Algorithmic Music Composition: Survey, Critical View and Future Prospects

CS 333 Mini Project Report

gr90@duke.edu, taj26@duke.edu, ah450@duke.edu, ert26@duke.edu

1 Introduction

Music is one of the most significant achievements of the human race. The ability to produce a multitude of sounds, that have the power to invoke strong emotions is a remarkable feat. Since ancient times, music has fascinated mathematicians and philosophers alike. Pythagoras proposed that good music arises from having harmony with the laws of nature, and following the mathematical structures present in nature. Algorithmic composition is an attempt to formalise this endeavour of creating music into a sequence of instructions so that a machine may be able to compose music. Over the last few decades, a variety of approaches have been tried to compose music including Stochastic approaches, that involve randomness and can be as simple as generating a random series of notes, as seen already in the case of Mozart's Dice Music and in the works of John Cage, to more sophisticated computations through the computer with statistical theory and Markov chains.

Our purpose in this paper, is to merely explore this rich and fascinating area of algorithmic composition, examining two algorithmic approaches that are currently in use and to explore the direction in which the field is evolving.

2 Algorithm: Controlled Random Walks on Seeded Networks

Most traditional algorithms for music composition begin by analysing patterns in existing songs. While some approaches focus on theories of signal processing, in this example we will use a method built around network properties. While not as complex as some of the modern machine learning generation techniques, this approach illustrates a promising area of exploration at the intersection of music theory and graph theory.

While different cultures and genres will vary in the particulars of their songs, most human music is built off of progressions between harmonic and dissonant tones. Liu, Tse, and Small [1] observed multicultural similarities in the relative quantities of different notes within compositions. Notes follow a power-law distribution in their connections to other notes.

Liu, Tse, and Small begin their analysis by parsing MIDI files into weighted, directed graphs. MIDI files provide pitch start and end times over the course of a song and offer an ideal input for this analysis. We will denote a unique note i by the tuple (p_i, d_i), representing its pitch and duration. Each note is represented as a node in the graph G. When a note (p_j, d_j) at time t_j immediately follows note (p_i, d_i) at time t_i, a weighted edge is added to the graph, from i to j. If such an edge already exists, its weight is incremented. G includes different edges between i and j for each value of $t_j - t_i$. The authors demonstrate a sample output of this procedure in Figure 1.

Using this transformation, each MIDI song can be converted into a connected graph. They then combine multiple songs into a unified graph for each composer. In their analysis, they sample several classical pieces as well as albums from Chinese pop music.

In all graphs, Liu, Tse, and Small found that the mean shortest distance between unique nodes was close to 3 steps, and the clustering coefficient for the nodes in the network remained around 0.3. Using the relationship of the strength of a node (defined as the sum of its connected edge weights) with its de-

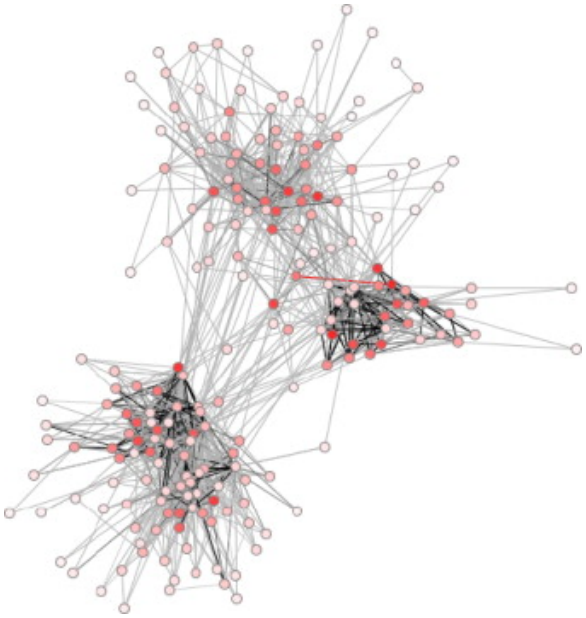

Figure 1. A sample network from Bach's violin solos. Darkness of colouring of nodes indicates relative degrees, and darkness of colouring of edges indicates relative weights [1].

gree,the authors observed a common power-law relation for each composer, with β values close to 1.2 in:

$$s(k) \approx k^{\beta}$$

While observing these similarities, Liu, Tse, and Small also noted that there were wide variations in the tempo, number of nodes, and number of edges for the different genres.

