The Role of Artificial Intelligence in Music Composition

David Barton Harris
April 11, 2019

Advances in music technology have automated processes that were once the sole responsibility of the human composer. It may be tempting to view these advances as proof that machines can compose music independently, however this is not the case. Humans must design, program and interface with artificial intelligence (AI) composition models to achieve any musical result, much less a good one. This paper will introduce three types of AI composition models and demonstrate the role humans play in each. Oliver Sacks defines music as a construction in our minds "using many parts of the brain" by the integration of "tones, timbre, pitch intervals, melodic contours, harmony, and ... rhythm." (Sacks 2008, xi). This definition is used here because it frames music as a product of the human brain; the point being that even if sound could be generated solely by a computer, if it has no human listener, no audiator, it cannot, by definition, be music.

The three AI models explored in this paper use algorithms, stepwise processes for solving problems. Computers are programmed to perform algorithms. As such, it is possible to assume an algorithm would produce only robotic results. History tells us this is untrue for it is undeniable that algorithmic music has existed far before the invention of modern computers or any discussion of artificial intelligence. In fact, any compositional technique is arguably algorithmic. In his book *The Algorithmic Composer,* David Cope enumerates many algorithms that defined musical eras. Fourteenth-century isorhythmic compositions had strict rules on the incorporation of "*color* (set of intervals) and *talea* (rhythmic pattern)." Isorhythmic motets by Guillaume de Machaut were finalized when their patterns coincided, occurring when the repetitions of color and talea were extended a number of measures equal to the least common multiple of both their durations. (Cope 2000, 3). Cope also demonstrates compositional algorithms in the contrapuntal practices codified by Johann Joseph Fux in 1725, twelve-tone serialism defined by Schoenberg in the 1920s, and aleatoric instructions by John Cage in the 1950s, to name a few. (Cope 2000, 6-12). All of these step-by-step compositional practices, even those that include chance, can be translated to computer code with relative ease. Did these compositional algorithms limit creativity in their respective musical practices? Stravinsky would argue they limited choice, but not creativity:

> [M]y freedom will be so much the greater and more meaningful the more narrowly I limit my field of action and the more I surround myself with obstacles. Whatever diminishes constraint, diminishes strength. The more constraints one imposes, the more one frees one's self of the chains that shackle the spirit. (Stravinsky 1947, 65).

Furthermore, algorithms can provide a puzzle-like enjoyment for the listener. There would be far less impetus to listen to serial music if knowledge and appreciation of its algorithmic construction were absent. To enjoy a Webern piece is to take pleasure in the unfolding of a set pattern. It is this paper's position that computers cannot independently compose music, but this is not because they use algorithms.

The preceding non-computer algorithms were provided to demonstrate that the operations of machine and human composers are not, in fact, so distinct. The artistic inspiration we may have once attributed to Plato's muses will vanish to accommodate new mysteries. The magician's trick will be revealed, only to make room for new tricks. Composers of today have a choice; they can lament that their power has been demystified or they can acknowledge that discovering the science underpinning their craft raises the bar of artistic merit. After all, the algorithms have shortcomings, which demand that humans incorporate into their works a quality called *taste*. In this sense taste is not an ethereal additive quality, but rather a subtractive process; the discernment of when to break from an algorithm.

Multiple methods of computer composition exist, yet none of them can generate music without some element of human input. This input can be a rule set in the case of symbolic AI models, a fitness feedback loop in an evolutionary model, or an initial dataset in a neural network. These three models of computer composition are defined and explored below.

The oldest and most basic model of AI composition is called symbolic AI and is any approach "concerned with hard-coding a set of rules, which prescribe the behaviour of the machine." (Miranda, Williams 2015, 77). The essence of a symbolic method of composition is that musical best practices are written prior to the running of the computer program, and the output of the program envisions a variety of realizations of these practices. These could include voice leading principles like the avoidance of parallel fifths or a preference for stepwise melodic motion over large leaps, but could also be extended to non-pitch considerations such as form, dynamics, and rhythm. Musical output from such programs would seem to owe its authorship to the designer or programmer of the rule set, not the computer.

