5-2017

Stochastic Improvisation of Jazz Solos

Tevin Rouse
Montclair State University

Abstract

We propose using stochastic methods to generate new Jazz solos in the style of an artist of interest. To accomplish this, we implement several Markov models that use an artist's known solos in order to mimic their pitch selection tendencies. Construction of two unique solos were generated for each artist considered as well as analysis of the characteristics the solos possessed in comparison to the artist's original solo. This software implementation seeks to offer a new method for creating computer music compositions.

MONTCLAIR STATE UNIVERSITY

Stochastic Improvisation of Jazz Solos

by

Tevin Rouse

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Mathematics

May 2017

College of Science and Mathematics

Department of Mathematics

Thesis Committee:

[redacted]

Dr. Bogdan Nita
Thesis Sponsor

[redacted]

Dr. Ashuwin Vaidya
Committee Member

[redacted]

Mr. Ethan Hein
Committee Member

Stochastic Improvisation of Jazz Solos

A THESIS

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

by

Tevin Rouse

Montclair State University

Montclair, NJ

2017

Acknowledgments

I would like to thank my Thesis adviser, Dr. Bogdan Nita for his guidance and insight in this research project. I would also like to thank my committee members Dr. Ashuwin Vaidya and Ethan Hein for there assistance and recommendations in this project. Thanks to my family and friends for continuous support and encouragement.

Contents

List of Figures

Chapter 1

Introduction

Computer Music has become a common technique used by composers in creating new musical pieces. Many approaches have been used stemming from scientific and mathematical formulations. With the use of these approaches, much effort has been put into creating new classical music pieces that yield convincing representations of classical composers like Bach and Mozart. In [2], Benson uses a group theoretic approach to construct new pieces. In his implementation he uses the action of the Dihedral group on the pitch class set to take a preexisting piece and apply rotations and symmetries to the musical measures. This yields a new piece with the notes of the original shifted by some arbitrary angle θ originating from a "center" note. This approach could also yield an inverted form of the original piece. The main premise of the article was that construction of sequences of symmetries and rotations on the original piece yields new pieces. Several examples were constructed in which the technique was applied to classical pieces.Agmon in [16], discussed a mathematical formulation of the diatonic system using modular arithmetic. He expressed the diatonic system as a pair of integer classes $mod(12, 7)$. This mathematical formulation stems from the fact that the set of scale steps is equivalent to $Z_7 = 0, 1, 2, 3, 4, 5, 6$ while the set of diatonic intervals is equivalent to a subset $A = (0, 2, 4, 5, 7, 9, 11)$of Z_{12}. Diatonic intervals are referring to the possible distances that can occur between pairs of notes such as minor second, major second, perfect fifth, etc. Agmon analyzes the result of applying modular arithmetic to these pairs and concludes what is the corresponding music interpretation of these operations. Agmon demonstrates that taking the difference between pairs yields a different interval depending upon the order in which the operation is applied. Computing the difference (which corresponds to distance between notes) also alluded to the cyclic nature of the construction.

Some research that has been done in blending the ideas of math and music is the study of gestures that an artist makes while they are playing the piece. Such topics in mathematical music theory are categorized under Performance theory. In [17, 12], Muller discusses the creation and implementation of a computer software which creates the movements (referred to as gestures) of the hands during a piano performance. Muller defined a gesture to be a group consisting of (Λ, X, g) where Λ is a directed graph representing the structure of the gesture being performed, X is a topological space representing the space in which the gesture occurs, and g is a map from Λ to the directed graph generated from the space when considering all possible paths in space that connect two vertices. The complexity of this representation can be seen by considering the following example. Consider the gesture involved in playing a single note on the piano. In order to model this gesture, three curves have to be considered: the curve when moving the finger to that key, the curve when pressing the key, and the curve when moving away from the key. Because of the number of curves needed to represent one action, Muller defines the concept of hypergestures which is a collection of gestures. Using this approach, Muller describes the construction of this problem to that of equivalent questions from Homotopy theory. Some work has been done for determining what collections of musical pieces have particular traits as in the work of Burgoyne et al [4]. In their work a statistical technique was shown to effective in determining the harmonic structure of large collections of musical pieces. Techniques from compositional data analysis were used in order to determine whether a collection of Pop music pieces had particular harmonic similarities. Pieces were chosen from the 1950s to 1990s in order to determine how much change has there been in chord choice. Compositional data is essentially a vector (typically defined as compositions) in which each component is a numerical value related to some aspect(i.e. portion) of each entry in a data set. This is typically expressed as ratios between the components. With regard to this study, each data entry corresponded to a song in the collection. The components of the compositions, which were referred to as root compositions in this work, correspond to the scale degrees which are the roots of the chords in the song. The numerical value for that component corresponded to the ratio of time spent on that particular degree in the entire song. From this work they were able to determine that changes in chord usage occurred in the 1980s. Statistical analysis techniques like MANOVA (mutlivariate analysis of variance) were employed to determine information such as the effect of hit singles on the music of that time period.

