

Machine Learning and Music Composition

Daniel Woeste
Division of Math and Science
University of Minnesota, Morris
Morris, Minnesota, USA 56267
woest015@morris.umn.edu

ABSTRACT

1. INTRODUCTION

2. BACKGROUND

Things to explain
-Machine Learning
-Decision Trees
-over fitting
-Finite Automata

2.1 General Framework

All of the methods described in this paper share a similar framework that would be useful to understand. The commonality between all of the methods is that every note is generated in relation to a cache of previous notes. Meaning that every note is dependant upon a set of notes before it, whether it is only the previous note or a larger set of notes. This allows the program to determine a melodic progression for the music.

2.2 Methods

In this section, we will be describing three separate subcategories of machine learning. The three subcategories of machine learning that we will be describing are *random forests*, *markov chains*, and *neural networks*.

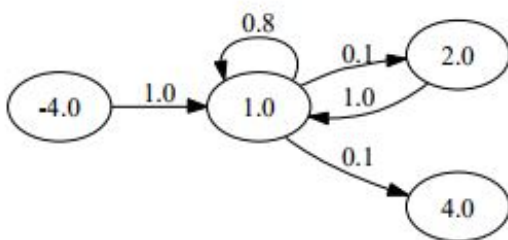


Figure 1: Finite automata detailing the Markov Generated in Automatic Composition of Music with Methods of Computational Intelligence

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>.
UMM CSci Senior Seminar Conference, December 2017 Morris, MN.

First, we discuss the use of Random Forests. Random forests operate by generating and altering a large collection of decision trees. The generated output of the random forest is the result of the average answer of all of the decision trees in the forest. As with all machine learning a set of amount of time is required to teach the algorithm to properly generate answers. For the case of Random Forest, each decision tree is made on only a small part of the set of data, with overlap among all of the trees. This overlap in the data that the trees are built on helps to reduce over fitting of the trees to the data. When it comes to music we discuss a program known as ALYSIA, that uses two random forests to generate music from a given set of lyrics.

Next, markov chains function fundamentally differently than random forests. Markov chains use a state machine and a statistical models to predict the next value. Figure 1 shows a simple finite automata for a markov chain, containing only the possibility of having quarter, half, or whole notes, that was generated for the paper Automatic Composition of Music with Methods of Computational Intelligence. The automata visualizes the way the markov chain uses probabilistic determination to predict what the next note or value will be. In the terms of music this means that for each state in the automata it has a percentage chance of moving to each of the other nodes in the automata. In this case, the first note will be a quarter note with an 80% chance of having a repeated quarter note, and only a 10% chance of having a half note following.

Neural Networks offer yet another approach to using computers to generate music, this time being a deterministic style of algorithm.

Need to add much more to the section on neural networks.

3. METHODS AND MUSIC

In this section, we will discuss how the previously stated methods can be applied to the composition of music with specific examples.

3.1 ALYSIA

ALYSIA, Automated LYrical Song-writing Application, is a program that uses random forests to generate songs and melodies for a given set of pop lyrics. ALYSIA contains two different random forests that work in tandem: one to produce rhythm, the other to produce pitch values [ALYSIA]. Random forests can provide some inherent benefits over Markov chains. Even though both models work from a cache of previous notes and both predict what the next note should be.

Markov chains offer only a statistical probability of what the next note should be, while random forests include a reasoning to what the next know should be.

3.2 Citations

3.3

Acknowledgments

Fix the section heards because currently they are not organized in a way that makes sense.