A Study on Machine Learning for Music Composition

Ren He

ABSTRACT

To train computers to perform humane tasks is a trend in the development of artificial intelligence. Among all, while music composing is one of the simplest area to approach, it still require the machine to mimic human thinking process to an extend, I therefore chose to research on it before I go into deeper topics. In this report, I will discuss an immature attempt I practiced, and possible future attempts after I learn and deliberate over existing methods including genetic algorithms and deep learning network.

Introduction

Attempts to program machines to simulate human's way of thinking is a topic of interest in modern computer science. Currently, existing AIs can perform a variety of tasks ranging from basic dialogue translation to chess gaming. Consider when a human plays chess, his decision on the next move is based on factors including current situation on the board, expected outcomes of the move and prior experience of similar situations. A chess AI works in the same way. However, tasks like chessing are just introductory application of AI. The area has vast possibilities for further exploration.

Before approaching researches on more advanced AIs, I considered a try teaching the machine to compose musics as a first step attempt. Music, according to historians, originated before the existence of language. It emerged as a form of expression for emotion and thoughts. Some discovered prehistorical murals depict people making sounds by blowing horns, hitting rocks to celebrate harvesting or winning battles. Throughout history, music evolved in complexity and branch into various genres, but most fundamental elements of a piece always include melody, tempo and timbre. Those properties can be easily presented in computer languages. However, balancing and combining of the elements to produce an acceptable song requires knowledge of the flow of mood. For example, connecting a falling paragraph and a rising paragraph without reasonable transition usually makes a song sound fragmented. To counter the problem, algorithms of machine learning are worth considering.

In this report, I am going to compare a classical decision tree model with two existing machine learning algorithms, the genetic algorithm (GA) and deep neural network (DNN), to show how the latters are more promising and efficient in the work of music composition. I will also discuss possible improvements for the algorithms and potential future steps in the area.

Results

Most fundamental of all, in each algorithm, we have output of the same format, and some algorithms share similar logic and theory.

First, we use the piano scale as the basic units of nodes. Piano is the most prevalent and standard instrument in the world. It has 88 nodes ranging from a 27.5 Hz A (la) to a 4186 Hz C (do), wide enough for most existing songs. Unlike that of string instruments, all nodes on a piano are discrete. This characteristic greatly simplify the computation models.

Second, the algorithms shall export the music in MIDI files. The MIDI format has structure like a matrix, each block of it contains information of a node and whether it lasts till the next block, as well as the instrument it uses. The format can well represent melody, tempo and timbre of a song, and those are enough information for purpose of my project.

Third, we consider all melodies in a C Major or minor. On one hand, it can simplify the model. It is the interval between nodes that makes a melody. On the other hand, it can avoid confusion of the program. Specifically in the case of the deep learning algorithm, because songs in the data base can be in a variety of scale, normalizing any scale to a C Major or minor can help the program to focus more on intervals instead of exact location on the piano. For better sound of the completed piece to avoid having notes with pitches too high or too low, the scale can shift accordingly.

Fourth, for the decision tree and GA methods, we consider the pop music genre. Songs of the genre usually does not include any nodes other than the seven nodes in a Major or minor scale. It can simplify the model, such that we can avoid considering the many combinations of half steps.

For notations of melodies, I note the value of central C as 1. For each node in a Major or minor scale, its value is 1 higher than its left neighbor and 1 lower than its right neighbor. That is, for example in a Major scale, D next to the central C has value of 2, and A closest to the central C is noted as -1. For notation of tempo, we use an $1/32$ node as the smallest time unit. So time value of a $1/32$ node is 1, and that of a whole node is 32. For instruments, we consider only piano in the decision tree and the GA method. We potentially can consider adding more instruments when using the DNN method.

With the rules and conditions, I considered the following three approaches:

A Decision Tree Model

A decision tree is among the most controlled structure for machine learning. Basic idea of the algorithm is to first instruct the computer know as much rules as the programmer does, while giving it freedom to innovate on combinations of melodies and tempos reinforced by responses from a user. As illustrated in Figure 1, with personal experience of composition, I built up a tree from draft with the following rules and constraints:

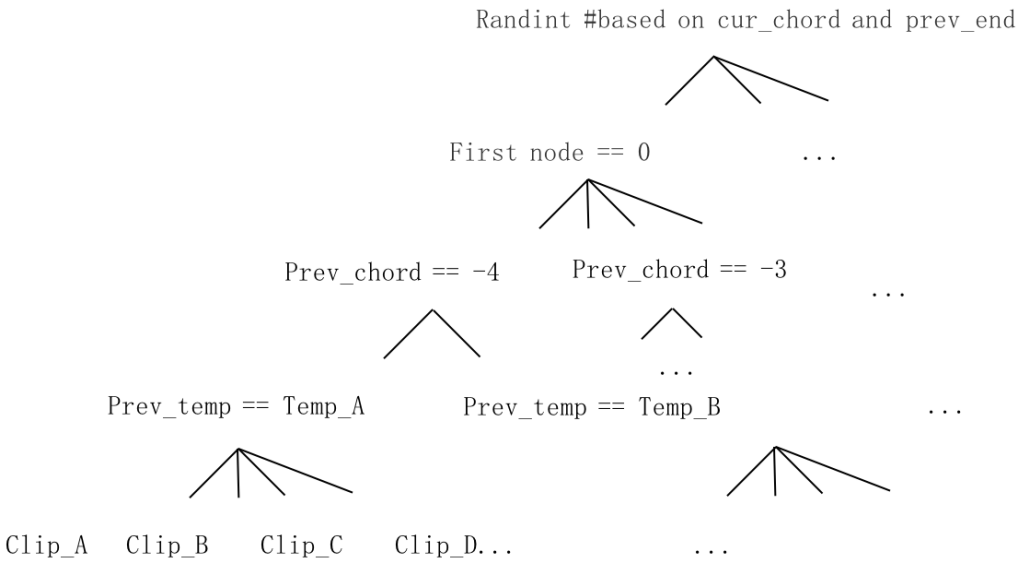

Figure 1. A conceptual composition decision tree.

1. I used the chords to lead melodies. While the contrary may seem more intuitive, in fact, when a human composer write a melody, he implicitly has considered certain chord patterns to match or even lead the melody. The initial choices of chord includes $\{C,G,A,F\}$, $\{A,F,C,G\}$ and $\{F,G,F,G\}$ etc. for C major, and $\{c,\flat b,\flat a,g\}$, $\{c,\flat b,\flat a,\flat b\}$ and $\{c,\flat b,\flat a,\flat a\}$etc. for c minor. The algorithm may also generate variations, and the new patterns can be reinforced based on feedback from the user.

2. Starting from nodes B,C,E,F may affect rules of a lot of the upper branches in the tree. Because the interval between B,C and E,F is $\frac{1}{2}$, while distance between any other neighboring nodes are 1.

3. Interval between nodes in melodies. Intervals shall not be too large, as it not only hinders performance of a song, but also usually makes a melody sound bad. Intervals shall not be too small, or it will be the same node repeating through out time. I take into MIDI files of multiple pop songs and calculated probability of different intervals, and used it in the model.

4. In a same chord, the starting node can differ. For example, in chord C stanza, nodes A,C,E,F,G are more often used as the starting node than the two remaining nodes. If the stanza is the first stanza in a song, a starting node is randomly picked. Otherwise, possibility of each node calculated from the last node in the last stanza using algorithms similar to rule 3.

5. Flow of tempo. Tempo cannot be totally random. Consider a $4/4$ time signature. By beginning of a paragraph, stanzas would have have comparatively more lively tempos, for example, $\{4,4,2,2,2,2,4,4,4,4\}$ and $\{6,2,6,2,4,4,4,4\}$, which most nodes usually have similar lengths, while by end of a paragraph, stanzas may be slower, and contain longer ending nodes, for example, $\{8,4,4,16\}$ and $\{2,2,2,2,4,4,16\}$. The algorithm contains a variety of different tempos analyzed from existing songs, and innovative combinations of tempos can be reinforced by evaluation from users.

6. With every thing organized in a micro perspective, a song needs an overall structure. A pop song usually goes as $\{$intro, verse I, chorus, verse II, (expansion), chorus, outro$\}$. Paragraphs can be separated into the form, then melodies and tempo divisions follow accordingly.

On one hand, the advantage of using a decision tree is that it is highly controllable and easy to comprehend. As a programmer, when the algorithm goes wrong, I can quickly locate the problem and modify the constraints.

On the other hand, the algorithm has many disadvantages. First of all, the feedback system is questionable. There are too much elements for the user to evaluate. As users of the algorithm are human beings, their feedback is subjective, and may be inconsistent as their fatigue grows. Secondly, its immense complexity is problematic. To implement all the rules and constraints into the algorithm grows the tree to an inefficiently large size. Last but not least, its creativity is very limited. As the programmer had put into a large amount of constraints, the algorithm's space for further creation is restricted, and its output would not differ too much from a song directly written by the programmer.

Overall, the algorithm is inefficient and problematic. It may not be able to compose creative songs, but only be able to assist the programmer to gain ideas of composition.