Because of the success found in applying mathematical techniques to generating classical music, researchers have consider their application to Jazz improvisation. Jazz, as opposed to other musical genres, is based entirely on musical freedom. This freedom allows for each artist to have their own methods of self expression and thus establishing a mathematical formulation of Jazz approaches is very difficult and thus requires complex mathematical formulations. In [1], Bergomi and Portaluri propose the use of braid theory to model the modal chord progressions used by Jazz artist in their improvised solos. Taking the planar graph representation of the chords used in the artist's solo and considering their corresponding topological braid classes, they were able to determine the sequence of braid concatenations used to construct the solo. Thus establishing a mode allows for the construction of solos using the sequence of concatenations used by the artist. A recently proposed technique, established by

Mazzola [13], in generating solos is that of formalizing music in terms of category theory in which the characteristics of a note (or a collection of notes) such as pitch, duration, and onset are described as morphisms between spaces. Thus the generation of new solos stems from the manipulation of these mappings. In his work Mathematics of Jazz ,[18], Maurer explains the mathematics of symphonic music and how this mathematical approach does not perfectly represent Jazz music. Several cases are discussed considering first the approach of note duration in Jazz. In symphonic music, the standard note duration notation is taken and is maintained throughout every bar in the piece. For example, if a note occurs in a classical piece with the duration of a eighth note then it should be held for that duration when played. Because of this clear interpretation, analysis of duration from a mathematical standpoint can be done using a binary representation. An eighth note has a duration value of $\frac{1}{8}$, so its corresponding binary representation is 0.001. Thus every note duration has its own unique binary representation. However in Jazz, this is not necessarily the case. In Jazz the duration approach can be either the standard approach or can be swing style approach. When a bar of music specifies that it should be played with swing, then the duration of the notes change. Even more so, in the Jazz community there are two approaches to swing. This therefore leads to at least two binary representations for each duration. Further differences also occur when referring to chord selection. In symphonic music, chords are based on preset harmonics depending upon whether playing in a major or minor key. Because of its 'fixed' nature, patterns can be established based on constant multiples of its frequency. For example, if a piece is in a major key, then when a chord is played, there is a choice between augmented, diminished, major, and minor depending upon the feeling the composer is trying to express. However because of the freedom in Jazz, it is possible to play for example a minor seventh chord with the fifth note either sharped or flatted. There is also the concept of chord substitutions in which it is perfectly acceptable to play for example a dominant seventh chord in replacement for a minor seventh. Maurer even expressed that, because of the flexible approach of Jazz, the final chord of a piece does not have to be the tonic which is typical of symphonic music.

Among the mathematical techniques incorporated in constructing Jazz Improvisation, one of the popular approaches is that of Markov Chains. Ames [20] in 1989 was the first to give an overview of all of the approaches established in composing music through the use of Markov Chains. In his work, Ames discussed the now standard method (which is employed in this work) of using pitches that occurred in a piece of music as the state space and constructing Markov Chains from the work of classical composers. Another method for composing that he discussed was the use of combining several excerpts generated from different Markov chains. This allows for choosing the 'favorable' aspects from the output of the chain in order to create pieces as well as better control of musical variability and expression. The idea of evolving transition matrices was discussed in which the probabilities found in the transition matrix of the Markov chain was allowed to chain based on preset parameters. Marom ,[14], considered the application of Hidden Markov Models(HMM) in which unknown or "hidden" phenomena that may have affected the artist's transcribed solo (i.e. the artist's state of mind at the time of soloing) is accounted for in the construction of their transition matrix. In this work, we present the creation of software that allows for the user to create computer generated improvised jazz solos which mimic the tendencies of a jazz artist of their choosing from a preset collection. We focus on using the pitch data of solos from Jazz artists Miles Davis, John Coltrane, and George Benson to construct Markov Chains that model their improvisational behavior for a particular song. These artist were chosen because solos they played in the piece we chose did not contain block chords which our approach does not effectively handle. We generate solos using 1^{st} and 2^{nd} order Markov chains and provide analysis strictly for the 1^{st} order case using probabilistic model checking. It is worth noting that we attempted to apply this same construction to durations (i.e. note values). However we were unsuccessful in creating interesting rhythmic output since the dominating duration value in the solos used were eighth notes. This led to eighth notes having a high probability of occurring and therefore yielding solos which did not capture the interesting rhythmic approaches that these artists employed in their corresponding solos. We then propose a Markov model construction based on pitch classes and show how to generate a vector representation of an artist using the properties of Markov chains. We applied this construction to create a vector representation of Jazz saxophonist Charlie Parker and applied techniques from Monte Carlo Markov Chains to generate a solo using the vector representation of the artist. Probabilistic model checking is again used to analyze the effectiveness of the model in producing solos which resemble the improvisational nature of the artist.

Chapter 2

1st Order Markov Process on Pitches

We will use the theory, described in [7, 6, 5], of the following terms. Let the following pair (Ω, S) be a measure space with set Ω along with its σ-algebra S of subsets of Ω and define P to be a probability measure on (Ω, S). Thus (Ω, S, P) is a probability space.

Definition. *A stochastic process X with state space χ is a collection ${x_n}_{n=0}^{\infty}$ of χ-valued random variables on (Ω, S, P).*

A random variable x_i is simply the value of the process at the i^{th}*timestep.* To define a Markov process, we will use the following notation: For every $m \geq n$ the σ-algebras $S_n^m = \sigma(x_n) \vee ... \vee \sigma(x_m)$ where $\sigma(x_i)$ is the σ-algebra generated by a random variable x_i.