Using these observed properties, they propose a simple Markov chain algorithm to algorithmically generate music from a seeded network. We begin by starting at a random node. We will then randomly proceed to a connected node via an outgoing edge, with the probabilities of the different edges proportional to their edge weights. This process continues until a node with no outgoing edges is visited or until a desired note quantity is reached.

The result of this process is a melody that is based on a particular composer's work. The quality of the output varies substantially based on the diversity of the input. With a smaller sample size, the output tends to align closely to the original source. With a larger collection of songs, the output produces distinct melodic strings that do not harmonise with one another, but which tend to function as discrete pieces.

While straightforward, this process lacks many of the higher level grammars that define human-made music. While some cadences exist based on the MIDI input, there is little representation of motifs or phrasing. This particular method of music composition lacks the charm of machine learning systems trained on large datasets, but it provides a frame for analysis and generation with deeper insight into the underlying musical patterns [2].

3 LSTM Recurrent Neural Network Algorithm: Melody RNN

Another powerful way to generate music which relies on machine learning and deep learning is to use a recurrent neural network trained on existing melodies. Primarily such an approach, means that our algorithm acts upon a given training data to detect patterns of notes and then returns a new melody constructed based on the input data. Specifically, we consider a model from the HelloMagenta library that applies language modelling to melody generation using an LSTM called melody RNN. [3]

An LSTM RNN or a Long short-term memory is an artificial recurrent neural network architecture that uses feedback connections to process entire sequences of data (such as speech or video). While the specifics of this are complicated, a simple way to understand the architecture is that the LSTM RNN basically takes some sequential data as input (most commonly as a vector) and applies a transform on it so as to in some sense preserve temporal properties of data. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate and the cell remembers values over arbitrary time intervals while the three gates regulate the flow of information into and out of the cell.

LSTM networks help in classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. In order to process our data, we use a basic one-hot vector encoding to represent extracted melodies as input to the LSTM.

For the task, all melodies are pre-processed so that each melody is set up as a vector based on the MIDI pitch range [48, 84] and therefore the training

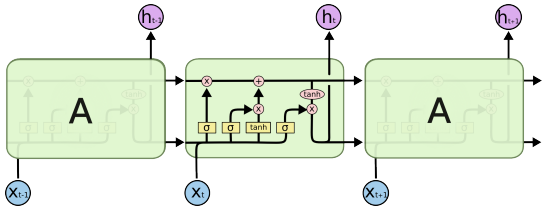

Figure 2. The repeating module in an LSTM contains four interacting layers.

data is reduced to a sequence of one hot encoding vectors.

The task of melody generation therefore is now essentially reduced to finding repeating patterns of vectors in the training data- a task which from an intuitive sense is also at the heart of predictive text. This configuration acts as a baseline for melody generation with an LSTM model that uses basic one-hot encoding to represent extracted melodies as input to the LSTM. [3]

3.1 Extra details on LSTM

Let us now try to understand the LSTM based structure. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram in Figure 2.

The cell state runs straight down the entire chain, with only some minor linear interactions and essentially acts as the prime way in which information is transferred unchanged. However, the LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates which regulate how information is let through. Gates are compose of sigmoid neural net layers and point wise multiplication operations and essentially the sigmoid function acts as a graded switch to decide how much of any value is to be allowed. In the specific case of generating a sequence of notes, this might be to decide which subsequent note to allow based on all the notes generated so far, and the model might suggest a sigmoid value of 1 to fully allow a note it believes is most likely at this point.

4 Obstacles and Challenges

While the machine learning for music generation has made tremendous strides in recent years, there are many technical and ethical limitations that create bottlenecks to its growth.

