopoulos (rdn21)

February 18, 2014

Contents

1 Abstract

2 Background

3 Intended Audience

4 Architecture Design
- 4.1 The Echo Nest Interface
- 4.2 Learning Agent
- 4.3 Composition Agent

5 Project Management
- 5.1 Communication
- 5.2 Source Control
- 5.3 Work Division

6 Completed Work

7 Software Requirements
- 7.1 The Echo Nest Interface
  - 7.1.1 The Echo Nest API
  - 7.1.2 Interface Component
- 7.2 Learning Agent
  - 7.2.1 Internal Architecture
  - 7.2.2 Parser to Learning Agent Interface
  - 7.2.3 Learning Agent to Composition Agent Interface
- 7.3 Composition Agent

8 Software Specifications
- 8.1 UI
- 8.2 The Echo Nest Interface
- 8.3 Learning Agent
- 8.4 Composition Agent
- 8.5 User Feedback

9 Wishlist Features

10 Project Challenges
- 10.1 User Feedback
- 10.2 Extracting Patterns

11 Glossary

1 Abstract

The purpose of this project is 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 composition including hidden Markov models [7], genetic algorithms [5], and neural networks [10]. Additionally, a field known as "combination theory" has combined these methods to create more advanced learning and composition algorithms [9]. Hidden Markov Models utilize an element of probability and uncertainty that can lead to much more interesting compositions [7], which is the primary method of algorithmic composition that the system in this report will use.

The argument can be made that innate creativity plays a large role in being able to compose interesting music. However, a goal of artificial intelligence is to eventually develop systems that can think and have personalities of their own. Thus, this innate creativity when composing music will develop with more advanced artificially intelligent systems that can think for themselves and exhibit such behavior.

3 Intended Audience

Researchers who would like to incorporate different learning techniques for algorithmic composition will be able to easily integrate with this system. Additionally, it will be easy enough to use for someone who has no musical talent, but still wants to compose music.

4 Architecture Design

4.1 The Echo Nest Interface

The backbone of the Echo Nest interface 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 fingerprint, timbre, pitch, and loudness. Additionally, The Echo Nest provides 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 Echo Nest and discover the relevant patterns in the music. There are a number of different algorithms that could be used to achieve this goal, but this project 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 known. This is useful because music is very complex and it is very difficult to determine every variable that goes into determining

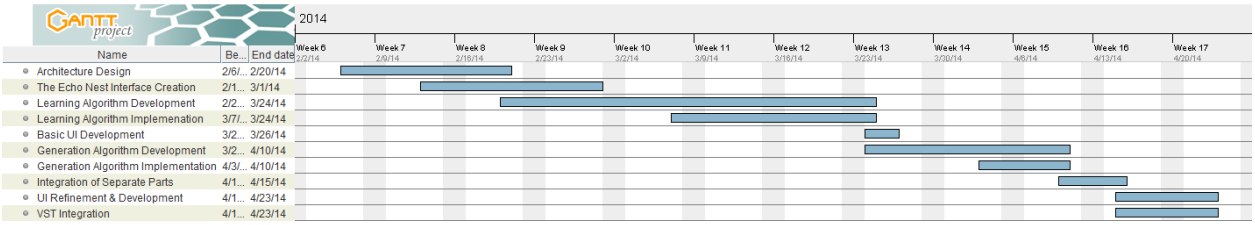
what should come next in a song. Another reason that hidden Markov models were chosen for this project is because they have been successfully applied the automatic generation of music [7]. The complexity of a hidden Markov model is also very easy to expand. This can be done by looking at data that is farther in the past from the current observation, or by adding in more variables to the states you are considering [3].

4.3 Composition Agent

The composition agent will take the information that the learning agent provides and decide which notes, patterns, rhythms, etc. to incorporate 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 and schedule:



These dates were decided by the best estimate of how long each component will take to implement. The development of the learning and composition algorithms will take the most time and will be in continuous development; therefore, they overlap with many other aspects of the project. The user interface will take less time to implement, and can be incorporated into other stages of development. Additional features such as VST support will be completed towards the end of the project as time permits.

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 interface to The Echo Nest's API, as well as the development and implementation of the composition algorithms. 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 [6, 8] . Second, because it appears to be complex enough to produce interesting music, but simple enough that it will be possible to complete the project in the given timeline.

7 Software Requirements

7.1 The Echo Nest Interface

7.1.1 The Echo Nest API

The Echo Nest API allows a user to search their catalog by a variety of methods that respond in JSON or XML, but only the JSON responses will be used. Many of the attributes in the JSON responses have confidence levels down to the beat, which is the Echo Nest's confidence of how likely the logical beat represents a real physical beat in the song.

7.1.2 Interface Component

This component will interact 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 with which to reason. The Echo Nest's API returns JSON objects that can be passed on by the song parser, in order to be used by the learning agent in its reasoning.

7.2 Learning Agent

7.2.1 Internal Architecture

The learning agent will use the GHMM library [2] to perform the required machine learning. This library was chosen because it offers all of the features needed to perform the required learning task. It is also free, another important feature.

7.2.2 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.

7.2.3 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.

