University of California Santa Cruz

CSE 185 Final Paper:

Music and Artificial Intelligence

Dylan Sutro, 6/3/2022

Abstract:

Since the inception of digital computing, artists have found a plethora of ways to expand their creativity or find new forms of expression leveraging the technology at their disposal. This has led to a relationship in which technological advancements can allow new musical territory to be explored and likewise musical goals can inform the design of new technology. In the context of music, this has long been in the form of modifying soundwaves or analog signals produced by a human musician to alter their timbre (sonic character). One such example of this is the advent of additive synthesis and then later Frequency Modulator (FM) synthesizers. Only recently in the history of music has this involvement been expanded to other aspects of music such as music composition and already a wide variety of approaches, techniques, and goals have been explored. I aim to investigate these methods and research their conclusions in this paper. (Abstract)

Keywords:

Music, Artificial Intelligence, Music Composition, Signal Processing (key words)

Introduction:

Music has been connecting people through a universal language for tens of thousands of years and continues to bring people together to this day. With the rise of globalization and forms of long distance communication, it has somewhat recently (in the history of music) been possible for people all around the world to share compositional, improvisational, and educational knowledge. In addition to this, technological advancements have allowed for new sounds and styles to be explored. Among these technologies is artificial intelligence, the capability for a

computer to employ “human” thinking procedures such as perception, decision making, and translation. Engineers and musicians alike are rapidly finding new ways of applying this technology to the field of music and numerous goals have emerged from this process. Music can be very difficult to interact with using logical systems because such models often rely on intricate rule sets which restrict the output of the model and music is often referred to as having no rules. While there are certainly approaches musicians employ to play harmonically or “pleasing to the ear,” it is often the risks that a musician takes to deviate from the expectation that makes their playing interesting or creative. This poses a great challenge to computer models attempting to understand music. In the course of this paper, I will attempt to investigate a wide variety of algorithms and systems designed to perform some musical task.

Goals of AI in Music:

Before delving into the algorithms and procedures used to produce some musical model, it is important to understand the different applications and goals that artificial intelligence in music may have. One application that most people are familiar with is classification, or the categorization of music based on a variety of factors that can be extracted from a piece of music. One example of this would be the process by which a streaming service automatically determines the genre of a song. Likewise, comparison algorithms attempt to define a value that represents the relative similarity between two songs or artists which is often used by streaming services for algorithmic recommendation to the user based on their listening history. Another application of artificial intelligence in music is compositional models which attempt to create a new piece of music. This can be used in standalone to generate entire compositions or in a collaborative environment in which a human musician is working with an artificially intelligent tool to inspire

new avenues of creativity. Artificial intelligence is already used heavily in the field of audio mixing and mastering, the process by which raw instrument recordings are combined into a comprehensive mix that best represents the goal of the artist to produce a final track. The final application that we will explore is signal processing, specifically in the context of synthesis, or creation of a new sound. Overall, there are many applications that artificial intelligence has in the field of music and provides both musicians and engineers insight into how humans interact with and produce music.

Preliminary Concepts:

It will be necessary to have a very basic understanding of music theory and be familiar with the concept of statistical determinism. Deterministic behavior with respect to the musical models we will be discussing refers to receiving the same output from our model given the same inputs. This is opposed to non-deterministic behavior which correspondingly means observing different outputs given the same input. The advantages and disadvantages of determinism in musical models will be discussed later in the paper. Musically, it is important to understand the fundamental building blocks of music theory, the standardized language of communication used to convey musical ideas. For this paper we will be working only with the western approach to music based around the concept of twelve tone equal temperament. This refers to the subdivision of the audible frequencies into discrete pitches (frequency of a note) of equal size called half-steps. A musical note defines the pitch and duration of a sound, typically produced by an instrument. An interval is the distance (in half-steps) between two notes. Finally, a musical composition is the collection of musical elements that define the form of a song. This includes melody, harmony, rhythm, and lyrics if applicable.

Representing Music on a Computer:

Audio waveforms in nature represent the change in air pressure with respect to time. This pattern of vibrations is then received by our eardrums and converted into sound by our brain. Representing music in a digital environment can be difficult due to the sonic complexity of a typical composition which can include many simultaneous voices (e.g multiple instruments) over a high range of frequencies and the requirement for digital systems to process discrete signals as opposed to a continuous waveform. Figure 1 shows a snippet of the song “Roller Skate” by Vaughan Mason & Crew and demonstrates some of the challenges with machine listening, or detection of musical features given audio waveforms.

