Computational Models for Musical Composition: A Review

Junyoung Sim (Cornell NetID: js2992)

Abstract

Various computational models for musical composition are discussed and compared in this review. Specifically, an algebra-based model, a Markov-based model, and a deep learning-based model for musical composition are discussed and compared to each other by their methods, the quality of generated music, practicality, and strengths and weaknesses. Compared to an algebraic system having explicitly programmed musical components and rules, a Markov-based model and a deep learning-based model is suggested to be more effective and practical in generating plausible music by exploiting the statistical patterns in existing music. In terms of generating appealing music with high statistical likelihood, a deep learning-based model is suggested to be the most successful. The originality of music generated by the three types of computational models discussed in this review still remain as a continuing challenge in which further work may pursue.

1 Introduction

Musical composition is the exercise of creating music that has been continued throughout history and across cultures. Particularly in the Western hemisphere, the legacy of composers such as Bach, Mozart, Beethoven, and Chopin established the belief that the exercise of creating music is an ability exclusive to certain talented human beings. However, this belief may be questioned by the rise of computational methods that are able to create music by modeling the rules and patterns in existing music. For instance, an algorithmic process would be able to generate a piece of music by having a set of algebraic operations that manipulate the values of various musical components and rules, such as the pitch of notes, harmonization, rhythmic pattern, and more. This particular method can be considered the most intuitive approach since it explicitly defines and programs the various components and rules in musical composition. Besides explicitly-programmed algebraic systems, data-dependent statistical methods can also be used to generate a piece of music by extracting and applying patterns in existing music. For instance, Markov processes which assume that each state is dependent on its previous state and denoting such "transition probability" of states as $P(s_t \mid s_{t-1})$ can be used to generate a sequence of notes with high statistical likelihood. A more complex model known as neural networks in deep learning can also be used to generate music

with high statistical likelihood and a particular style by stochastically optimizing the parameters involved in extracting patterns from a large dataset of music and predicting a set of musical elements that are likely to follow a given tune.

The three aforementioned computational models generating music (algebra-based, Markov-based, and deep learning-based) are studied in the remainder of this review. For each model, its method will be explained and followed by a discussion of its results, the quality of generated music, practicality, and comparisons to other computational models.

2 Literature Review

2.1 Algebraic Musical Composition (*TransProse*)

An algebraic system having explicitly programmed musical components and rules is one of the various computational models that can be used for generating music. Davis and Mohammad proposed such an algebraic system called *TransProse* to generate music conveying emotions derived from textual sources [6]. The method of *TransProse* is explained as follows. An "emotion profile" describing the emotions of a given textual source (anticipation, anger, joy, fear, disgust, sadness, surprise, trust, positivity, and negativity) is extracted by using a framework called NRC Emotion Lexicon [10]. Subsequently, the emotion profile of a given textual source is used to generate values for tempo, scale, octave, and notes and their sequence. The values for such musical components and rules are computed by explicitly programmed algebraic equations. For instance, *TransProse* uses the equation below to compute the octave of a melody based on the joy-sadness (JS) density range of a given textual source.

$$Oct(M_0) = \frac{(JS - JS_{min}) * (6 - 4)}{JS_{max} - JS_{min}} \text{ [6]}$$

Besides determining the octave of a melody, *TransProse* uses relative emotion densities to select the duration of notes; sections of a textual source with high emotion densities is programmed to be composed of sixteenth-notes, whereas sections of a textual source with low emotion densities is programmed to be composed of whole-notes. For selecting notes of a melody, a linear mapping of the emotion densities and the pitches of a given scale is used; low emotion densities are mapped to more consonant pitches, whereas high emotion densities are mapped to more dissonant pitches.

Thus, a textual source with emotions such as anger and confusion are programmed to be represented by sixteenth-notes at dissonant pitches. Having such specific algebraic operations for various musical components and rules allowed *TransProse* to generate the melodies that are made public in http://transprose.weebly.com/clips.html.