Definition. *A stochastic process X is Markov, for every $n > 0$ and every measurable bounded function $f : \chi \longrightarrow R$ (where R is the set of real numbers)one has $E(f(x_n)|S_0^{n-1}) = E(f(x_n)|S_{n-1})$ almost surely.*

Definition. *A Markov process is time-homogeneous if $\exists$ a measurable map P from χ into $P(\chi)$, the space of probability measures on χ, such that $\boldsymbol{P}(x_n \in A|x_{n-1} = a) =$*

$(P(a))(A)$ *for every* $A \in B(\chi)$ *the Borel algebra, almost every* $a \in \chi$ *and every* $n > 0$*, where* P *is the transition probabilities for* x*.*

Definition. *A Markov chain has order* n *if the probability of state* x_i *occurring depends upon the previous* n *states. If a Markov chain has order* n*, we may say that it has memory* n*.*

We will be discussing the application of time-homogeneous Markov Chains which are time-homogeneous Markov Processes that have finite state space. Our measurable map P will be represented by a transition matrix. Because we are considering time-homogeneous Markov Chains, the transition matrix will stay the same after each time step. In our application, we will consider the state space to be the collection of all unique notes used in a given piece of music. Thus each entry in the transition matrix will correspond to the probability of going from one note to another. For all examples and analysis in this work will be based on 1^{st} and 2^{nd} order Markov Chains.

2.1 Example of 1st Order Markov Process on Pitches

Lets consider the following example of this construction. Using the nursery rhyme 'Mary had a little Lamb' we will consider the construction of its transition matrix. Below is the sheet music for the piece.

We will start by first collecting all unique notes in the piece. So the first unique note encountered is $E4$ (the first note in the piece of music). Note that $E4$ means that we are referring to the tone E which occurs in the 4^{th} octave. With this note chosen we can now ignore all other occurrences of $E4$ as a unique note. Continuing in this manner we get the following collection of notes: $C4, D4, E4, G4$. Each of these notes will be a state x_i for our Markov chain and the collection of these notes is our state space χ as defined above. Thus we can begin constructing the transition matrix which will look like this:

$$\begin{array}{c} \\ C4 \\ D4 \\ E4 \\ G4 \end{array} \begin{array}{c} \begin{array}{cccc} C4 & D4 & E4 & G4 \end{array} \\ \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{pmatrix} \end{array}$$

Taking the first note $C4$, we will look through the piece and count all occurrences of $C4$. The first occurrence of the note $C4$ is the third note in the sheet music. Its important to note that we will not count the last note which is also a $C4$ since we will only be considering pairs of notes in which $C4$ is the first note of the pair. Because the last note has no note coming after it, it cannot be considered. Therefore the number of $C4$s that appear is 2. Starting at the first entry in the above matrix we will count the number of times that the pair $[C4, C4]$ appears (i.e. when a $C4$ is followed by a $C4$ in the piece). Since this pair does not appear the probability of $C4$ being played after a $C4$ is 0 and therefore the $(C4, C4)$-entry is 0. Next is the $(C4, D4)$-entry. Using the same process we see that the $[C4, D4]$ pair appears twice which is the only two occurrences of $C4$ (that we count in this construction) in the piece so the probability of $D4$ following a $C4$ is 1. Continuing this process for each entry in the matrix we get the following complete transition matrix:

$$\begin{array}{c} \\ C4 \\ D4 \\ E4 \\ G4 \end{array} \begin{array}{c} \begin{array}{cccc} C4 & D4 & E4 & G4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ {}^{3}/_{10} & {}^{3}/_{10} & {}^{2}/_{5} & 0 \\ 0 & {}^{4}/_{9} & {}^{4}/_{9} & {}^{1}/_{9} \\ 0 & 0 & {}^{1}/_{2} & {}^{1}/_{2} \end{pmatrix} \end{array}$$

With the transition matrix constructed from this piece, we can now begin to build a new song using the rand function [10] from MATLAB to give us a uniformly distributed random number that is in the interval $(0, 1)$.

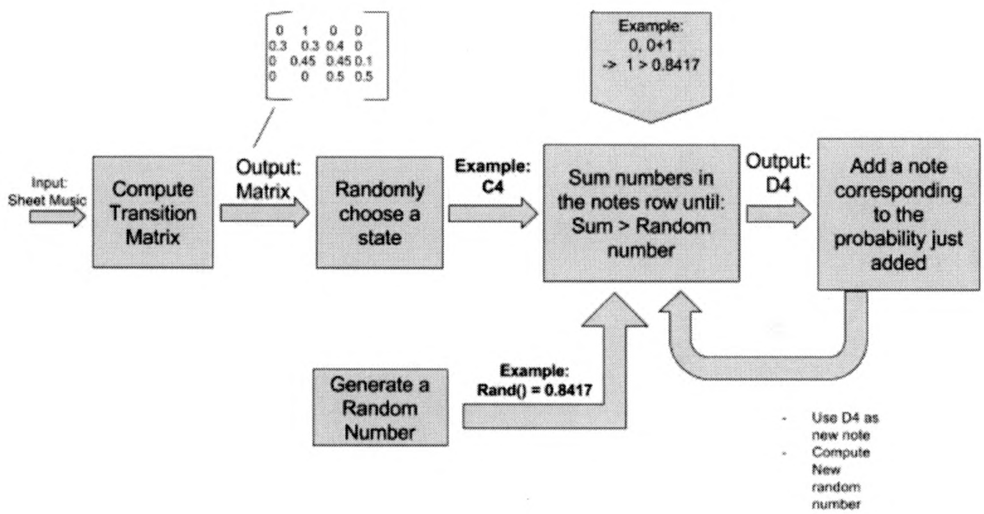


Figure 2.1: Illustration of the Markov Process

Letting the first note in our new song be $C4$, we call the rand function which gives us a value of 0.8417. We then begin by adding up the values in the $C4$-row until we get a number larger than 0.8417. Once this is achieved the note whose value we last added will be chosen as the next note in the new song. Since in this example the only note that can come after $C4$ is $D4$, our next note in our song is $D4$. We then consider the next note by looking at the possibilities that can come after $D4$. The rand function now gives us 0.9058. Adding the values in the $D4$-row, the moment we get a number larger than 0.9058 is when we add the number in the $(C4, E4)$ entry. Thus the next note in the new note is $E4$. Thus our new song is the following sequence: $C4, D4, E4$. We continue the process until we get the desired song length. Figure 2.1 shows the step-by-step action of the process just described.