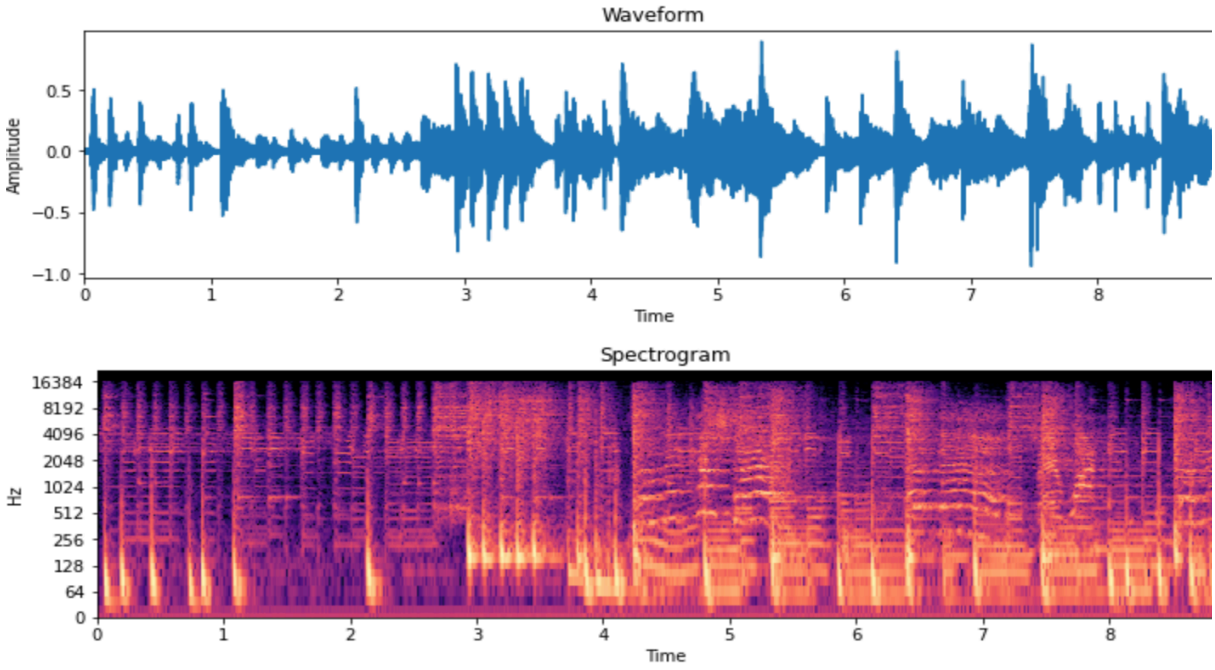


Figure 1: Waveform and Spectrogram of “Roller Skate” Excerpt

The top graphic plots the audio waveform with time as the x-axis and signal amplitude on the y-axis. This represents the electrical signal sent to a speaker or amplifier which in turn uses a speaker to emit vibrations causing pressure variations in the air (or whatever medium the signal

travels through) of the pattern of amplitude changes in the original waveform. Similarly, a waveform can be generated by a microphone which captures the changes in amplitude and outputs them as an electrical signal. This format can be very difficult for computers to interpret because all of the compositional elements and voices are combined into a single continuous amplitude value. The bottom plot is a spectrogram which shows the intensity of frequencies over time represented by the brightness of the color at a given frequency. This spectral nature of music can make it difficult for a machine to delineate different musical voices which may "bleed" into overlapping frequency spaces. In order to process and compute using musical data, we need to supply the model with digital inputs which typically take the form of numerical values.

Musical Instrument Digital Interference (MIDI):

MIDI is a technical standardardized communication protocol that acts as a digital music interface. This allows for the abstraction of the musical compositional information from the sonic information (how the music is physically realized by an instrument). Compositional information includes things such as the tempo, melody, harmony, and dynamics (e.g indications for notes to be played at different volumes). On the most basic level MIDI represents a monophonic (one note at a time) voice as a list of pitches and a corresponding list of duration and velocity (loudness) values for each note. This allows us to perform mathematical operations on the composition or "DNA" of a song rather than a specific performance.

