Dissertation Type: research

University of BRISTOL

DEPARTMENT OF COMPUTER SCIENCE

Performing Algorithmic Co-composition Using Machine Learning

Stephen Livermore-Tozer

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Wednesday 20th April, 2016

Declaration

Stephen Livermore-Tozer, Wednesday 20th April, 2016

Contents

List of Listings

Executive Summary

For my research, I have investigated the hypothesis that existing methods of performing melodic composition using machine learning can be adapted to successfully perform co-composition with a human composer. In this context, success may be measured by subjective impression received from experts in composition. The exact set of tasks involved in the co-composition process is not fixed; part of this research involved finding the tasks that are both a) useful to the composer, and b) within the capabilities of current AI.

In the pursuit of this objective, I have completed the following tasks:

- I surveyed a number of expert composers to identify the tasks within the domain of composition whose partial automation would provide the greatest benefit to the composer.
- I performed research over existing machine learning methods for algorithmic composition, determining their applicability and estimated efficacy within the tasks obtained above.
- I implemented an application that may be trained on a set of music data (stored in the Midi format) and will provide an attempted continuation of a given musical input, according to the tasks obtained above.
- I designed and performed a series of objective and subjective tests to assess the performance of the application and the algorithms that comprise it.

Supporting Technologies

I used the Anaconda implementation of Python to create my application. Furthermore, I used a set of libraries implementing machine learning techniques to create the application. These libraries are:

- NumPy for efficient mathematical operations
- SciPy for clustering and regression
- PyBrain for neural networks
- HmmLearn for hidden Markov models

Acknowledgements

I would like to thank my supervisor Peter Flach for his guidance over the course of this project in both its technical aspects and in its effective management.

I would also like to thank my good friend Thomas Norton, for his expert musical advice and support throughout my work.

Chapter 1

Contextual Background

1.1 Algorithmic Composition

1.1.1 Overview

Algorithmic composition is defined as the use of algorithms to compose, in whole or in part, a musical score. This is generally accomplished using a computer, although many earlier systems could be traced out by hand. There exist a large and diverse number of algorithms designed for this task, accompanied by many different representations for musical data. Among these algorithms are a large number of methods for learning musical creativity, including supervised learning, unsupervised learning, rules and constraints, and stochastic modelling, among others.

Although many methods for algorithmic composition have been defined over the past several decades, few have achieved notably useful or successful results. Even relatively sophisticated and advanced methods tend to produce music with notably inhuman idiosyncrasies or that is subjectively described as “bland.” There are several key contributing factors to this problem, two of which are the lack of both computational power the unavailability of sufficient quantities of useful data. One way of describing the core issue however is that there does not currently exist a good way of measuring or analysing human subjective impression to a broad or deep enough scale. Because of this nearly all current methods either attempt to codify compositional principles as used by humans, extract features from music known to be “good,” or perform some combination of the two. This is not necessarily an incorrect approach, but knowledge directly inscribed by humans is universally insufficient to capture all the necessary complexity of music created by humans, and it is currently prohibitively difficult for machine learning to extract meaningful features from existing music.

1.1.2 Previous Work

Using algorithms or similar formal methods in the composition of music is not a purely digital phenomenon - certain forms in classical music, such as *canon*, involved using a set of instructions to develop a simple melody into a whole segment of a composition.

The introduction of computers to the process of algorithmic composition occurred in the mid 1950s. At this time a number of experiments were proposed, and the first actual production of a computer-composed piece took place in 1957.

1.1.3 Rule-based systems

1.2 Other stuff

Artificial Intelligence in its current state is a major component of the current technological landscape. As a broad field, AI plays a significant role in an enormous number of areas with a large technological component, including large industries such as finance, medicine, marketing, and internet services. In particular one of the most frequent and powerful uses is to learn from large quantities of data and perform analytical tasks, such as classifying samples in some way or identifying complex trends.

There exist several reasons for the interest in the creation of AI that can compose music. The most directly practical of these is that music is a multi-billion pound industry, and music production has been revolutionised by technology on many occasions. The power of introducing a significant degree of automation to the process may be immense on both a financial and cultural scale.