4.1 Technical Limitations

One problem with machine generated songs today is the ability to generate outputs with lyrics that make sense and are not offensive. Algorithms today are already used to create songs without words, but when it comes to lyrical generation, the nuances of language and the informal versions used in music make it difficult for programs to learn and replicate. Attempts have been made to solve this obstacle, such as conditioning the algorithm by feeding it linguistic features of a specific language alongside input music, but the lack of rules in informal speech makes it incredibly challenging. Another issue machine generated music faces today is the ability to create songs with structure (referring to pieces long enough to contain repetitions/chorus and separate verses). Outputs of today's algorithms for music generation may produce music without the sense of direction listeners are used to hearing from songs. A number of approaches have been explored to improve upon this aspect of music generation, including manipulating the input data, using a hierarchical architecture involving multiple RNNs, and unit selection (selecting each part of a song separately based on semantic relevance), but all have encountered challenges such as homogeneous outputs and inefficiencies in storage or retraining time. Especially with the variety of music structures seen across different genres and cultures, a more complex approach to this problem may be necessary.

4.2 Ethical Limitations

An obvious problem that may arise with machine generated music is creating a song that may be too similar to an existing piece of work. This is always an issue with any data-driven algorithm, as outputs may mirror the characteristics of the input too closely, but for the case of music generation, it is a legal copyright issue. As of now, a few approaches have been taken to try and resolve this concern. There are some researchers that believe that "creativity" can only be assessed by a human, leading to Tur-

ing Test-like approaches for evaluating the creativity of algorithms and their outputs. Other algorithm creators have defined self-assessment frameworks within their programs that assess creativity based on a number of set parameters. Regardless, this issue of creativity must be solved before machine generated songs can break into mainstream music alone. Another problem facing machine generated music is the issue of ownership. If an algorithm generates a profitable song, do the programmers or the musicians of the input music retain ownership? Or do they both have ownership? If an artist uses machine learning to generate only part of a song, what are ownership splits then? In order for this field to continue to expand, rules must be defined for these types of situations.

5 Remaining Frontiers

While the majority of our analyses has focused on the end-to-end creation of a singular music piece given a set of inputs, the front of research in algorithmic music composition is much more expansive. As the ultimate goal of algorithmic music composition is for human use, one evolutionary approach focuses on tailoring production for a particular user. McArthur and Martin [4] created an interactive application that iteratively trains autoencoder neural networks based on user feedback.

By using an autoencoder, McArthur and Martin generated selections of meaningful fragments of music, denoted as latent space, and applied principal component analysis to this output. The result was a normal distribution of snippets, arranged by their components. Populations could be sampled from this distribution (thus favoring certain common components), fed into the decoder, generating new measures of music. Each generation of music would be rated by a user, before being combined and "bred" to produce a new generation. This approach draws on other models like Picbreeder, which allows users to generate novel evolved images by selecting other simpler images.

Even with such adaptive processes targeted at a human audience, this realm of generation still relies on a large level of subjectivity and user burden. The realm of Affectively Driven Algorithmic Composition (AAC) seeks to apply physiological correlates as a feedback method into the generation process, allowing different motifs, rhythms, dynamics, etc. to be associated with different emotional responses [5]. These models have been built to rely on either user feedback or neurophysical signals measured in electroencephalogram instruments. Current experiments still demonstrate that AAC is in its infancy, but a final AAC system could "generate affectively charged musical structures automatically and reactively in response to a user's emotional state" [5].

