David Mash

3 February 2023

No Composer-Pilot:

On Machine Learning and Generative Music

Abstract:

The beauty of generative music is not so simply understood. To the layman, machine learning algorithms and generative music systems are a clear match-up that is bound to blend well, but to those that understand the history, art, and science of generative systems, there is a different story occurring. Starting with some of the simplest generative music systems that exist, like a set of wind chimes on the porch, then touching on the generative nature of the fugues of Bach, this essay explains how the long history of generative music has come to explode into boundless possibilities with the advent of algorithmic music and artificial intelligence. But where machine learning might seem like the next logical step, it may be uprooting an entire art form. This essay details some blaring problems for the future of generative music artists while not shying away from the new possibilities that machine-learning approaches open up to multi-faceted musicians up to the task.

Introduction

The field of machine learning and the art of generative music may seem like a match made in heaven. After all, the main goal for generative music artists is to create a system that can compose music on its own, which seems to imply some level of computerized autonomy as is seen in machine learning. Contrary to this assumption, though there are many ways to utilize machine learning in the arts, and indeed methods to train a machine learning algorithm to produce new compositions, I am prepared to argue that, ultimately, machine learning is not as effective as an artistic tool for musicians as other methods have proven themselves to be. However, in order to understand what makes a generative system effective for artists, we need to first have an agreeable baseline for what generative music really is. In this paper, I will explain the state of generative music, in terms of its scholarship, history, and modern use. I will then, utilizing scholarship in the field of artificial intelligence (AI) as well as the existing applications of machine learning to music, demonstrate the differences between what has been achieved in the field and what is possible with machine learning. In the end, this paper identifies machine learning approaches to generative music as a divergence for the art form, and perhaps a negative one for generative music artists; however, its potential uses in other aspects of art and music are recognized and appreciated.

Generative art has been defined by AI and philosophy scholar Margaret Boden as “work that has been produced by the activation of a set of rules and where the artist lets a computer system take over at least some of the decision-making (although, of course, the artist determines the rules)” (Boden et al. 2009, p.4). Applied to music, this definition is further expanded when we consider the fact that the “computer” aspect of the system does not necessarily have to be the part that executes the composing process, but that the composing process is determined by a set of rules. With this expansion, generative music can also include physical and even conceptual systems as methods for creating generative music. Thus it is not a stretch by any means to view process music, like the works of Steve Reich, or even the fugues of Bach, as examples of generative music in their own right. Another generative art scholar, Philip Galanter, has gone so far as to say that "while it shouldn't be terribly surprising that the earliest forms of generative art used simple systems, some will find it surprising and perhaps even controversial that generative art is as old as art itself" (Galanter 2003, p.12). Indeed, even a set of wind chimes can be considered a generative system, in which case the craftsperson of the chimes can be considered the artist behind the process (Brown 2005 p.217). Represented below in figure 1, we can see that an artist's relationship to the music generated by their system is not overly complicated and easily attributable.

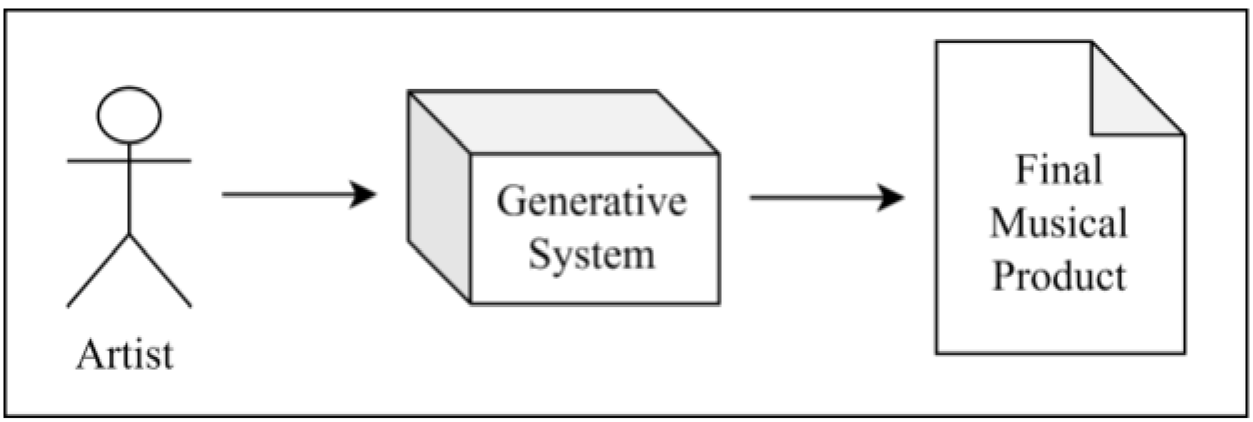

Figure 1 Representing the artist's relation to the final product in generative music.