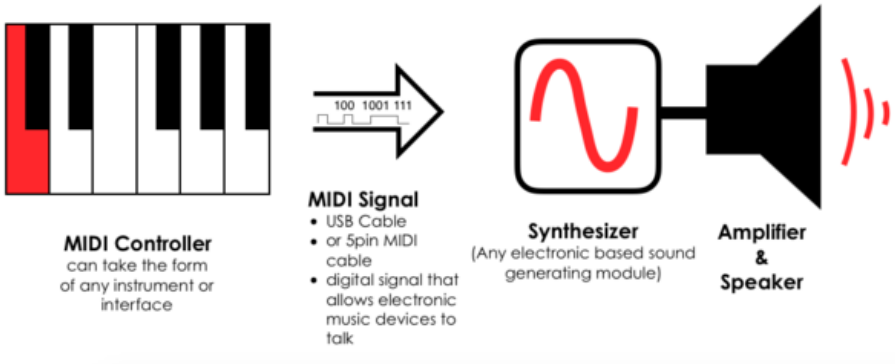


Figure 2: Simple MIDI Signal Chain [1]

Figure 2 shows a simple MIDI setup in which a device called the controller (seen here as a piano keyboard) outputs a digital signal corresponding to the performance played on the controller. This signal is then received by a synthesizer which produces an audio signal to be physically realized by an amplifier and speaker rig. Digitally, we can store songs in MIDI form by mapping each instrument in an arrangement to a MIDI channel and manually transcribing each part or by importing sheet music and converting it to MIDI. Alternatively, the process of machine listening can be employed to extract MIDI information from a raw waveform.

Musical Features:

In order for our model to make sense of music, it must be able to detect and compose with musical features – musical characteristics that can be stored numerically. One simple example of this is tempo, which is represented by a single value, the BPM or beats per minute. This defines the pulse of the song and the duration of notes with respect to the tempo that the composition follows. More challenging musical features include mood, genre, rhythmic patterns, and “creativity” which have less obvious ways of being interpreted by a machine. The subsequent chapter on parameterization goes into more detail on how musical features are

represented and the later chapter on algorithmic analysis explores how a model might extract these features from a given composition.

Generative Music:

Generative music is an early approach to AI music composition which was developed by Tima and Peter Cole and used by Brian Eno for his 1996 album “Generative Music 1” [2]. The basic premise is to use chaotic systems to produce dynamic semi-random musical compositions by mapping some signal source into musical features. Eno asserts that generative music is a composition approach that predates digital computing and can, at a very rudimentary level, consider the wind chime as a form of generative music. Eno used the KOAN Generative Music Engine which takes a set of constraints from the composer and generates voices in accordance to the constraints as well as implementing semi-random determination to produce a unique piece. Eno believed that this approach combined the best elements of studio recording with the authenticity and distinctive characteristics of live performance. This approach was later developed by Joseph Nechvatal in his work “Viral SymphOny,” which investigated the application of genetic models towards generative music [2]. Discussion on genetic algorithms and evolutionary music models appears later in this review as well as examples of algorithms used to produce generative music.

Parameterization:

Parameterization is the fundamental component of artificially intelligent music models that allow them to perceive, learn, and generate music in a format that is computable for machines and remains familiar to humans. Parameterization is the process of finding numerical values for musical features by defining quantitative ranges for parameters which the models use to analyze compositions and generate music. Typically we have two types of parameters, physical and probabilistic. A physical parameter has a discrete value and typically defines composition level parameters such as tempo or pitch center as well describing ranges. A probabilistic parameter on the other hand can be stochastic (indicates the probability of an event occurring) or a maximum allowed deviance from some norm (typically the rhythmic pulse defined by the tempo). In order to better understand the process and purpose of parameterization we will use Larry Polannski's 1997 parametric music composition model *Anna's Music Box* (AMB) as an example of implementation of parameterization. [3] AMB is a parametric musical model which quantizes musical features such as rhythm and melody into a set of integer and float values which can be used to generate "critters", musical voices whose playing can be modified by various musical parameters.