2.2 Design and Testing of Markov Processes

With the model clearly established we now focus on the design process of our software. The first step in the process was to obtain the note data for a chosen song. The song is taken as input in the form of a MIDI file which is the standard file format for computer music. To manipulate such files we use MATLAB's MIDIToolbox package [9, 8] along with a collection of methods which we created. The functions provided allowed us to import MIDI files and construct a matrix containing information about a song such as the pitch, duration, and onset of each note. With the matrix we take the column corresponding to the pitches and construct the transition matrix in the same manner covered in the example from Section 2.1. We use the subset of all unique pitches as the states. The transition matrix generated will be a $n \times n$ matrix where n is the number of unique pitches. With the transition matrix, we begin our construction of the new improvised solo. We start our solo by choosing the first note (called the seed) of the original solo to be the first note of the new solo. This is not a requirement but merely a choice of implementation. The method still performs the same even if the seed was randomly chosen. With the seed, we use a randomly generated number to choose the next note based on the probabilities corresponding to the accessible states. We then change the seed and repeat the process. With the new solo, represented as a vector, we overwrite the pitch column of the song matrix with the new solo, maintaining the duration and onsets of the original song.

To listen to the newly constructed solo and determine its melodiousness we played it along with a backing track within the same musical key. To do this we used the software REAPER [11] which allows us to manipulate MIDI files and combine several MIDI tracks into one unified track. This software gave us the opportunity to adjust tempo issues if they occurred when trying to sync the MIDI tracks. The piece that we chose to use for our simulations was 'So What' by trumpeter Miles Davis. This piece was chosen since it is well known piece in Jazz music. We chose to use the solos of Miles Davis, guitarist George Benson, and saxophonist John Coltrane. These solos were chosen since these are the most popular solos recorded for the piece and thus their transcriptions are readily available. Using the solo for each artist we generated a transition matrix representing each artist and then randomly generated a new solo equal in length to the original solos. Figures 2.2-2.4 are each artist's solo overlayed with a new solo generated from their respective 1^{st} order Markov Chains. The x-axis represents the time step while the y-axis represents the MIDI number of the notes. The variation in the original solo is much greater than the solo generated from the model. The generated solo has a much tighter oscillation pattern which shows the 'dominate' note transition that was present in the original solo. This phenomena is present in all three figures for the 1^{st} order case.

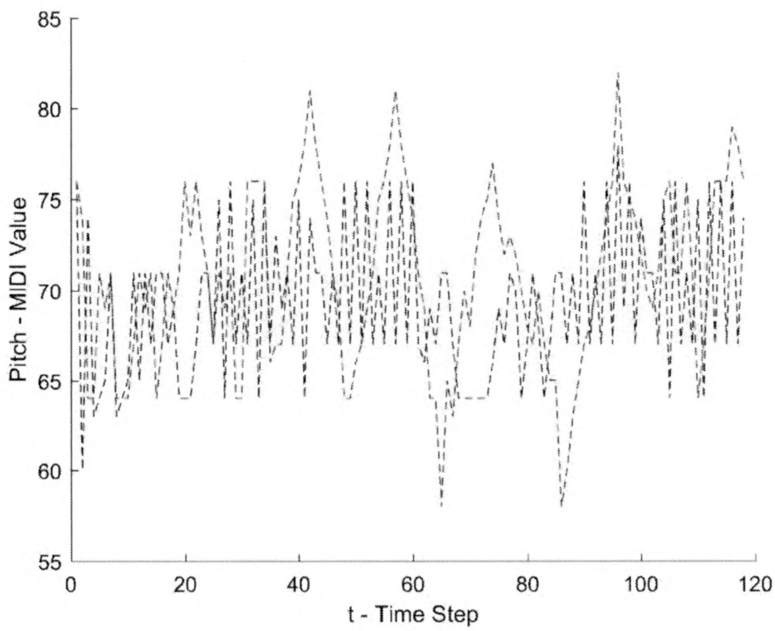


Figure 2.2: Miles Davis Solo(red) with 1st order MC(blue)

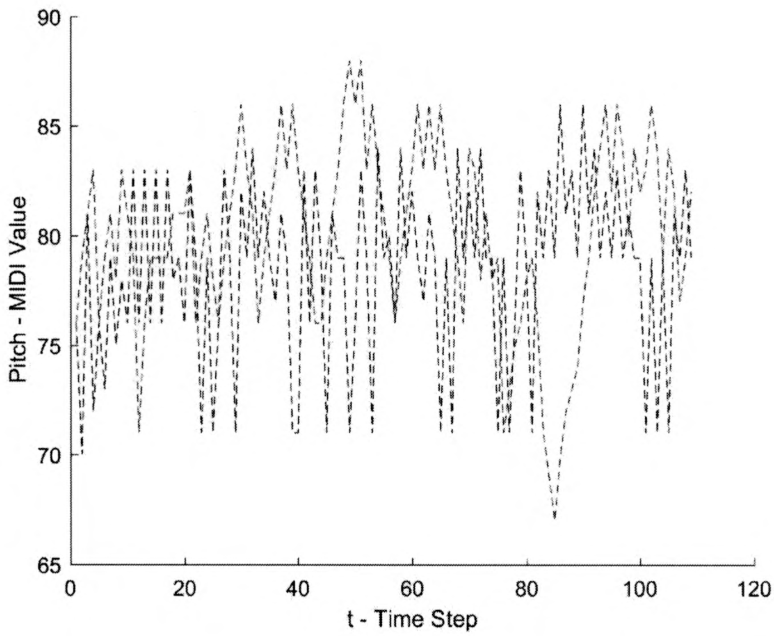


Figure 2.3: John Coltrane Solo(red) with 1st order MC(blue)