The Implications of Computer-Generative Music

Computers are the most commonly used mediums for generative music today, which should come as no surprise, as they provide artists with closed and infinitely re-creatable environments that enable the exploration of much more complex generative systems than ever before. Artists can now utilize massive sets of highly refined and meticulously tweaked digital parts in the form of different algorithms and data structures to build generative systems that have rule sets that go far beyond the bounds of those found in traditional composing and the execution of these rules is fully automated, so unlike Bach writing his fugues, modern generative artists do not have to calculate, write, or follow their own ruleset in any ways in order to generate a composition. The introduction of computerized generative systems did not diverge generative music from its fundamental art form but allowed its possibilities to explode into further boundlessness. I am fond of how Nick Collins, a modern generative music artist, and scholar, describes the expressive abilities of computerized, algorithmic generative systems in his 2008 paper, *The Analysis of Generative Music Programs*.

> "Algorithmic music is compositional design at a metalevel, human creativity in musical representations, examination of particular rule sets in a space of multiple music theories, with the composer–designer–musician becoming a 'composer-pilot' through musical modeling space. Composers model composition itself, and such systems give us valuable insight into the relations of music theory, musical design and aural instantiation" (Collins 2008, p.239).

Collins attributes his usage of the term "composer-pilot" to Iannis Xenakis's paper *Formalized Music*, in which Xenakis explains that in quantifying the rules of music in a system, an artist must determine the "certain uncertainties" that constitute music composition itself (Collins 1992, p.144). The elevated level of intricacy made possible by algorithmic music has created an expressive medium for generative artists that is truly meta-compositional, enabling multi-faceted engineer-musicians to craft systems that produce music based solely on one's musical identity and subjective compositional reality. This kind of algorithmic generative music is something I will refer to as CG-music, as in “computer-generative music,” which is inspired by Boden’s use of the term “CG-art” in her works (Boden et al. 2009, p.10).

CG-music has been explored in a plethora of projects with varying methods of implementation. On the more conservative side, we have Gustavo Díaz-Jerez’s generative system, Nodal, which produces music by stepping through user-created graphs, in which the nodes, or vertices of the graph, hold note information, like pitch, and the connections between the nodes, or edges of the graph, express other musical information, like timing (McCormack 2007, p.5). There is also Peter Chilvers’s and Brian Eno's generative system, Bloom, which is a mobile app that responds to user gestures to create musical patterns but can also make music completely on its own (Siegal, 2018). On the other side of the complexity spectrum, George Lewis created a groundbreaking generative system that he called Voyager, back in the 1980s. In his paper, “Too Many Notes: Computers, Complexity and Culture in ‘Voyager,’” Lewis notes that his system is "not asking whether machines exhibit personality or identity, but how personalities and identities become articulated through sonic behavior” (Lewis 2000, 38). He argues that Voyager’s sonic behavior has embodied the “African-American aesthetic and musical practices” (Lewis 2000, 33). Though Lewis holds the code behind the system rather tight to the chest, it is clear from its abilities that he has created an AI that functions over a specific compositional ruleset defined by his personal understanding of musical grammar.

Staying in the 80s with the 1981 paper, *Using Generative Grammars for Music Composition*, by S. R. Holtzman, we learn that similarly to linguistics, the processes that musicians use for traditional music composition can often be expressed as a set of grammar rules. Holtzmann explained how one could use a Generative Grammar Definition Language (GGDL) to express rules for music composition on both a macro and micro level. A GGDL represents generative processes by expressing rules for transforming inputs (Holtzmann 1981, p.52). These transformations can be simple replacements and transitional matrices, or more complex structures, like Markov chains, or really any other applicable high-level function. The output can represent any kind of musical object or structure, like an entire section of music, a note's length, a sound envelope, etc. Once an artist has defined all of the macro and micro level rules that make up their generative grammar, they only need to apply those rules to code, which creates a form of handcrafted artificial intelligence, like that of George Lewis's Voyager.