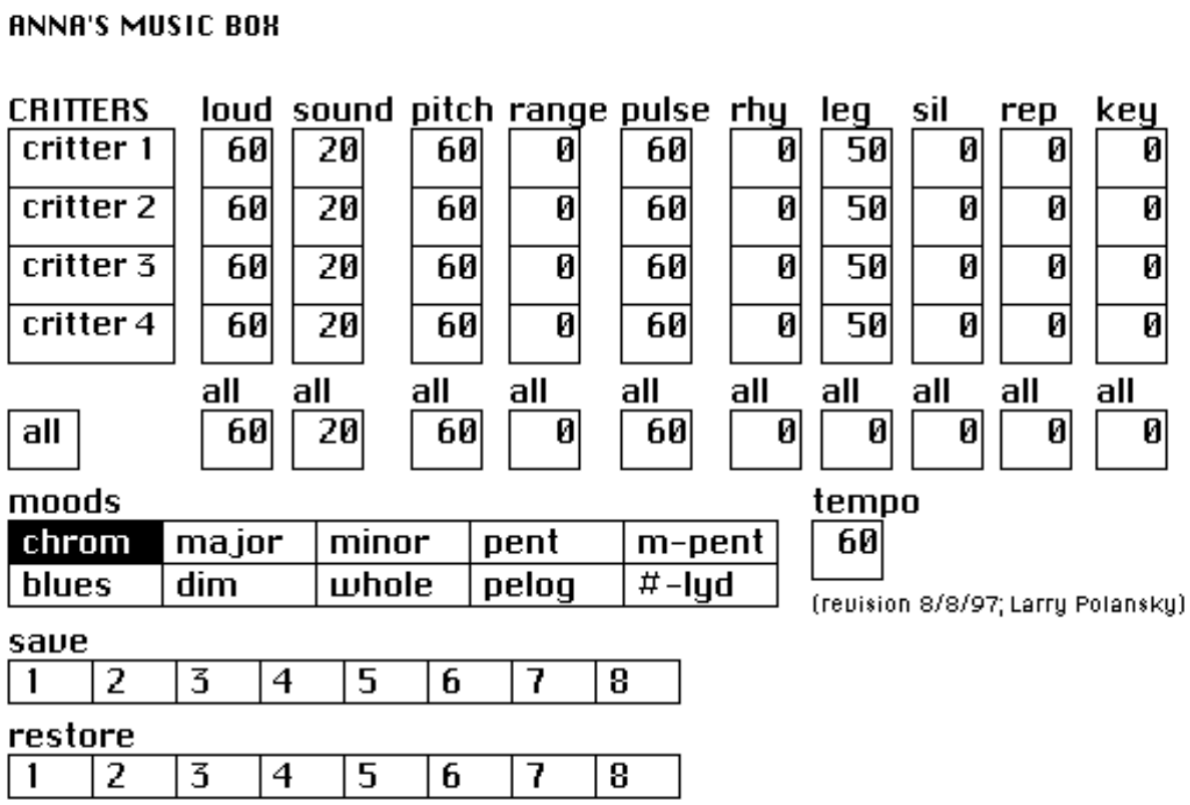


Figure 3: Graphical User Interface (GUI) of Anna's Music Box [3]

Anna's Music Box was designed by Polanski as a toy for his five year old daughter and her kindergarten class. The figure above shows the GUI for the software and could be used by a device as a MIDI controller which could then use a MIDI synthesizer and a speaker to play the generated compositions. The various parameters used by AMB are shown as the values to the right of the critters and each represent some impact on the final composition, beit stochastic or discrete. A brief description of each parameter is given below followed by a simple pseudocode implementation of the basic algorithm employed by AMB.

- *loud:* Volume of the specified critter.
- *sound:* Sets the instrument preset to be used for the given critter's MIDI channel when synthesized.
- *pitch:* Pitch center of generated melody (average pitch).

- *range:* Maximum interval allowed above and below *pitch.*
- *pulse:* Defines the duration of each beat (parameterization of tempo).
- *rhy:* Probabilistic parameter that determines the maximum allowed deviance of a note's duration from the pulse.
- *leg:* Sets the MIDI legato parameter which controls how long notes take to release.
- *sil:* Controls the probability that a note is a musical rest (is silent for the duration of the note rather than producing a pitch).
- *rep:* Represents the probability that a note is repeated.
- *key:* Tonal center used by the mood parameter
- *mood:* Enumerated list of preset scales to constrain note selection to the given harmony rooted around *key.*