6 Non Fungible Tokens

Even as algorithmic audio synthesis becomes more advanced, there are two requirements which must be fulfilled in order for it to become commonplace within society. The first is a comprehensive theoretical understanding of what makes a song quantifiably good, which is necessary in order to devise an algorithm capable of producing listenable sounds. The second is profitability - in order to truly gain traction in our culture, there must be a means to generate revenue to ensure content creation is sustainable. Part of the beauty of music is in it's freedom of form and resulting variety in sound. Though perhaps none of Beethoven's concertos bear much sonic resemblance to the work of DaBaby, you will still find that both pieces of music have a tempo, volume and energy. Spotify has broken down what it determines to be the most character defining elements of a song into the following traits: Danceability, Acousticness, Energy, Instrumentalness, Liveness, Loudness, Speechiness, Tempo and Valence. [6] These are formulated based on a multitude of variables and a corresponding confidence interval. The open source code is yet to be perfect, for example "Liveness" detects the presence of a background audience, with greater values leading to a higher confidence value. This doesn't account for live performances on radio shows, songs with chants in them, or more recently virtual concerts. However, it continues to be developed by one of the largest music streaming platforms, and the technology is offered and encouraged to be used by the public. Spotify have broken down art and quantified its beauty, it seems inevitable that they are on track to soon produce it. The radio industry has lit-

tle room or desire for this prototypical technology; a new space is required for this new genre - and the most apparent avenue through which computer synthesized audio may be implemented for profit is in the NFT market. NFTs are Non-Fungible Tokens, and put simply are pieces of digital art being created and sold globally, with an astonishing market growth over the past couple years. Each NFT is unique, or "non-fungible" and this is where much of the appeal stems from. This desire to own something individual helps to explain the current trend in the NFT space of what are known as "generatives". [7] These are often a series of characters or faces with an overarching theme (for example, monkeys, where for each facial or physical feature, there is one of a number of potential styles/accessories randomly selected. The result is thousands of computer generated, but human curated, unique variations of said character, which are then sold, often for significant portions of money. NFTs can take an extreme variety of forms - the first tweet ever was sold for over 2.9 million dollars - yet surprisingly, fewer than 150 audio NFT sales have ever taken place. In my own time spent browsing the marketplace, I could see why - a few people had attempted to replicate the "generative" style of production, but the sounds were far from as catchy as a cartoon baboon, nor could they be used as a status symbol such as by setting your NFT as your profile picture on Twitter. NFT collections explode in popularity everyday, some due to the artists reputation/community, but more often due to the implementation of a new technology. It seems likely that someday soon one of these new implementations will lay the foundation for an appealing format to present generative audio, and that the resulting surge in interest and profitability in the field, building on Spotify's years of audio analysis development, will bring a genesis of mainstream computer generated audio to the forefront of the NFT market.

7 Conclusion

The realm of algorithmic music composition continues to be dominated by Neural Network approaches, which have so far demonstrated their wide potential. Nevertheless, continued advancements and analysis of existing music provide insight and benchmarks into the characteristics that generated music should demonstrate. The evolving field comprises a rapidly expanding research space with room for academics, composers, and opportunists alike.

Individual Contributions

Andy He: Obstacles and Challenges
Tyler Jang: Controlled Random Walks Algorithm and Remaining Frontiers
Gaurav Parikh: Introduction and LSTM Algorithm
Eric Tishler: Non Fungible Tokens

References

[1] X. F. Liu, C. K. Tse, and M. Small, "Complex network structure of musical compositions: Algorithmic generation of appealing music," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 126–132, 2010.

[2] K. S. Phatnani and H. A. Patil, "Symmetry in the structure of musical nodes," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 353–358, 2020.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[4] R. N. McArthur and C. P. Martin, "An application for evolutionary music composition using autoencoders," in *Artificial Intelligence in Music, Sound, Art and Design* (J. Romero, T. Martins, and N. Rodríguez-Fernández, eds.), (Cham), pp. 443–458, Springer International Publishing, 2021.

[5] D. Williams, A. Kirke, E. Miranda, I. Daly, J. Hallowell, J. Weaver, A. Malik, E. Roesch, F. Hwang, and S. Nasuto, "Investigating perceived emotional correlates of rhythmic density in algorithmic music composition," *ACM Trans. Appl. Percept.*, vol. 12, June 2015.

[6] "Web API Reference | Spotify for Developers."

[7] "BAYC, boredapeyachtclub.com."