Additionally, the application of AI to the arts is a heavily romanticized achievement; “creativity” is quite universally considered to be a shortcoming of AI, and the creation of art is generally considered to be one of the most significant expressions of human creativity. Because of this, AI that is capable of creating art comparable to that created by humans represents a technological and cultural milestone.

Within the sphere of the arts, music is a key target due to its heavily pattern-based, mathematical nature. Even with these advantages however, AI performance in composition is generally considered to be distinctly poor. According to the majority of subjective measures, AI is identifiably so due to being lacking in creating both interesting short-term melodies and coherent long-term structure.

Chapter 2

Technical Background

2.1 Overview

There are many different methods for performing algorithmic composition, with a broad range of algorithms and data structures used. Most major AI techniques have many corresponding applications to algorithmic composition - it is also quite common for these techniques to be combined, such as combining neural networks and evolutionary algorithms. Additionally, many of these methods are focused on a specific subtask of composition; popular targets are chord identification, continuation of single line melody, and counterpoint, while many other algorithms focus on more specific tasks (typically relating to a particular genre, such as jazz or baroque).

2.2 Knowledge-Based Techniques

Knowledge-based systems are those that use a symbolic representation of knowledge to complete a task, and were one of the first forms of AI created. This generally means storing a database of facts and then inferring with this data to resolve statements. In the case of algorithmic composition there are several common abstract approaches that use and represent knowledge in this way.

2.2.1 Rule-Based Systems

The first attempts at digital algorithmic composition were knowledge-based, and typically involved encoding explicit, hand-written compositional rules into code and using those rules to create or modify music. These systems notably incorporated no learning of any kind - the behaviour of the program was entirely determined by the initial set of rules and the input (typically a randomly generated melody). This had the effect of making the systems very "brittle," meaning that performance typically dropped dramatically whenever the system encountered an unideal input - a very common occurrence due to the limited scope covered by the initial rules. This can be attributed to the fact that while the compositional principles used by humans are useful and powerful in human hands, they are heavily dependent on the composer providing their own creative input throughout the process.

2.2.2 Musical Grammars

Other more computationally focused attempts at musical knowledge-based systems have made use of formal grammars. A grammar consists of a set of nonterminal symbols N, a set of terminal symbols Σ (s.t. $N \cap \sigma = \emptyset$), a start symbol $S \in N$, and a set of production rules that transform strings containing at least 1 nonterminal symbol. A grammar can be used to produce strings by taking a string consisting of the start symbol and repeatedly applying valid production rules to the string until it contains only terminal symbols. Grammars are thus well suited to parsing and generating hierarchical structures recursively, which makes them a good fit for music.

Formal grammars for composition first began to gain popularity in the mid 1970s, with hand-crafted grammars used to represent the structure of music [2][3].

2.3 Supervised Learning Techniques

Another major branch of work within the field is the use of supervised learning to train AIs through a set of examples, instead of encoding knowledge by hand. This does away with many of the disadvantages of knowledge-based systems, particularly their brittle nature - by learning from examples, the AI can feasibly adapt to work with any particular style of music (with varying difficulty). Currently supervised learning is a very popular method for algorithmic composition, with a majority of papers from the past decade relying on it in some fashion [1]. This is linked to a combination of the newly widespread availability of musical data to learn from, the increase in computational power (and hence potential complexity of learning models compared to static models), and the high popularity of supervised learning in general.

Generally however these methods still have some significant drawbacks. One of the most fundamental of these flaws is the fact that due to their knowledge being entirely derived from a training set they are limited in their ability to develop novel material, instead simply imitating the styles observed during training. There have been attempts to overcome this issue, with many of the most successful attempts using a combination of different techniques (see **??**).

2.3.1 Artificial Neural Networks

The most popular class of supervised learning techniques in the context of algorithmic composition are artificial neural networks (ANNs). This can be related to their ability to learn a mapping from one set of data to another, allowing them to be adapted to a complex problem domain such as music. The earliest use of neural networks for algorithmic composition was during the late 80s, where a simple 3-layer recurrent neural network (RNN) was used to compose short monophonic melodies [5]. Many subsequent ANNs have used a similar design, with adjustments made to the topology and data representation.