These parameters are used by a composition algorithm to generate MIDI sequences which are then sent to a MIDI synthesizer. The pseudo code below is a simplified implementation of a parametric model's composition algorithm.

```
2
3   def compositionAlgorithm(
4       pitch_center,
5       pitch_range,
6       pulse,
7       ryhthm,
8       repeat,
9       silence,
10      memory,
11      length):
12
13      pitch_list = []    # stores selected note pitches
14      dur_list = []      # stores selected note durations
15
16      while length(pitch_list) < length:
17          # check to see if we repeat the previous note
18          if random(0, 1) <= repeat and len(pitch_list) > 0:
19              new_pitch = pitch_list[-1]
20              new_dur = dur_list[-1]
21          else:
22              # check to see if we rest (pitch of -1 indicates silence)
23              if random(0, 1) <= silence:
24                  new_pitch = -1
25              else:
26                  # select new note between pitch range around center
27                  lo = pitch_center - pitch_range
28                  hi = pitch_center + pitch_range
29                  new_pitch = random(lo, hi)
30
31              # duration is computed using pulse (in BPM) and rythmic vatiation paramater
32              rythm_variation = random(1-ryhthm, 1+ryhthm)
33              new_dur = (60/pulse) * rythm_variation
34
35          # add new pitch and duration to list
36          pitch_list += [new_pitch]
37          dur_list += [new_dur]
38      # send MIDI data to synthesizer
39      return pitch_list, dur_list
```

Figure 4: Pseudocode Demonstrating an Example Composition Algorithm

This function loops until a composition is of the specified length and uses a simple inclusive uniform random number generator and the parameters provided to select pitches and durations to be sent to a MIDI synthesizer. The first control statement generates a random number between zero and one and compares it to the repeat parameter, it then sets the new pitch and duration equal to the previous note if it satisfies the condition. If the note is not repeated or is the first note in the composition, the algorithm must select a new pitch and duration. Following the same procedure as the repeat control statement, we check to see if a random value satisfies

the silence parameter and set the new pitch to negative one (indicating no sound to be made). If the note is not a rest, we select a new pitch from the set of notes in the range above and below the pitch center determined by pitch range. Finally, we generate a duration for our new note or rest. To do this, we define a rhythmic variation multiplier which will displace the length of our note around the pulse by the percent variation parameter rhythm. We continue this process for each critter until our compositions are generated and the MIDI data can be sent to a synthesizer. Some interesting uses of a simple model such as this are generating harmonies or polyrhythms using ratios of parameters of the various critters. To produce harmonies, we can set the same pulses for the critters and they will transition together resulting in simple harmonies. Likewise, using ratios between the pulses of our critters can generate polyrhythms where voices play at various tempos and alternate between sharing beats and being rhythmically out of phase. Overall, rule based parametric models are limited because they do not have the ability to learn and thus lack the ability to be intelligent systems. Additionally, it can be difficult to balance designing a model that behave's "creatively" without introducing too much randomness which can make a critter too chaotic and arrhythmic to be pleasant. Parametric models are a good introduction to application of music features which will be used throughout the rest of the paper. To begin building more intelligent models, we need to build a system that can learn some understanding of music and the common structures and patterns that appear frequently in the training data supplied.

Algorithmic Analysis

Algorithmic analysis describes the process of using a digital system to perform some feature extraction and memory storage from an input of musical information. In this context, musical information could be parameterized (in MIDI form) or continuous (audio waveform) in which case we would have to perform machine listening, or parameterization of an audio file. In either case, once the musical data has been converted to a digital format for our model to interpret, our goal is to extract musical features and build some symbolic description of the analyzed signal. As the feature extraction is being performed, a temporal memory layer can detect patterns in symbolic descriptors for each analyzed signal. This short term memory can be passed to a deep learning layer which uses clustering techniques to build a long term memory. Algorithmic analysis is employed in all of the layers and we can generally categorize them either as listening algorithms which perform feature extraction, or learning algorithms which are trained by the listeners output.

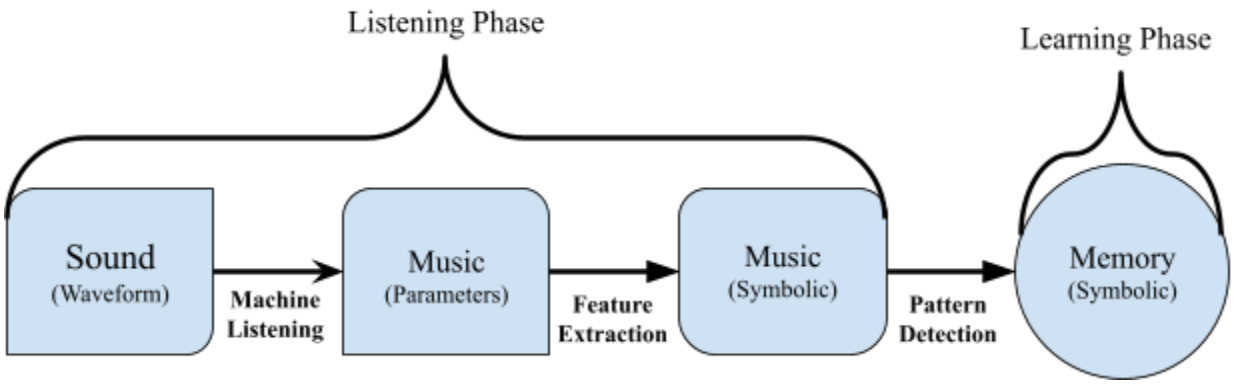


Figure 5: Algorithmic analysis model to perform listening and learning over music data