However, not all symbolic AI programs have a predetermined rule set. Some may derive their rules from input material (a trait they share with neural networks, but functioning in a different way). That is, a program could infer Chopin best practices by scanning a number of Chopin pieces and finding their commonalities. David Cope's Experiments in Musical Intelligence (EMI) is one such program. EMI can emulate the style of a particular composer when given a set of his or her works as input. In fact, when EMI was fed all the Mozart piano sonatas, its output sounded so much like Mozart that "even the most trained classical musician has identified the master as its author." (King 2007, 0:55) This raises some intriguing philosophical questions. Is Cope the author of this piece or *is Mozart*? This would bring a whole new meaning to the concept that an artist's works live beyond his death. Maybe this suggests an artist's works can be born after his death. Perhaps authorship is shared between Cope and

Mozart in a fashion similar to when audio samples are often credited when reused and reimagined in hip hop. These programs seem to have the power to create nightmarish copyright issues as well as requiring a new definition for the term “composer.” One conclusion is definite however; it is untrue to say the computer output has absolutely no human author.

One of David Cope’s composition programs, *Melody Predictor,* takes as input just a few bars of a classical melody, but predicts to a high degree of accuracy, the subsequent notes. The program was particularly successful with Mozart’s music:

> While we might imagine that each note of Mozart’s music was an inspired original, the fact remains that a great deal of his music, like most of his contemporaries of the classical period, follows algorithms and that these algorithms produce relatively predictable results. (Cope 2000, 10).

The algorithms may not even be of the composer’s design. They may be borrowed or inferred from the common practice or from one or many tutors, which seems to imply a diminishing of Mozart’s artistic power. However, Cope caveats that the most inspiring of Mozart’s choices originate from breaking out of the algorithms he uses. The frequency of these breaches and how they are executed is the artistic challenge that computers seem to lack, although randomness has been utilized to approximate this in a number of AI compositional programs. But even if a computer succeeds at breaking algorithms in a way satisfying to human ears, it has still been programmed by a human to do this, and as such, that human deserves part, if not all, of the authorship of the works.

The next two approaches toward AI composition are considered biological methods. The first of these, the evolutionary approach, is demonstrated in a 2001 program called *MutaSynth* by composer and programmer Palle Dahlstedt. *MutaSynth* uses a reproductive model that starts with a set of synthetic aural possibilities in a matrix and generates a random (insofar as any music confined to a set of timbres and within a matrix can be considered random) source material. A human listener then decides which parts, if any, she prefers. This selection process produces “parent sounds,” which, through a process called “mating” are combined into other preferred sounds. The result of this mating, the “child,” is again subject to human feedback and the cycle perpetuates. Throughout this process *MutaSynth* also introduces random musical “mutations” which are evaluated by the human listener for their musical “fitness.” Unfavorable mutations are discarded and not given the chance to mate. (Dahlstedt 2001, 123-24).

This approach is highly dependent on human input, as in the relatively confined parameters of sound generation (*MutaSynth* partitions its components into musical roles such as “drum pattern,” “bass line,” and “melody”), the finite range of mutational expression (no mutations beyond the synthesizer’s capabilities are possible), and of course, most notably, in its dependency on the human selection of favorable musical traits. In this latter way it reduces the role of the human using this program from musical originator to musical rater. Of his program Dahlstedt wrote, “I have a slight feeling I did not write that music, and yet I am quite sure no one else did. I designed the algorithm, implemented it and chose the parameters, and still I feel alienated.” (Dahlstedt 2001, 122). The evolutionary model has been at work on music since its dawn, just on a much larger scale. Have not whole cultures favored certain practices, discarded others and introduced mutations in the field of music for centuries? One interesting philosophical consideration implied here is that the authorship of a piece may be shared across all of humanity for we all take part in and are products of evolution.

The evolutionary model is a promising tool for developing musical ideas that may not readily come to mind for a composer. Like a thesaurus, which is a resource for content not known or present in the mind of a writer, but used to enable her craft, an evolutionary model may provide similar possibilities for the composer. The computations are an extension of human art, but by itself, the model cannot compose.