The Genetic Algorithms

Genetic algorithms (GA) is a structure inspired by evolution of biological genes. Its basic functions are to randomly generate variations based on a existing clip, and promote or demote the variation based on user feedback. The method emerged by end of the last century in assistance of music composition.[1]

The algorithm is similar to the decision tree one. It contains methods such as "no operation", "delete", "mutate", "rotate" that works on an existing clip. The clip can be either randomly generated, or input by the programmer or user. The number of methods called on a clip can be randomly generated, or too input by the programmer or user. For example, if the user input $\{C,D,E,F\}$, the algorithm can first call a "delete" method, then a "mutate", and the clip may first be modified into $\{C,D,E\}$, then $\{G,D,E\}$, which becomes the final output. In this algorithm, the user also need to evaluate the outputs to help the algorithm reinforce certain clips, or combinations of methods.[2]

On one hand, firstly, the genetic algorithm has more freedom for the algorithm to innovate compared to the decision tree model. Secondly, reinforcing on method combinations is an aspect not considered in the decision tree model. It can potentially discover extensible ways of varying a clip.

On the other hand, it has disadvantages, too. First, due to a lack of restrictions, outputs at early stage of the algorithm can be be too random, and it would take a long time of training before the algorithm can improve. Second, the user fatigue problem also exists, and is enhanced by the randomness problem, so the situation might be even worse than that we encounter in the decision tree algorithm.

Overall, the algorithm might be able to assist the programmer to gain ideas of composition, but it still rely an abundant on user interaction and it is not fundamentally different from the decision tree algorithm.

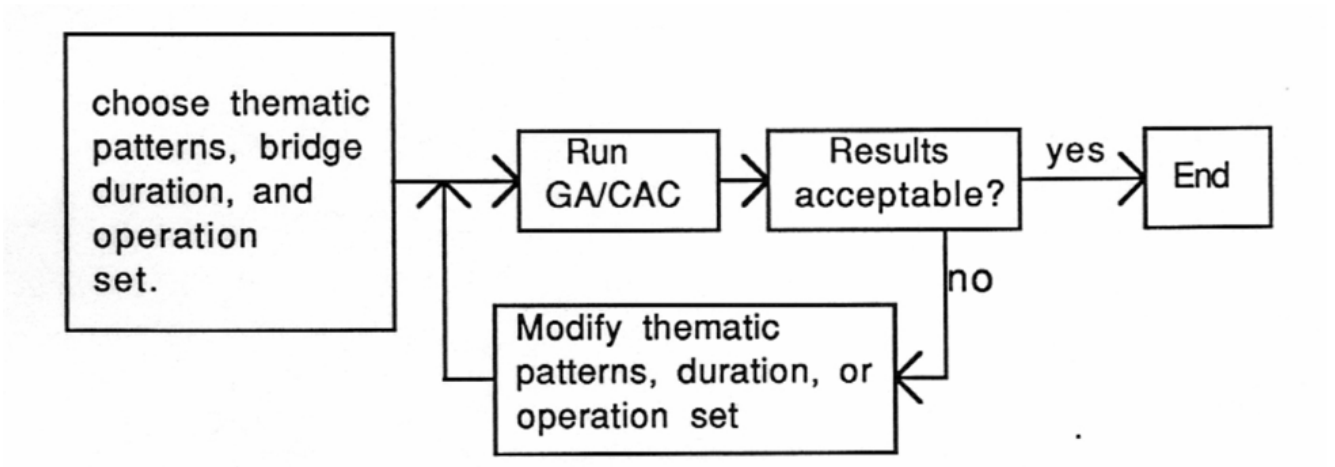

Figure 2. The overall structure of a genetic algorithm for composition.

The Deep Neural Network

Deep neural network (DNN) is a neural network structure with multiple hidden layers. The structure is inspired by the biological neural system. Each neuron can be triggered multiple times in a run, unlike that in a decision tree, each branch node can only be triggered once. By minimizing a cost function that quantifies deviation of the result from the desired output, the algorithm seek for a feasible solution based on optimization theory and statistical estimations. General neural networks have been used in

various scenarios to look for algorithms to convert certain inputs to certain outputs. Compared to other methods, it is more accurate in recognizing and finding nonlinear relationships.

Currently in the world, there exists a developed music composing AI, AIVA (Artificial Intelligence Virtual Artist) developed by the Aiva Technologies founded by Luxembourg and London by Pierre Barreau, Denis Shtefan, Arnaud Decker, and Vincent Barreau in 2016[3]. The AI applied deep learning algorithms. After learning from various songs written by classical musicians such as Mozart, Bach and Beethoven, it can now compose pieces that are recognized by the SACEM (Société des auteurs, compositeurs et éditeurs de musique), an official music society that distribute rights to original composers. The company did not release their algorithms, but if I were programming a composing AI using DNN, I would consider:

1. Take into MIDI files of a desired genre and use them as the database of outputs. Depending on requirements, we can start out by only using piano, and, if the database allows, we can add more instruments in the composition.