Figure 5 illustrates a simple layered algorithmic analysis approach that includes four major components:

1. Machine Listening: Convert an audio signal in raw form into a MIDI representation.
2. Feature Extraction: Build some symbolic description of the music piece being analyzed.
3. Pattern Detection: Create some temporal memory for the audio signal actively being processed.
4. Learning and Clustering: Apply deep learning techniques such as optimization, classification, regression, or machine learning to name a few.

Throughout this section on algorithmic analysis we will classify several algorithms under these categories and subsequently investigate how these models can then be applied to generating new compositions.

Machine Listening:

The goal of machine listening is to perform a function similar to the process of human transcription. Transcribing refers to the practice of listening to a piece of music and determining its notation (how to represent the song on sheet music). This involves some simple procedures such as determining the tempo of a song, as well as more difficult detection such as delineating the melody from the harmony or separating the rhythm of the drums from a potentially busy frequency domain. As introduced earlier, audio waveforms are most understandably represented by spectrograms which show the intensity of frequencies in the audible range with respect to time. To begin processing the signal, we need to perform sampling, or converting the continuous time signal into a discrete array of values corresponding to the intensity of defined frequencies at some time interval referred to as the sampling rate. We then take the matrix of sampled frequency data and perform pitch detection to estimate the average pitch of the signal and

attempt to separate the various sources (separating the bass line from the melody line for example). We can employ many spectral analysis techniques at this point to statistically describe the shape of the signal to better inform our feature extraction, some of these spectral features include [4]:

- Spectral Centroid: Measure of the "center of mass" or weighted average frequency of the spectra.
- Spectral Skewness: Indicates how symmetric the signal is around the average frequency.
- Spectral Kurtosis: Measures the average noise in the signal.
- Spectral Rolloff: Frequency below which a specified percentage of the signal's energy is contained.
- Spectral Bandwidth: Difference between highest and lowest frequencies in a signal.

These various spectral features can be used both to aid the process of transcription as well as being stored in some memory system which allows a generative model to optimize its output signal given the spectral features of the training data.

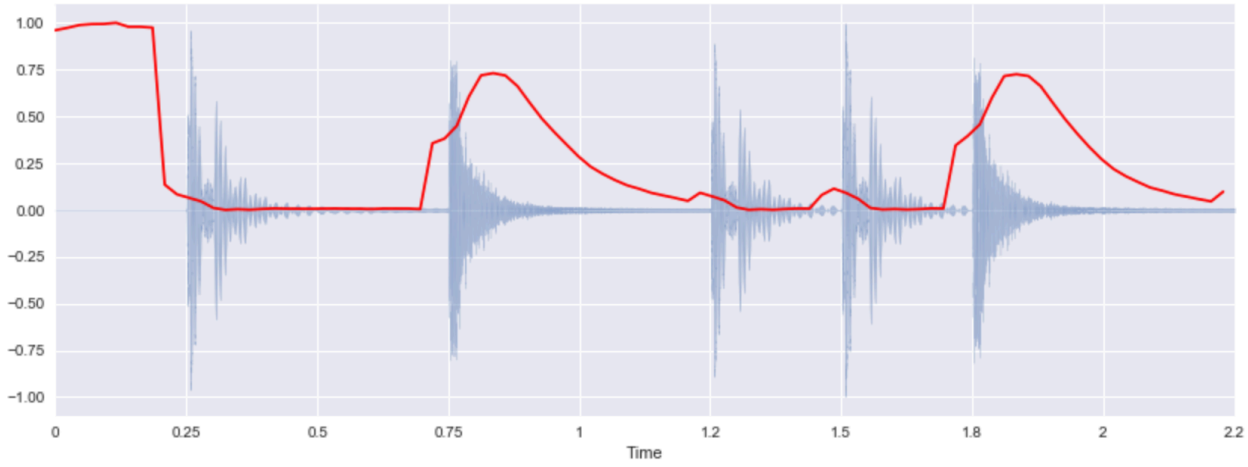


Figure 6: Spectral centroid plotted over original waveform [4]