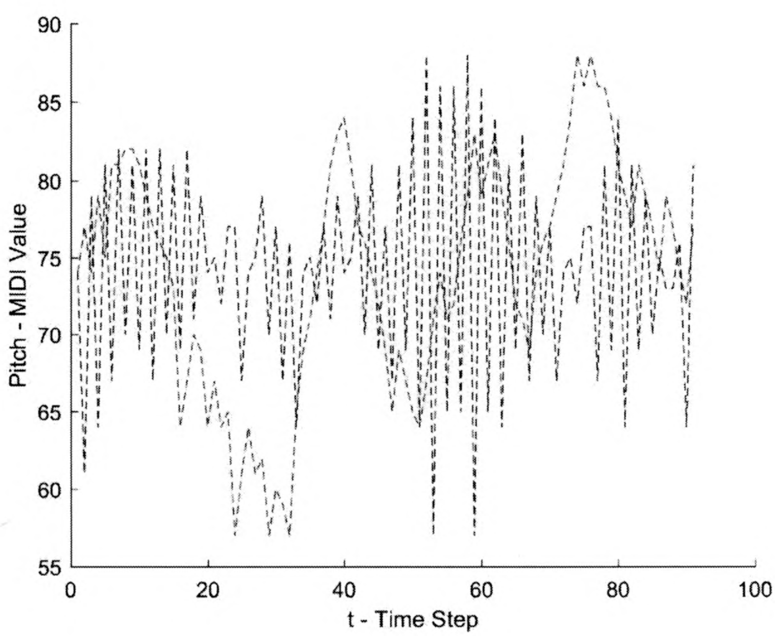


Figure 2.4: George Benson Solo(red) with 1st order MC(blue)

We next consider the case of n^{th} order Markov Chains. These are Markov chains in which the probability that a note is going to occur depends upon whether a certain sequence of n notes already occurred. In this case our state space now contains not only just the single notes but all combinations of sequences of n notes that occurred within the original solo. The process for constructing a new solo remains the same as the 1^{st} order case. The following graphs show the similarities and differences between the original and new solo. Figures 2.5 - 2.7 are each artist's solo overlayed with a new solo generated from their respective 2^{nd} order Markov Chains. All three solos generated for each artist seem to show more variation and fewer oscillations amongst the same notes than the 1^{st} order case. This seems to stem from the fact that using pairs of notes (or sets of notes in general) to represent a state rather than using single notes improves variability by lowering the chance of encountering three note length sequences which continuously repeat.

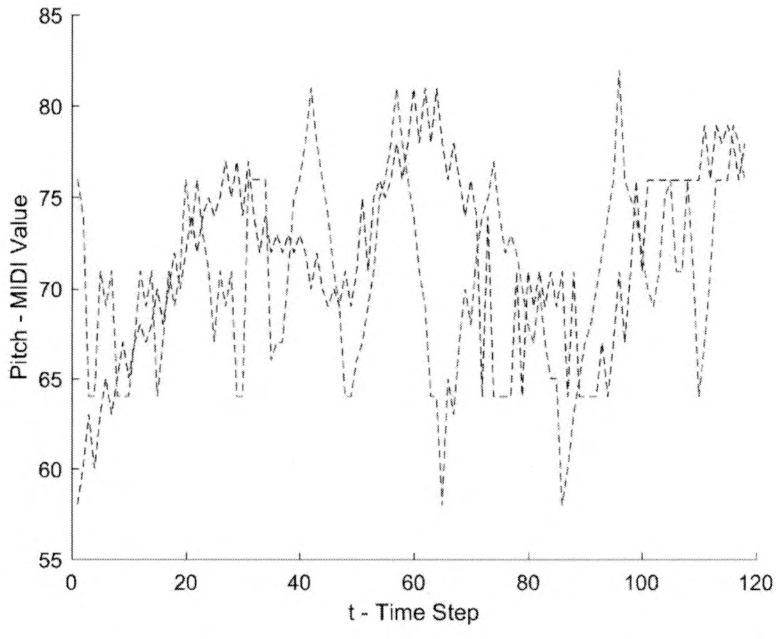


Figure 2.5: Miles Davis Solo(red) with 2nd order MC(blue)