Good Old Fashioned Artificial Intelligence vs. Machine Learning

There is a widespread misconception that the term artificial intelligence (AI) implies the presence of a neural network or machine learning (ML) algorithm. It is important to understand that ML is a subcategory of AI and that ML is not present in any of the CG-music systems I mentioned earlier or in the process of turning musical grammar into AI. The kind of AI used in those examples, where it exists, is referred to as symbolic AI, or Good Old-Fashioned AI (GOFAI) because it is a classical approach to AI that does not use the relatively newer concepts of neural networks and machine learning in its processes (Boden 2014 p.89). In a typical GOFAI, probabilistic selections are made from symbolic data structures and tables that represent solutions to fractional problems that, together, result in a final solution. Programmers often use expert opinions to transform these symbolic data structures to make their probabilities favor the accepted correct answer. In her essay included in *The Cambridge Handbook of Artificial Intelligence*, Margaret Boden talks about the fundamental concepts behind GOFAIs:

> "The program is provided before it starts with a number of possible differences, a list of actions (operators) that can eliminate the various differences, lists of prerequisites that must hold if a certain operator is to be used, and heuristics for

ordering the actions when more than one action is possible in the current circumstances. Indeed, most of the 'intelligence' involved lies in the choices of actions, operators, and heuristics specified by the programmer." (Boden 2014, p.90).

The important thing to focus on in Boden's description here is the recognition that the programmer and creator of the system is the one that determines the choices, operators, and heuristics available to the generative process. This relationship is illustrated below in figure 2, which shows that the use of GOFAIs to create CG-music may increase the complexity of the system, just like any algorithmic music generation would, but that at a metalevel, it does not interfere with the artist's relation to the final product nor the artist's control over the system.

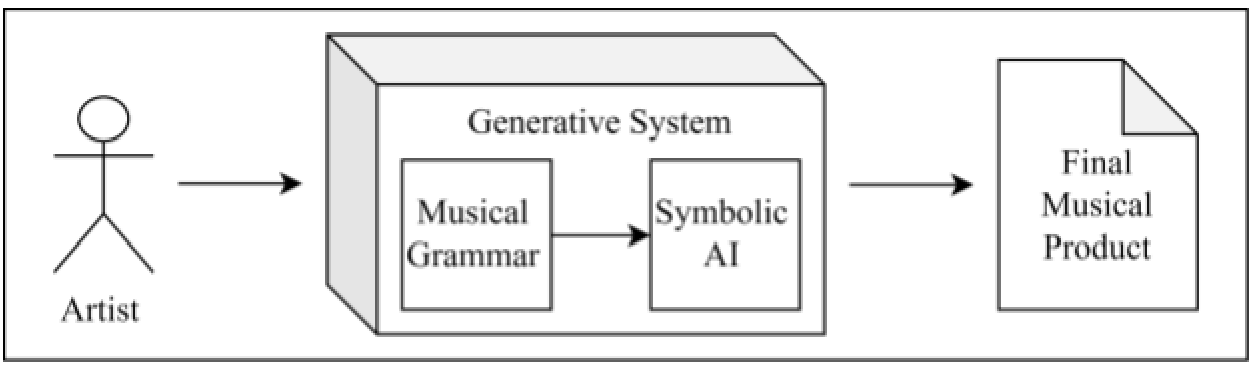

Figure 2 Representing the artist's relation to the final product in generative music with the use of musical grammar analysis and symbolic AI.

Symbolic AI and machine learning (ML) are often blurred as synonymous concepts when discussed by laymen, but in reality, they have since their conception remained two distinct categories of AI that differ in philosophy and implementation (Franklin 2014 p.15). In contrast to GOFAIs, machine learning (ML) approaches do not allow the programmer to decide what choices, operators, and heuristics the process should use. In David Danks's essay included in the previously mentioned AI handbook, he explains the fundamental concepts behind machine learning algorithms.