As intended, *TransProse* is indeed able to generate melodies conveying the emotion of multiple textual sources, such as *Peter Pan*, *Heart of Darkness*, *Lord of the Flies*, and etc. Thus, the music generated by *TransProse* suggests that conveying a sense of emotion through music is not necessarily an ability exclusive to human beings because it is an aspect that can be modeled and performed algebraically. However, besides conveying a particular sense of emotion, the general quality of the music generated by *TransProse* may not be considered appealing; notes are constrained within certain pitches, the sequence of notes tends to be repetitive, and musical development cannot be heard. Although the unique idea of generating music from textual sources may contribute to the difficulty of creating "good" music, the limitations of algebraic systems can be considered. Since musical composition requires specific components and rules regardless of a composer's style and genre, it is intuitive to program a set of algebraic operations to model such conventions. In a practical perspective, having such algebraic operations may not be a robust solution due to the high number of musical components and rules one must consider. Therefore, it is challenging for algebraic systems such as *TransProse* to generate appealing music when given the difficulty of defining, programming, and refining the numerous components and rules in musical composition. This ultimately leads to the suggestion that a model capable of generating music without having to explicitly program various musical components and rules would be a more plausible approach.

2.2 Markov-based Musical Composition

Given the aforementioned difficulty of explicitly programming various musical components and rules, one may consider statistical models based on Markov processes to computationally generating music. The work proposed by Herremans et al. can be discussed as an example [7]. Herremans et al. implemented a variable neighborhood search algorithm (VNS) [11] modified for musical purposes [4] that randomly explores a sequence of notes of the bagana (a ten-stringed box-lyre) until that sequence is evaluated to be plausible by various Markov models used as quality metrics constructed with the transition probability of notes in existing music. The first metric

Herremans et al. used to evaluate the quality of music generated by a VNS algorithm is based on the cross-entropy and the Markov models proposed by Forbood and Schoner [1] and Lo and Lucas [5]. Specifically, Herremans et al. used the equation below to measure the cross-entropy of the transition probability of notes (e_i) explored by a VNS algorithm. Having this metric allowed Herremans et al. to program a VNS algorithm to choose a sequence of notes with high statistical likelihood since the cross-entropy would be minimized when the transition probability of notes is close to 1.0.

$$h_i = -\log_2 P(e_i \mid e_{i-1}) \text{ [7]}$$

Another Markov-based metric in which Herremans et al. used to evaluate the music generated by a VNS algorithm is based on the work proposed by Davismoon and Eccles [3]. Davismoon and Eccles suggested that a generated output would replicate a given model when the Euclidean distance between the transition matrices of the given model and generated output is minimized. Thus, having this model allowed Herremans et al. to evaluate whether a sequence of notes explored by a VNS algorithm is a viable sequence containing patterns in existing music. The equation below shows the objective function Herremans et al. used to minimize the difference between the transition probabilities of notes in a given music and generated music denoted as $P(b \mid a)$ and $\bar{P}(b \mid a)$ respectively.

$$f(s) = \frac{1}{N}\sqrt{\sum_{a \epsilon \xi}\sum_{b \epsilon \xi}(P(b \mid a) - \bar{P}(b \mid a))^2} \text{ [7]}$$

Herremans et al. observed mixed results from using the Markov models introduced above to evaluate the music generated by a VNS algorithm. The figure below shows some examples of music in which Herremans et al. was able to generate. Herremans et al. acknowledged that the music in Fragment 1 of Figure 1 is plausible but may sound uninteresting since maximizing the statistical likelihood of notes resulted in generating a repetitive sequence of notes constrained within certain pitches. A less repetitive music was generated when minimizing the Euclidean distance between the transition matrices of a given music and generated music as shown in

Fragment 2 of Figure 1. In fact, Herremans et al. reported that the expert listener judged Fragment 2 of Figure 1 to be good.

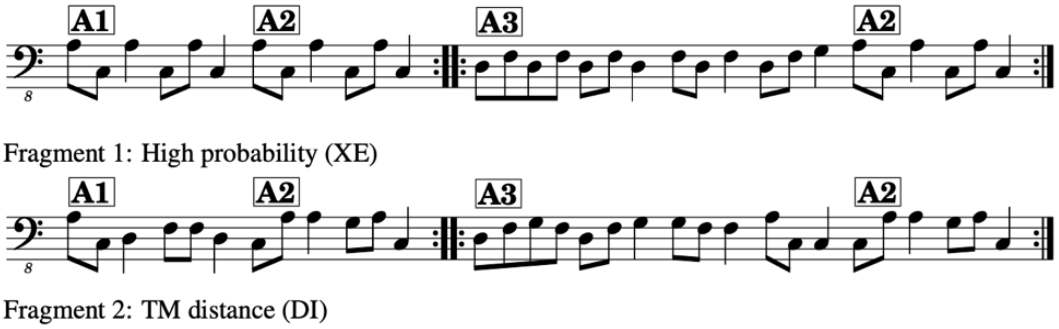

Figure 1. Musical output by using the three main evaluation metrics (XE, DI, and DE) [7]