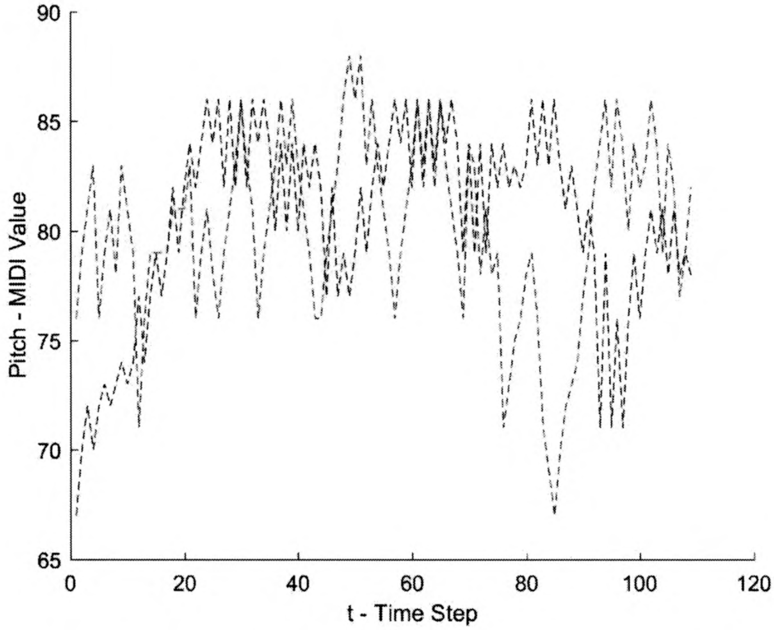


Figure 2.6: John Coltrane Solo(red) with 2nd order MC(blue)

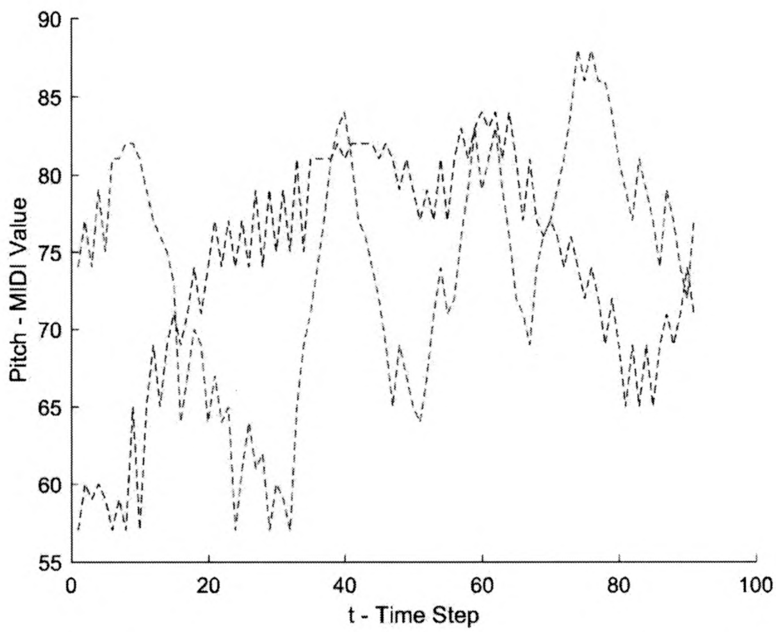


Figure 2.7: George Benson Solo(red) with 2nd order MC(blue)

We considered upto the case when $n = 4$. It becomes evident that as n increases, the original and new solos become the same. The reason for this is that as the order n of the Markov chain increases the more of the piece the chain "remembers". Consequently this produces more of the original piece. Consider the "Mary Had a Little Lamb" example from Section 2.1. If we considered a 5^{th} order Markov Chain of this piece, then we would have to consider the probability of say, $C4$ coming after $E4, D4, C4, D4, E4$ which are the first 5 notes of the piece. So this demonstrates that the chain has "recollection" of the first 5 notes. If for example we considered instead a 24^{th} order Markov Chain and considered the same probability event of $C4$ being our next note, we would get the entire piece. It is worth noting that we also attempted this construction for the note values of the original solo. However we found that it did not yield any musically worthwhile results since the dominating note note values were eighth notes for all solos considered. In order for there to be an interesting duration sequence for our new solo, we would have to consider solos which have a wide variety of note values without one that is overly dominating. The method seems to have some success in capturing the style of the artist. This can be seen partly from the Figures in which there is is some overlap of the graphs as well as similar distances maintained between successive notes. This is also the case from an aural perspective. Certain passages in the solos generated seem to represent a course of action that the artist may have taken. However, sometimes the defining traits of an artist's improvisation style are techniques that may not be reoccurring themes in their solo. To determine whether our model captures and is able to reproduce these subtleties in the artist's style requires a more mathematical approach.

2.3 Analysis of Markov Process

We will now offer analysis of the models focusing on the 1^{st} order Markov Chain case. To determine the effectiveness of the model in the replication of an artist's improvisation style we employ the techniques from Probabilistic model checking. Particularly we consider the construction of temporal logic statements to test whether particular traits in the artist's style is captured. To our knowledge this is the first time in which probabilistic model checking techniques has been employed to test for musical traits. This approach provides a way of constructing logic statements that can represent complex musical techniques and determine whether a given model will be able to produce such an event. We seek to demonstrate the application of this approach in testing whether our model replicates Miles Davis' improvisation style. One particular musical technique that Miles Davis is known to demonstrate in his solos is that of nonharmonic tones (i.e. the transition from scale tones to non-scale tones and back). For example, if we consider a song in the key of C major, according to the diatonic scale the only notes that can occur are: C, D, E, F, G, A, B. However, if we were playing a song in which the first two notes were C and D, then we could add the passing tone $C\#$ (i.e. a non-scale tone) in between the C and D. Note that nonharmonic tones do not have to strictly be minor second distance. So for example going from C to $C\#$ to E also counts as a case of nonharmonic tone usage. To test for this kind of melodic transition, we constructed the following logic statement

$$s_i \wedge X(S - \{s_i\}) \wedge X(X(s_i)) \tag{2.1}$$

where $\wedge$ is the conjunction operator (i.e. all events have to be true for the entire statement to be true) and the X is the next temporal operator. Temporal operators in general are boolean functions that output true or false depending upon whether the statement it is referring to comes true or not after a specified amount of time. The next temporal operator X simply checks that the value that occurs next will return a value of true to a previously defined condition. So we are using the above logic statement to determine whether the value s_i is going to be in the key of the song followed immediately by a nonharmonic tone and then immediately back to a tone in the key.