The plot above shows an audio waveform with the calculated spectral centroid overlaid in red. As we can see, the spectral centroid generally follows the peaks of the waveform and can be a useful tool in performing peak detection and pitch estimation. Some features such as the spectral centroid also roughly define perceptual elements of a sound such as its brightness, roundness, or sharpness. One interesting application of machine listening is Austrian composer Peter Ablinger's speaking piano, which was capable of sampling a waveform of a human voice and resynthesizing the speech using a physical piano played by a mechanical system controlled by a machine listening program [5]. The speaking piano works by decomposing the sampled input signal into single notes or ratios between two notes to reproduce the peaks of the signals and roughly recreate the speech used as input.

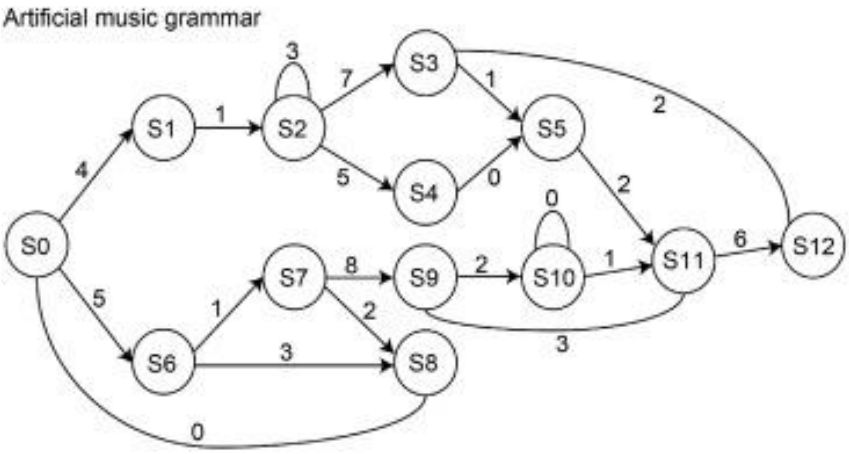
Feature Extraction:

In this stage of the model, the aim is to receive musical information from the machine listening algorithm and construct a set of parameters which describe the musical identity of the piece being analyzed. Many different algorithms may be applied in this phase to extract different features and we typically aim to define as many parameters as possible to give the model an adequate amount of data to process. Many mathematical equations are often used to statistically describe the composition and often use the spectral features determined in the machine listening stage. One example of feature extraction is beat detection which samples a fragment of the audio signal and searches for evenly spaced impulses denoting a constant pulse. Other parameters such as the ones seen previously in Anna's Music Box can be defined such as pitch center, pitch range, and rhythmic variation. Some features can be hard to define because of the difficulty in quantizing perceptual elements of music such as mood or sentiment. Individually, the parameters

defined for a given composition do not provide much intelligence unless they are clustered and stored in memory as will be seen in a later section.

Pattern Detection:

Still analyzing pieces individually, the pattern detection stage employs conditional probability methods such as Markov chains to build a short-term memory modeling the frequently used melodic and rhythmic “phrases” (multiple notes constituting a musical line) occurring in the composition. Using Markov chains as an example, state transition matrices can be utilized to build a musical grammar, or language of commonly occurring phrases. In the context of musical grammars, a state transition matrix represents a sequence of pitches, durations, or both. We can use different order Markov chains to represent phrases of various lengths. For example, a fifth order Markov chain would build a state transition matrix by representing a state as a sequence of five notes and transition as the subsequent note in the voice being analyzed. We can then use this data to build a set of probabilities that defines the likelihood of a note occurring in the original melody given the five previous notes. This will be used later in resynthesis to transform our pattern detection into pattern prediction.



Musical intervals

0 = F D	3 = C G	6 = A E
1 = E F	4 = F A	7 = B E
2 = A G	5 = D G	8 = C B

Example sequence with IDyOM probabilities

Interval	4		1		7		1		2		6	
Note	F	A	E	F	B	E	E	F	A	G	A	E
Probability	2.53	1.88	0.86	0.16	3.28	0.89	3.30	0.30	0.77	0.77	1.06	0.20

Figure 7: Example state transition diagram for music grammar [7]

The graphic above demonstrates an example state transition diagram for a second order Markov model which creates a probability table for the occurence of various intervals given the previous interval. We will see later in the chapter on resynthesis how this can be applied to the composition side of the model.