The neural network AI approach, sometimes referred to as machine learning, emulates the interconnected parallel structure of a brain, creating a robust, fault tolerant network, which can parse input into layers of abstraction. On March 21, 2019, the 334th birthday of Johann Sebastian Bach, Google presented on its homepage an application, affectionately called a "Doodle," which uses machine learning to generate Bach style harmonizations of short, user-inputed melodies. The Google Doodle team provides a layman's definition of machine learning as:

> … the process of teaching a computer to come up with its own answers by showing it a lot of examples, instead of giving it a set of rules to follow as is done in traditional computer programming. (Google Doodle, 2019).

The programmer of such an AI model does not code its program to resolve a dominant seventh downward; yet this practice is inferred from many examples. (306 chorales, as in the Bach Doodle.) Machine learning offers a more expansive and robust set of possibilities than hard-coding a rule-set, but requires a dataset instead. Human input is necessary to both.

StructureNet is the proprietary neural network program used by Jukedeck.com, a London based scoring company that "brings artificial intelligence to music composition and production." (Jukedeck R&D Team 2018). This deep neural network "learns about structure from a dataset consisting of structural elements and their occurrence statistics, which is created using a structure-tagging algorithm from an existing dataset of melodies." (Medeot et al 2018, 725). StructureNet requires some human input to provide a launch point. The composers of music generated with StructureNet are a combination of designers, programmers and users who provide this launch dataset, but not StructureNet itself.

RNNs (recurrent neural networks) feed their output back into their input and are thus said to learn from their own processing. "Unlike symbolic approaches, these systems can often continue to adjust their rule-sets, potentially developing further without continued human intervention." (Miranda, Williams 2015, 79). This might suggest the RNN approach has even more promise to generate musical content with less input from a human, but in practice, these programs tend to encourage more human choices, emboldening the perspective that such programs are tools of composition, not composers. Aiva, a Luxembourg based company that uses AI to score soundtracks for film and video games, employs a recurrent neural network algorithm, but as of yet, its musical output has been subject to significant human massaging. The company touts its rock song, "On The Edge" as "composed by Aiva," but also caveats:

- The bridge contained a melody, which we decided to remove altogether, to give room for the piece to breath and let the instrumentals shine

- It did not have any drums: those were manually added during the production process

- The rhythmic pattern of the verse's melody was awkward, so we proceeded to using our newly built melody generator to recreate a new melody from scratch, by specifying the chord progression of the verse to condition the creation process. (Aiva Technologies 2018).

The fact that Aiva's own creators and evangelizers feel the need to edit its output highlights the necessary role of humans in music composition. Additionally, initial datasets must still be selected by human operators. After all, a rock song bred from the input of U2 and Kings of Leon will sound quite different than one informed by Chuck Berry and The Rolling Stones. Also notable is that the human rated quality of pieces composed by RNN computation is higher in those with smaller datasets. If you want an RNN to sound like Debussy, you should not feed it everything from Pérotin to Reich, but rather a few Debussy pieces. Does this not reduce the role of the neural network machine from composer to that of emulator or worse, imposter? It is better that RNNs be considered powerful tools than ersatz composers.

The human input required by all the aforementioned AI methods disqualifies them from being composers in and of themselves. However, perhaps the question of whether or not AI can compose is the wrong one to be asking. Maybe the more important question is whether or not music created via these methods can pass a Turing Test? In his 1950 paper, “Computing Machinery and Intelligence” Alan Turing argued that gauging a machine’s ability to think was too complex a task. He proposed a simpler test for measuring artificial intelligence. Can a blinded human testee, presented with multiple text conversations, some from human sources and some from machine sources, determine which are which? If he cannot the machine has successfully passed the Turing Test. While his original idea tested conversation, it has been expanded to the realms of visual art and music. An artificial product is said to have passed a Turing Test if a human testee can determine whether its source is artificial or human no more accurately than chance. (Turing 1950, 433-34). If classical music experts attributed Cope’s EMI output to Mozart, has it not passed the Turing Test?

Thinking about Sack’s definition of music leads us to acknowledge that hearing has more to do with the human brain than with sound waves, which so often trigger our perception of music. This is the principle behind audiation, the sense of hearing music in one’s mind. As Sacks notes, this can occur even when no music is physically present:

> “Since the mid-1990s, studies carried out by Robert Zatorre and his colleagues, using increasingly sophisticated brain-imaging techniques, have shown that imagining music can indeed activate the auditory cortex almost as strongly as listening to it.” (Sacks 2008, 34).