The value s_i can be any value within the key signature. Since the song is in the key of C, the only values that s_i can take are $\{60, 72, 74, 64, 76, 65, 77, 67, 69, 81, 71, 79\}$ since these are the MIDI values of pitches that the song uses which are in the key of C. The set $\{s_i\}$ refers to the collection of all values that s_i can take while the S is the set of all values that occurred in the solo (i.e. all notes that were used in the solo by Miles Davis). Thus $\{s_i\}$ consists of the list of MIDI values listed above and $S - \{s_i\}$ are all the other MIDI values that s_i cannot be. Using the probabilistic model checking software PRISM [19], we input the Markov chain generated from the Miles Davis solo and input the following statements:

$$Pr_{\geqslant 0.5}[s_i \wedge X(S - \{s_i\}) \wedge X(X(s_i))] \tag{2.2}$$

$$Pr_{=?}[s_i \wedge X(S - \{s_i\}) \wedge X(X(s_i))] \tag{2.3}$$

The first statement tests whether the probability of running into a nonharmonic tone is greater than 0.5 while the second statement computes the actual probability of running into a nonharmonic tone. PRISM generated that the nonharmonic tone will occur with probability 1 with an initial state of E. So it can be concluded that in every execution of our software, any solo that is generated will contain nonharmonic tones since the probability of this event occurring is 1. This result shows that our model is successful in producing solos which will contain a particular aspect (in this case the use of nonharmonic tones) of Miles Davis' improvisational style.

Similarly, we will now consider the validity of the model in replicating the improvisation style of John Coltrane. John Coltrane was known for using modes to construct his solos. Modes are a collection of scales built from the notes of the diatonic scale of the primary key. In his album Ascension, Coltrane is shown to have employed a technique called an extension of a perceptual space [12]. This involves taking a chord and for every note in the chord play its corresponding mode. The series of modes that are played can be organized in different structures. For example in his piece "A Love Supreme" on the album Ascension, Coltrane imposes a symmetric structure to the series of modes used in which he plays a modal series of aeolian, phrygian, phrygian, aeolian. Because of his heavy use of modes, an effective model of Coltrane's style requires the testing of the model to produce solos centrally rooted in modal soloing. Through analysis of the original piece, the primary mode used was the Dorian mode. The C Dorian scale is C, D, Eb, F, G, A, Bb, C which only differs from the diatonic scale by the third note which was raised a semitone and the seventh note of the diatonic scale which was lowered a semitone. Thus we choose to test whether the Markov chain will yield a solo that is based upon the C Dorian scale. Because the C Dorian scale is very similar to the C Major scale we chose to specifically compute the probability that a solo will be generated by our model which contains either Eb or Bb. To compute this probability we use the following logic statement

$$P_{=?}[F(s_i \wedge (Eb \vee Bb)$$

s$_j \wedge (Eb \vee Bb))](2.4)$

This statement computes the probability that eventually a four note sequence will occur which will consist of a note from the mode (or key since they are similar) followed by either an Eb or Bb , then another note from the mode, and finally either Eb or Bb again. Using PRISM to compute this probability, the model was shown to generate such a four note sequence with probability 1.

Doing the same procedure for our George Benson model, we attempted to determine the possibility of our model generating arpeggios since this is common technique that he employs in his solos. Because arpeggios are just melodic expression of chord tones, there are various ways in which they can be played. For example, if we consider the C major triad, the following are all the possible arpeggios that can occur:

$\{C, E, G\}, \{E, G, C\}, \{G, C, E\}, \{C, G, E\}, \{E, C, G\}, \{G, E, C\}$. Although some of these combinations may not be the "typical" way of playing it, these cases must also be considered. Also extending this same idea to 7^{th} chords, the number of possibilities increase. Thus we limit our testing strictly to triads, particularly major triads. To test for this we provide the following logic statement

$$P_{=?}[F(s_i \wedge ((s_{i+1} - s_i) = 4 \wedge ((s_{i+2} - s_i) = 7))] \tag{2.5}$$

The above logic statement allows us to compute the probability that at some point in the execution of the model a note s_i will occur followed by a note which is a major thirds distance from s_i and then immediately followed by a note which is a perfect fifths distance from s_i. To make sure that the successive notes are a major third and perfect fifths distance away from s_i respectively, we check whether the difference(i.e. the distance) between the MIDI value of the next note and s_i is equal to 4 semitones and 7 semitones respectively. The results of PRISM yielded a probability of 1 of a major triad in root position occurring. Particularly it was shown that an A-major triad in root position will occur with probability 1. Further testing is required to determine other types of arpeggios (i.e. arpeggiated seventh chords, arpeggiated ninths,etc) that may have been captured by our model. Testing is also needed to determine whether the model would yield the same results if a different initial state was chosen. This analysis has demonstrated the effectiveness in testing these kinds of models using probabilistic model checking. This approach provides a convenient way to analyze such models without having to focus on a case by case analysis of generated solos. Analyzing one particular generated solo can lead to incorrect conclusions since the probabilistic nature of the problem could yield results that in general are unlikely to occur in most executions of the model. So any analysis (regarding the traits of the artist's style that is captured in the model) that can be done, which strictly rely upon the generated solos, must be done using a collection of executions.