Based on such musical outputs shown above, one may suggest that Markov processes is indeed a useful tool for generating a plausible piece of music with high statistical likelihood. The usability of Markov processes for generating music can be attributed to its ability to exploit the statistical patterns of existing music without having to explicitly program various musical components and rules. Therefore, one may suggest that Markov processes can be more usable than the aforementioned algebraic model having explicitly programmed musical components and rules. However, the music in which Herremans et al. generated shows that the tendency for Markov processes to assume a particular state to be dependent on *only* its previous state results in producing music with high statistical likelihood but with repetitive sequence of notes constrained within certain pitches. Thus, the inability to consider the interrelated patterns between notes at different time-steps may explain the repetitiveness in which Herremans et al. reported from Fragment 1 of Figure 1 and other musical outputs not directly mentioned in this review. Moreover, the originality of music generated by Markov processes can be questioned since the maximizing the statistical likelihood of generated music would merely replicate the statistical patterns in music that already exists rather than discovering new, unique musical patterns.

2.3 Deep Learning-based Musical Composition (*MuseNet*)

Besides Markov processes, deep learning is another popular statistical practice that can be used to generate music. OpenAI's *MuseNet* can be discussed as an example [9]. To understand the

mechanism of *MuseNet*, consider the conditional probability mentioned from OpenAI's paper that is shown below.

$$P(s_n \mid s_{1,\ldots,}s_{n-1})\ [9]$$

Unlike Markov processes that consider a current state to be dependent on a single state from the past, the conditional probability above means that a current state (s_n) is dependent on its context represented by a set of states from the past ($s_{1,\ldots,}s_{n-1}$). Thus, one would have to encode a representation of the set of states from the past and decode that representation to predict the state that is likely to follow. *MuseNet* performs this practice to generate music by having a transformer language model called GPT-2 [8]. Although the architecture of GPT-2 has some changes applied to the model proposed in [2], the general architecture of GPT-2 can be understood by referring to the figure shown below.

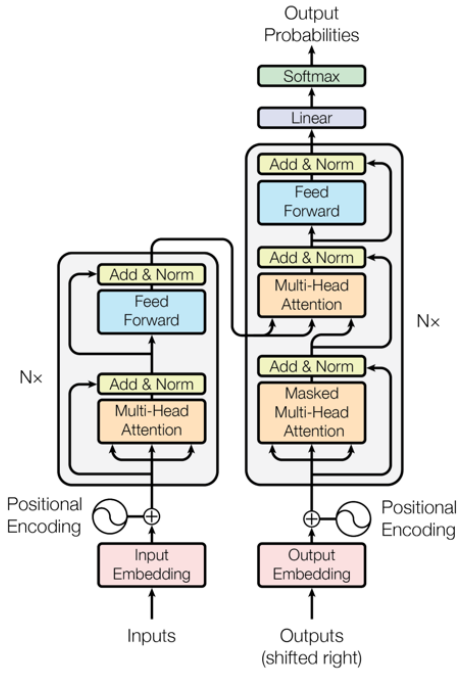

Figure 2. Architecture of the transformer model [2]

The mechanism of GPT-2 shown in the figure above can be explained as follows. When given a sequence of states as an input, the multi-head attention layer in the encoder of GPT-2 computes a

representation of the input sequence by relating different states at different positions. Subsequently, a feedforward neural network and residual connection are used to extract and preserve the patterns in the positional relationship between states in the input sequence. The encoded input sequence of states is then delivered to the decoder of GPT-2 having additional multi-head attention layers, a feedforward neural network, and residual connections used for predicting the output probabilities of states that may follow the input sequence of states. Although GPT-2 was originally implemented for natural language processing tasks, it can be applied to the context of musical composition since musical elements at multiple time-steps are interrelated to each other as in words used in any natural language. Thus, the use of GPT-2 enables *MuseNet* to predict the next possible "token" of musical elements (notes, harmony, rhythm, style) by encoding the positional relationship between multiple musical elements at multiple time-steps in a given tune.