> “The algorithm works by extracting – and exploiting – structural relationships among the variables without regard to the meaning or domain of the variables. For example, if doing classification using an artificial neural network, one might be provided with a dataset containing measurements of various features of widgets, as well as some target category. The neural-net learning algorithm then uses only the statistical regularities in the dataset to learn the relevant inter-variable structure, which can then be used to predict the target category for future widgets. The precise ‘meaning’ of the variables is irrelevant to the learning algorithm” (Danks 2014, p.152).

There exists a variety of methods when it comes to ML algorithms, but in general, ML requires a training dataset, which in our case would be a collection of raw audio clips or, more likely, MIDI files of existing songs of which we want our algorithm to be able to recreate the style of. These samples would be processed by the ML algorithm to create a trained model, which is a neural network with its weights properly tuned to the properties of its training set. This trained model should hopefully, after some inevitable tweaking through trial and error, be able to identify common features among the samples of the training set (Danks 2014, p.152). If used correctly, a trained model could be used to recognize these features in new inputs or even imitate those features in their own synthesis. Thus instead of having the artist perform the meta-compositional design of the system, applying their musical grammar rules to a system, the ML algorithm performs the analysis and creates the model on its own, which is not based on a grammar at all, but simply a checkbox of features that are visible from the outside of compositions. This ML approach is divergent in this way and for the purposes of creating CG-music it actually acts to limit the bounds of compositional power to that of what exists in its training set. It takes away the decisions regarding its generative grammar and subsequent implementation. These were elements that made GOFAIs and other approaches boundlessly expressive for generative artists. ML has opened up some new artistic opportunities, but in terms of CG-music, it seems to be closing doors, removing the historic and unlimited artistry of generative music as it marches toward becoming synonymous with machine learning generated music.

In 2016, Google launched Magenta, an art-focused AI model based on TensorFlow, which is their proprietary ML algorithm. There are many examples of computer scientists using Magenta to do all sorts of interesting things with sound. One of my personal favorites is the NSynth project, which dynamically mixes sounds together and has created a new way for music producers to synthesize unique timbres (NSynth Team 2017). Projects like NSynth are meant to inspire artists to create new sounds and push the limits of their own compositions. Other projects use Magenta to make generative music systems. With varying success, these projects often achieve what they set out to do: train an ML algorithm on a massive dataset of music in order to have it output strings of notes such that it sounds like a human may have composed it. In fact, in 2016, there were blogs appearing about Magenta with titles like “The Beginning Of The End For Composers? Google Magenta Just Wrote Its First Song” (Sethi 2016). At the time, despite the excitement (and hysteria) around ML being used for music composition, the melodies produced by Magenta were short and obviously just imitations of the music it had analyzed in its training set (Sethi 2016). Now in 2023, Google has released details on its new music composing ML model, called MusicLM, which can generate genuinely human-like compositions from a simple text prompt. From a prompt not much longer than “the main soundtrack of an arcade game”, MusicLM can respond with what sounds like hours work of from a real videogame composer. The prompt can be as simple as that, or as detailed as necessary to get the output the user desires. It's honestly an extraordinary advancement in its field and it will certainly only get more and more advanced as time goes on, just as we have seen with chatbots, like OpenAI’s ChatGPT, but there remains a fundamental flaw from an artistic point of view. ML systems, like Magenta and MusicLM, understand music exclusively from the audible properties found in its training set and, as Danks explained, they do not care about and, in fact, have little to no regard for the “meaning” of the variables they process (Danks 2014, p.152). There is no meta-compositional understanding, only imitation. This ability to imitate is great news for managers, that need pop music now and do not want to pay a composer, but for musicians, and particularly generative artists, it can seem like doomsday is coming for their art.

In general, audiences perceive completed art far differently than the creators do. This can be largely attributed to the fact that viewers often do not have a complete picture of the self-imposed rules and contexts with which creators implement the objects of their work (Lefford 2021, p.129). When it comes to physical generative systems, the rules were never very difficult to understand. For example, in Steve Reich’s generative work, “It’s Gonna Rain”, the rules are clear: there are two identical recordings on tapes of slightly different lengths that are played at the same time which results in their contents slipping in and out of sync with each other, creating the final musical piece. Despite how abstract the final work is, listeners are grounded in the compositional techniques and rules that guided the execution of the piece. When the process gets more complex, as with algorithmic systems, and the compositional rules head toward invisibility for listeners, informed audiences can still be grounded in the humanitarian art of meta-composition that has allowed the artist to instill their musical grammar into the system, and when such an informed audience meets such a multifaceted musician, the genius of CG-music is appreciated for what it really is. When ML replaces the fragile connection between the artist and the final product, the transparency of the process is completely broken for both the artist and the audience, which disembodies the music, resulting in the audience losing all grounding and music that does not push any boundary, but exclusively conforms. This disembodiment of the final product is represented below in figure 3, in which the question mark (?) shows the unknown or debatable relationship between the artist and the system.