2. Qualitative input parameters. We need to consider qualitative aspects of a piece. We can include parameters such as "liveliness", "gothicness" and "fluency" etc. to describe a song. We can build a continuous scale for each of the parameters ranging from 0 to 10. In training of the algorithm, the programmer need to rate all the parameters and pair it with selected songs in the database.

3. We also need a series of structural parameters to reduce machine confusion, such as key signature, length of the piece and other structural qualities. It is possible that two songs of different lengths have all identical qualitative attributes. If we ignore the structural parameters, too much weight will be put onto the qualitative parameters.

4. Determine a reasonable cost function. This part requires music theory. We need to know whether deviation of tempo or melody is more severe a problem, or it would depend on conditions of certain parameters. We shall consult professional music scholars for more information to decide on a model.

On one hand, the algorithm has disadvantages. First, it has similar user evaluation problem as the previous two algorithms. The algorithm needs a large amount of training, and evaluating the songs in the database to rate the attributes is tedious work. Second, a cost function, as mentioned, is hard to find. Third, when crashes happen, origin of the problem is hard to track, as most process of the algorithm is hidden, and the user can only look through the database to seek the problems.

On the other hand, the algorithm has outstanding advantages. First, the algorithm can seek for nonlinear relationships between the input and output, and the programmer does not need to manually put into constraints. More importantly, once after the training succeed, the algorithm can yield more promising pieces than the other two can do.

Overall, the algorithm would be hard to train, but with appropriate amount of training, it has the potential to be the most flexible and creative algorithm of the three.

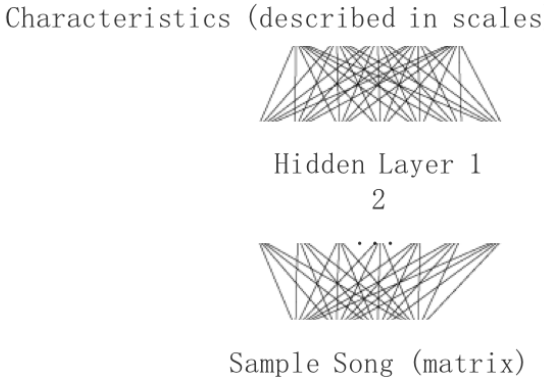

Figure 3. A conceptual DNN in composition.

Future Possibilities

To continue beyond current researches, we can consider to further work with the DNN algorithms, taking more creative inputs and outputs, and with implementation of other technologies.

We may consider taking more conceptual input parameters. For example, certain type of song can "smell good", while other types can "taste like a watermelon". Such method can help introducing qualities that are potentially ignored in the conventional parameters people used.

In the case of pop songs, adding vocals to the output can also be a choice. We can implement existing lyric writing algorithms to our algorithms. We can either add lyrics as a part of output, or built up another model to generate lyrics given our output melody. If the technology of artificial voices are more fluent in the future, we can even mix the vocals into our songs.

Combining our algorithms with picture analyzing and logic algorithms can yield an algorithm for the development of moving picture soundtracks. We can use existing muted pieces of movie clips as inputs, and the original background music as outputs. After training the algorithm with a sufficient amount of database, it can learn to compose soundtracks by taking a clip as input.

Even beyond that, when machine learning technologies are more matured, we can try simulating the action of composition from its very origin. That is, we can consider building up a machine that has a more humane thought process (made possible by combination of multiple DDNs), and has the ability to make sound. The sounds it make originate from the thought process, whether to be reinforced or not by the environment is decided by the thought process itself. In theory, the sounds very likely has the potential to be musics, and it can have more and more variety as the thought process increase in complexity, approaching or even transcending the human level.

Once we can achieve that, the technology can do much more than music already. Not only other kinds of arts, similar methods can also be used in other occupations which now requires human workers, for example, financial analyzing and model building. One day, probably when people master quantum computing technologies, even more algorithms can merge together into more powerful ones that can perform a variety of tasks. By then, the world can be a completely different place.

References

1. Tokui, N. & Iba, H. Music composition with interactive evolutionary computation (2000).

2. Horner, A. & Goldberg, E. D. Genetic algorithms and computer-assisted music composition (1991).

3. Machuron, C.-L. Aiva: The artificial intelligence composing classical music @ONLINE (2016).