This highlights how the human brain is the environment necessary for music to exist. Outside this environment sound is just the relative density and rarity of waves. John Cage stresses the role of the listener:

> Most people think that when they hear a piece of music, they’re not doing anything but that something is being done to them. Now this is not true, and we must arrange our music, we must arrange our art, we must arrange everything, I believe, so that people realize that they themselves are doing it, and not that something is being done to them. (Nyman, Cage 1974, 21).

This reinforces the idea that music is a personal experience within the mind of the audiator. How can a computer be said to compose if it has no capacity to audiate? It may be useful to imagine a sort of hierarchy of authorship as determined by audiation. If a piece has a human composer who audiated its content in the process of composition, he or she is clearly the composer. However, if the composer never audiated the work (say it was the output of a computer algorithm or an aleatory piece that is not determined until performance) the attribution of composition falls to the next audiator in the hierarchy, the performer and/or listener. Sound which has not been audiated is not music.

Computers can be programmed to algorithmically write pieces much faster rate than even the most masterful composers. At some point the technology allows this output to surpass the rate at which humans can listen to the output (even if dispersed throughout the entire human population). The authorship hierarchy would suggest that these pieces of sound will never graduate to music. By prioritizing audiation as an aspect of music we can still assert that a piece locked in a drawer forever, having never been performed, is still music as long as it was audiated in the mind of its composer.

Ultimately, AI cannot compose music by itself. Human involvement is needed in all of the computer composition models introduced in this paper. More importantly, the appreciation of sound is a requirement of music. Science has yet to discover exactly what permits this appreciation in the human mind, but it cannot argue that such appreciation is present in the algorithms of computers.

Aiva Technologies. “AI-generated Rock Music: the Making Of.” Blog. Last modified October 24, 2018. https://medium.com/@aivatech/the-making-of-ai-generated-rock-music-with-aiva-9ae0257e6d5c

Cope, David. *The Algorithmic Composer.* Madison, WI: A-R Editions, Inc., 2000.

Dahlstedt, Palle. “A MutaSynth in Parameter Space: Interactive Composition through Evolution.” *Organised Sound* 6, no. 2 (2001): 121–24. https://doi:10.1017/S1355771801002084

Google Doodle Team. “Celebrating Johann Sebastian Bach.” March 21, 2019. https://www.google.com/doodles/celebrating-johann-sebastian-bach

Jukedeck R&D Team. “Inducing Structure in Machine-Generated Melodies with StructureNet.” Research Blog. Last modified October 4, 2018. https://research.jukedeck.com/inducing-structure-in-machine-generated-melodies-with-structurenet-7bfeebb8bbf3

King, Cristina. “Tech Closeup: Music Professor.” Filmed 2007 in Santa Cruz, CA. Tech Closeup video: https://www.youtube.com/watch?v=yFImmDsNGdE

Medeot, Gabriele, Srikanth Cherla, Katerina Kosta, Matt McVicar, Samer Abdallah, Marco Selvi, Ed Newton-Rex, and Kevin Webster. “StructureNet: Inducing Structure in Generated Melodies.” (2018): In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR). http://ismir2018.ircam.fr/doc/pdfs/126_Paper.pdf

Miranda, Eduardo R. and Duncan Williams. “Artificial Intelligence in Organised Sound,” *Organised Sound* 20, no. 1 (2015): 76-81. https://doi.org/10.1017/S1355771814000454

Nyman, Michael and John Cage. *Experimental Music: Cage and Beyond*. University of California, CA: Schirmer Books. 1974.

Sacks, Oliver. *Musicophilia: Tales of Music and the Brain*. New York, NY: Vintage Books. 2008.

Stravinsky, Igor. *Poetics of Music.* Translated by Arthur Knodel and Ingolf Dahl. Cambridge, MA: Harvard University Press. 1947.

Turing, Alan. “Computing Machinery and Intelligence,” *Mind* 49 (1950): 433-460. https://www.csee.umbc.edu/courses/471/papers/turing.pdf