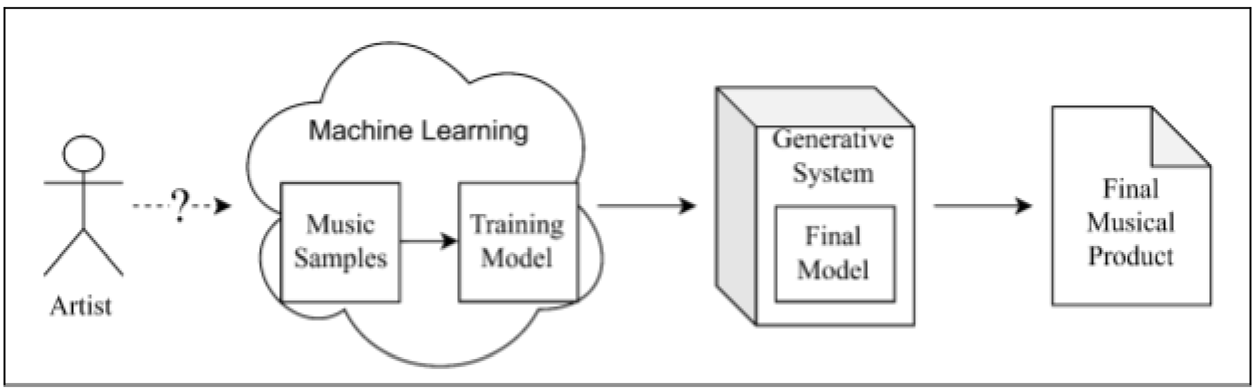

Figure 3 Representing the artist's debatable relation to the final product in music generated using machine learning.

Conclusion

The beauty of CG-music comes from "compositional design at a metalevel" (Collins 2008 p.239) and not simply from the presence of a system that can imitate compositional features. Because ML works by identifying "statistical regularities" in a dataset (Danks 2014, p.152), when used for art or music, it creates imitative work, that is also disembodied from its artist, alienating its audience. In this way, the heightened expressive power of CG-music is lost through the over-utilization of machine learning.

Machine learning may be diluting the art of generative music, but I am not in any way a Luddite, nor some kind of technophobe. Though it isn't as expressive for artists of generative music, ML can still be used as a powerful tool in the creation of art and music. For example, Google's Magenta was used by the band, YACHT, to create unique sounds that they have used in their music. They fed the learning algorithm with a dataset of samples from their released music and Magenta outputted melodies and lyrics that they were able to pick and choose from to compose music for a new album (Friedman 2019). Magenta has also enabled the creation of NSynth, as mentioned earlier, whose team has made it clear that rather than trying to use ML to replace some human function in music, their goal is "to aid the creative process" (NSynth Team 2017). There are boundless possibilities for how machine learning can be used by musicians in their work, though it may start leaving many such musicians without work in the near future.

Without machine learning, generative music as a genre has found ways to constantly evolve also. The "live coding" community has become an active place where generative music continues to develop. Live coding is the practice of actively programming and modifying a generative system while music is being produced. During performances of live-coded music, the artist typically shares their code with the audience as they write it by projecting it on a large screen (Brown et al. 2009 p.18). This extreme level of transparency lets the audience know exactly what is happening as the artist builds the process right before their eyes, which gives this form of CG-music an ideal amount of shared artist-audience context (Lefford 2021 p.130) - the type that made physical generative systems extremely approachable, even to inexperienced audiences. These systems can be very simple, triggering a drum pattern through some hard-coded digits, or as complex as any AI, so the potential is once again boundless for our live-coding composer-pilots. For the future of generative music as an art form, I believe that artists in the live coding community will be our best hope for carrying on the torch as we head toward a world with exponentially more advanced machine learning AI.