The main task in which *MuseNet* performs with the use of GPT-2 is generating a piece of music that mimics, transfers, and synthesizes the styles of various musical genres and composers. For instance, *MuseNet* can be trained to compose music written in the style of Chopin when given Mozart's Rondo alla Turca as the prompt tune [12]. In fact, the quality of music composed by *MuseNet* is arguably better than the music composed by the aforementioned algebraic model and Markov-based model; style mimicry, transfer, and synthesis are effectively performed, musical errors are difficult to be detected by the average listener, and a sense of musical development can be heard. One may attribute *MuseNet*'s results to the well-known characteristic of deep learning. Specifically, the data-dependent approach of deep learning enables *MuseNet* to extract complex patterns in existing music without having to explicitly program the various musical components and rules. Although the aforementioned Markov-based model is indeed a data-dependent approach, its effectiveness can be challenged by the deep learning methods enabling *MuseNet* to exploit much more complex patterns and relationships between multiple musical elements at multiple time-steps. Nevertheless, the deep learning methods used in *MuseNet* still has its own weaknesses caused by its data-dependent approach. Although learning the correlations in large datasets of existing music is what enables *MuseNet* to generate plausible musical output, it is a computationally exhaustive process requiring millions of parameters of the encoding and decoding deep neural networks to be optimized through stochastic gradient descent. Moreover, the data-dependent approach of deep learning also poses a challenge to the originality of the music generated by *MuseNet*. Just like the aforementioned Markov-based model using the transition

probability of notes in existing music, *MuseNet* also extracts patterns from music that already exists and what it was given to be trained with. Thus, what *MuseNet* composes is merely a rearrangement of musical elements with high statistical likelihood rather than an entirely new piece of music with a unique style.

3 Conclusion

An algebra-based model, a Markov-based model, and a deep learning-based model for generating music was discussed in this review. A general observation of the music generated by the three models discussed in this review suggests that musical composition is an exercise that can be modeled computationally and therefore is not necessarily an ability exclusive to certain talented human beings. When comparing the three computational models discussed in this review in terms of their practicality and effectiveness of generating music, one may suggest that the statistical methods exploiting patterns in existing music, such as a Markov-based model and a deep learning-based model, are able to compose more complex and appealing music than what an algebraic model having explicitly programmed musical components and rules is capable of. One may also suggest that the ability to perform the complex task of replicating, transferring, and synthesizing musical components by exploiting a set of patterns in existing music makes deep learning computationally exhaustive but more robust than Markov processes. However, all three types of models for generating music discussed in this review are yet to be able to create an entirely original piece of music and style. Such a limitation can be attributed to the inevitable dependency on existing rules, patterns, and data. Thus, further work in computational models learning through some degree of randomness and less dependency on existing data may challenge the continuing controversy of whether musical creativity is a quality exclusive to human beings.

References

[1] M. Farbood and B. Schoner, "Analysis and synthesis of Palestrina-style counterpoint using Markov chains," *ICMC*, 2001.

[2] A. Vaswani *et al.*, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, p. 2017.

[3] S. Davismoon and J. Eccles, "Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures," *European Conference on the Applications of Evolutionary Computation*, p. 361, 2010.

[4] D. Herremans and K. Sorensen, "Composing first species counterpoint with a variable neighbourhood search algorithm," *Journal of Mathematics and the Arts*, vol. 6, no. 4, p. 169, 2012.

[5] M. Lo and S. M. Lucas, "Evolving musical sequences with n-gram based trainable fitness functions," *IEEE International Conference on Evolutionary Computation*, p. 601, 2006.

[6] H. Davis and S. M. Mohammad, “Generating music from literature,” *CoRR*, vol. abs/1403.2124, 2014, [Online]. Available: http://arxiv.org/abs/1403.2124
[7] D. Herremans, S. Weisser, K. Sorensen, and D. Conklin, “Generating structured music for bagana using quality metrics based on Markov models,” *Expert Systems with Applications*, vol. 42, no. 21, p. 7427, 2015, doi: https://doi.org/10.1016/j.eswa.2015.05.043.
[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
[9] C. Payne, “MuseNet,” *OpenAI*, 2019, [Online]. Available: openai.com/blog/musenet
[10] S. M. Mohammad and Turney, “NRC Emotion Lexicon,” *National Research Council, Canada*, vol. 2, p. 234, 2013.
[11] P. Hansen, N. Mladenovi, and D. Perez-Britos, “Variable neighborhood decomposition search,” *Journal of Heuristics*, vol. 7, no. 4, p. 335, 2001.