Chapter 3

Project Execution

3.1 Design

3.1.1 Composition Algorithm

One of the key details that sets the intended method apart from most other algorithmic composition methods is its purely cooperative nature. This allows for a major performance benefit by offloading a certain portion the work to the human operator. There are three major places in which this offloading can take place: initializing state before generation, guiding the generation process, and adjusting the final generated output. The first is achieved by using the score written by the user as input to create new material from. The third is an intrinsic part of the process, as the final output of the algorithm is provided to the user to be used as they please. The second may be exploited to improve the performance of the algorithm in generating output that is useful or pleasing to the user.

There are a number of ways that this effect could be achieved. One would be the use of evolutionary algorithms. Due to their use of a fitness function to guide their search through the solution space, it is simple and straightforward to integrate user feedback into the EA by using user rating - either as the fitness function, or as some kind of augmentation to the fitness assignment process. A number of EAs have been designed with this behaviour in mind in order to overcome the challenge of defining an objective fitness function. This does however suffer a major drawback in that it is not feasible for an individual user to guide the search process single-handedly; due to the large number of samples that must be judged, user fatigue rapidly becomes a critical constraint. There have been attempts to mitigate this effect through various means, such as reducing the population, number of generations, or novel methods such as using clustering to avoid presenting the user with similar outputs. These methods each present their own limitations however, to an extent that was prohibitive in the case of this research project.

3.1.2 Training

As discussed in sections **??** and **??**, the AI contains several components using supervised learning. Therefore the results produced by these components, and by extension the AI, will be heavily dependent on the training data provided.

Providing training data for the AI to learn from is potentially a non-trivial task. The reason for this is that the data must fulfil a certain set of constraints, namely:

- The data must be stored in a machine-readable format
- The input must not contain any features that the AI is not capable of parsing
- There must be a strong degree of consistency within the data so that the learned patterns are coherent

The format used to store music throughout this project is the Midi file format [4], which stores music in the form of notes and events instead of sound. This is ideal for this application, as it allows for direct recovery of the exact musical elements contained within each piece of music, a significantly more challenging task when dealing with raw audio. It is also convenient for data acquisition as there exist various large collections of royalty-free midis available to the public. Unfortunately these midis are very often classical music of some variety which often fails to satisfy the second requirement.

The second requirement is one of the most constricting, as for various reasons the AI has many restrictions on the type of musical data it can work with. The most harmful of these is the inability to parse or generate polyphonic melodies. This causes some major issues when working with orchestral or piano music due to their frequent use of many melodic lines and chords respectively, which excludes most available classical music. Although it is in some cases possible to isolate a single melodic line, there are quite often no such lines that are both long and substantial enough to be useful for training.

To best overcome these restrictions, the data used was a selection of nursery rhymes transcribed for this research. These have the advantage of being relatively simple and sharing many commonalities such as key and meter, making it much easier for the AI to identify and recreate patterns. They are also universally monophonic due to their vocal nature.

3.2 Implementation

Bibliography

[1] Jose D Fernández and Francisco Vico. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, pages 513–582, 2013.

[2] David Lidov and Jim Gabura. A melody writing algorithm using a formal language model. *Computers in the Humanities*, 3:138–48, 1973.

[3] Gary M Rader. A method for composing simple traditional music by computer. *Communications of the ACM*, 17(11):631–638, 1974.

[4] Andrew Swift. A brief introduction to midi. *URL http://www. doc. ic. ac. uk/˜ nd/surprise_97/journal/vol1/aps2*, 6, 1997.

[5] Peter M Todd. A connectionist approach to algorithmic composition. *Computer Music Journal*, 13(4):27–43, 1989.

Appendix A

An Example Appendix

Content which is not central to, but may enhance the dissertation can be included in one or more appendices; examples include, but are not limited to

- lengthy mathematical proofs, numerical or graphical results which are summarised in the main body,
- sample or example calculations, and
- results of user studies or questionnaires.

Note that in line with most research conferences, the marking panel is not obliged to read such appendices.