Learning:

The learning phase of the model is what allows it to behave “intelligently” and extract information from a set of musical input rather than a singular piece to analyze. The goal is to use deep learning methods on the outputs from the previous stages (machine listening, feature extraction, and pattern detection) to construct a musical surface, or an N-dimensional space

representing the various parameters used by our model. We can use a variety of functions to analyze this musical surface such as regression or optimization. Typically, machine learning models are employed at this stage in the form of Neural Networks which attempt to construct a "fingerprint" for the training data to later generate new musical surfaces in a similar vein and synthesize them into an original composition. The learning phase can also be used for classification in which the model tries to group the various training files into categories and then classify future inputs into these categories. Another cool application of the learning phase is query by description which attempts to match a piece of music in the training set with a requested audio file (i.e humming a melody and receiving the original song back). Overall, the goal of the learning phase is to cluster the parameters previously defined and store them dynamically so the "musical space" is defined by the inputs to the model.

Compositional Resynthesis:

Compositional resynthesis is the process of using an analysis model to generate a new unique composition. This can be done by using the layers described in the previous sections to perform inverse functions to produce new compositions. Additionally, we can employ generative and analysis models to create non-deterministic compositions using the cognition gained from analysis and the "creativity" gained from generative models. One example of this would be to perform constrained optimization on the output of a translational algorithm. In this context, a translation algorithm takes any input (typically somewhat random) and attempts to represent it musically. An example of this would be to process the pixel values of an image as a spectrogram of an audio file and generate a composition from this data. To employ our analysis model, we would generate a musical surface for our new composition and use optimization functions to

minimize the difference between the new composition and learned musical surface. This allows a chaotic input to be “trained” to a preexisting model. Alternatively, we can use the analysis model in conjunction with a parametric model to make more informed decisions during note selection. Overall, the goal is to receive some musical input and generate a new musical output that shares perceptual identifiers.

Conclusion:

Overall, the field of artificial intelligence presents many new and exciting opportunities for musicians. Over the course of researching and writing this paper, I gained a newfound appreciation for the complexity of intelligent digital systems as well as perspective into the nuanced cerebral and emotional decisions that result in a good composition. While it is interesting to explore the role of artificial intelligence in the field of music composition, it is ultimately the human element of music that allows us to connect with sound on an emotional level. It will be interesting to see how the fields of music and artificial intelligence affect one another and inform their respective developments.

References & Works Cited

[1]: “History of MIDI - What Is MIDI?” *Hosa*, 8 Feb. 2021, https://hosatech.com/press-release/history-of-midi/#:~:text=When%20Was%20MIDI%20Invented,over%20a%205%2Dpin%20cable.

[2]: Fernandez, JD, and F Vico. “AI Methods in Algorithmic Composition: A Comprehensive Survey.” Journal of Artificial Intelligence Research 48 (2013): 513–582. Web.

[3]: Polanski, Larry. *Anna's Music Box Software*, http://eamusic.dartmouth.edu/~larry/AMB/amb.front.html.

[4]: Music Information Retrieval, *Spectral Features*, https://musicinformationretrieval.com/spectral_features.html.

[5]: Ablinger, Peter. “Speaking Piano.” *Peter, Ablinger, Speaking Piano*, https://ablinger.mur.at/speaking_piano.html.

[6]: “Beat Detection.” *Beat This > Beat Detection Algorithm*, https://www.clear.rice.edu/elec301/Projects01/beat_sync/beatalgo.html.

[7]: Zioga, Ioanna, et al. “From Learning to Creativity.” *NeuroImage*, Academic Press, 25 Oct. 2019, https://www.sciencedirect.com/science/article/pii/S1053811919309024#fig1.

[8]: Holtzman, S. R. “Using Generative Grammars for Music Composition.” *Computer Music Journal*, vol. 5, no. 1, 1981, pp. 51–64. *JSTOR*, https://doi.org/10.2307/3679694.

[9]: R. Sabitha, S. Majji, M. Kathiravan, S. G. Kumar, K. G. Kharade and S. R. Karanam, "Artificial Intelligence Based Music Composition System-Multi Algorithmic Music Arranger (MAGMA)," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021, pp. 1808-1813, doi: 10.1109/ICESC51422.2021.9532706.

[10]: Assayag, Gérard, et al. *Guessing the Composer’s Mind: Applying Universal Prediction to Music*. http://articles.ircam.fr/textes/Assayag99a/index.pdf.