7.3 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 be developed), which will decide which of the relevant patterns will be chosen, as well as a component to write a .wav file to disk. Additionally, there will be a component to compare the new song with the songs that were used to create it.

8 Software Specifications

8.1 UI

1. The UI will prompt the user for a musical genre
2. The UI will prompt the user for a song tempo
3. The UI will prompt the user for a time signature
4. The UI will prompt the user for a key signature
5. The UI will send these user choices to the Echo Nest Interface

8.2 The Echo Nest Interface

1. The Echo Nest Interface will take as input the user input from the UI
2. The Echo Nest Interface will make a call to The Echo Nest API using the user input
3. The Echo Nest will output JSON objects to the Learning Agent

8.3 Learning Agent

1. The learning agent will use the GHMM library
2. The learning agent will take as input JSON objects from the Echo Nest Interface
3. The learning agent will train a model [2] using the GHMM library and the input from the Echo Nest Interface
4. The learning agent will output a trained model to the composition agent

8.4 Composition Agent

1. The composition agent will take as input a trained model from the learning agent
2. The composition agent will choose an overarching chord progression
3. The composition agent will then fill in notes using the trained model
4. The composition agent will write the resulting composition to disk as a .wav file

8.5 User Feedback

1. The user will be prompted to rate a number of short song clips
2. The user ratings will be tabulated and used to update the learned model based on which musical patterns the user rated highest
3. The user will be presented with one longer song based on the updated musical model

9 Wishlist Features

Another feature that may be added is a different learning algorithm. If there is time, neural networks or genetic algorithms may be explored and 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.

10 Project Challenges

10.1 User Feedback

One of the main challenges with this project will be incorporating user feedback in some meaningful way. It would not be terribly difficult to write a program that produces some kind of music, but to write a program that successfully uses user feedback to improve the music created is much more challenging. It is also challenging to get useful data from a user who is asked to rate an entire song without having the user rate thousands of different songs. To address these challenges, the system will ask the user to rate short snippets of music that represent key patterns in the training data (i.e. the most common patterns from that music). These ratings will be used to determine which patterns the user likes the most, which will then be reflected in an update to the trained music model.

10.2 Extracting Patterns

The process of extracting the patterns that define a user specified set of music will be very challenging. To address this challenge, the task of learning has been simplified as much as possible. This has been done by choosing to use an already created hidden Markov model learning library that fits the needs of this project, rather than implementing an entirely new learning algorithm

11 Glossary

time signature: notational convention to specify how many beats are in each measure
tempo: the speed or pace of a given piece of music and derives directly from the average beat duration
segment: a set of sound entities (typically under a second), relatively uniform in timbre and harmony
tatums: subdivisions of beats
beats: subdivisions of bars
bars: a segment of time defined as a given number of beats; bar offsets also indicate downbeats, the first beat of a measure
sections: defined by large variations in rhythm or timbre (e.g. chorus, verse, bridge, guitar solo, etc.)
key: a track-level attribute ranging from 0 to 11, corresponding to one of the 12 keys: C, C, D, etc. up to B
mode: equal to 0 or 1 for minor or major, respectivey, and may be -1 in case of no result
loudness: given by three data points: dB value at onset (beginning of a piece), dB value at peak (loudest point in piece), and a segment-relative offset for peak loudness (where the loudest peak occurs)
pitch: given by normalized vector corresponding to the 12 pitch classes with values from 0 to 1. Noisy sounds are represented by values close to 1; pure tones are described by one value at 1 (the pitch) and others near 0
timbre: the quality of the musical note or sound that distinguishes different types of musical instruments or voices
audio fingerprint: a condensed digital summary, deterministically generated from an audio signal that can be used to identify an audio sample or quickly locate similar items in an audio database

References

[1] The echo nest api analyze. http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf.

[2] General hidden markov model. http://ghmm.org/.

[3] Hidden markov model. http://en.wikipedia.org/wiki/Hidden_Markov_model.

[4] Michael Edwards. Algorithmic composition: Computational thinking in music. *Commun. ACM*, 54(7):58–67, July 2011.

[5] B. Matityaho and M. Furst. Neural network based model for classification of music type. In *Eighteenth Convention of Electrical and Electronics Engineers in Israel, 1995.*, pages 4.3.4/1–4.3.4/5, March 1995.

[6] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(4):360–370, Apr 1999.

[7] W. Schulze and B. van der Merwe. Music generation with markov models. *MultiMedia, IEEE*, 18(3):78–85, March 2011.

[8] Xi Shao, Changsheng Xu, and M.S. Kankanhalli. Unsupervised classification of music genre using hidden markov model. In *ICME '04. 2004 IEEE International Conference on Multimedia*, volume 3, pages 2023–2026 Vol.3, June 2004.

[9] P. Sheikholharam and M. Teshnehlab. Music composition using combination of genetic algorithms and recurrent neural networks. In *HIS '08. Eighth International Conference on Hybrid Intelligent Systems*, pages 350–355, Sept 2008.

[10] K. Youssef and Peng-Yung Woo. Music note recognition based on neural networks. In *ICNC '08. Fourth International Conference on Natural Computation*, volume 2, pages 474–478, Oct 2008.

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.