Chapter 3

Markov Chain Monte Carlo

We next consider a different construction of the transition matrix to represent a solo. In the previous construction we decided to choose the state space to be all unique pitches. This meant that we consider a different frequency of the same tone to be a unique note. For example an A occurring in the forth octave is a unique note from an A occurring in the third octave. However the new construction will not differentiate between the frequencies but simply take it as a representation of the tone (i.e. the note A is taken to be the same whether it is in the third octave or the fourth octave). To preserve the octave that the artist is playing in (so as to avoid extremely high or extremely low played solos) we constructed a separate matrix that tracks the octave that a particular tone was played in. This is useful for the creation of the new solo. To find our stationary distribution, we had to make sure that our transition matrix had two properties. The first property is that our transition matrix had to be aperiodic.

Definition. *A state i is said to have period r if $P_{ii}^{(n)} = 0$ whenever n is not divisible by r, and r is the largest integer with this property. A state i with period 1 is said to be aperiodic.*

Musically speaking this led to choosing improvised solos of an artist which did not have repetitive patterns. Any improvised solo with repetitions as the dominating theme of the solo would yield a period equal to the length of the repetition. The second property that our matrix must possess is that of irreducibility.

Definition. *A transition matrix P is irreducible if it is possible to go from any state to any other state within the state space.*

Because we chose to define the state space as all 12 tones, we were guaranteed this property as long as every tone was used in the artist's solo. Since our transition matrix is both irreducible and aperiodic, the following theorems guarantee us a vector π which possesses a unique interpretation with regards to Markov chains.

Theorem (Perron-Frobenius Theorem). *If a matrix P is irreducible, then there exists exactly one eigenvector π with $P\pi = \pi$. Furthermore, π can be chosen such that all its entries are strictly positive. If P is aperiodic, all other eigenvalues satisfy $|\lambda| < 1$.*

Theorem. *Let P be irreducible and aperiodic and let π be its Perron-Frobenius vector. Then for any probability measure $v \in R^N$, one has $lim_{n \to \inf} P^n v = \pi$*

The vector π is called the Perron-Frobenius vector. This vector is the stationary distribution for our transition matrix. The i^{th} component corresponds to the probability that the i^{th} state (i.e. a particular tone)will occur over a period of time. Because the transition matrix is our characterization of the artist's style, we therefore make the assertion that the vector π is an equivalent representation since we can always reconstruct the transition matrix from this vector. This allows us to only have to store our vector representation of our artist instead of maintaining a transition matrix for each.Therefore computing of transition matrices for artist and generating their corresponding stationary distribution can be considered a preprocessing step. We then can just keep the stationary distributions which requires less storage space. We can effectively compute π by applying standard numerical techniques such as the QR algorithm. However because we are considering a single example for testing this construction, it suffices to take large powers of the matrix knowing that it would converge yielding π.

To generate our note sequence we use Markov Chain Monte Carlo methods to sample from our stationary distribution. We apply the Metropolis Hastings (MH) Algorithm for this. Starting with an initial note b (which we are still considering to be the first note of the original solo), the MH algorithm starts with proposing a state i with the conditional probability density $q(b,*)$ given that the previous state is b. From there we compute the Hastings Ratio,

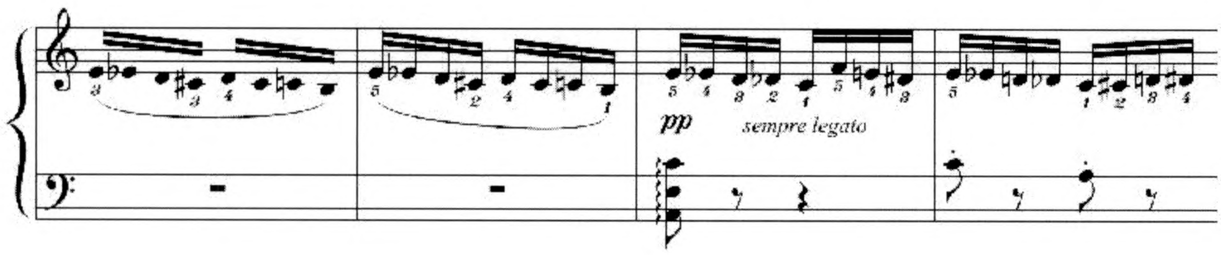
$$r(b,i) = \frac{s(i)q(i,b)}{s(b)q(b,i)} \tag{3.1}$$

where $s(*)$ is the density function of the stationary distribution. We then choose to accept the proposed note by computing $\alpha(b,i) = min(1, r(b,i))$ and comparing it to a number u which was generated from a uniform distribution. If $u < \alpha$ then we accept the proposed note. Otherwise we reject and choose the next note in the sequence to be the previous note b.

3.1 Example of Markov Chain Monte Carlo

Let's now consider an example of this construction using an excerpt from the classical piece 'The Flight of the Bumble-Bee' by composer Nikolai Rimsky-Korsakov. We will use only the first 8 measures of the piece that are shown below.

Using the same process as previously illustrated in Section 2.1, we construct the following transition matrix.



	C	$C\#$	D	$D\#$	E	F	$F\#$	G	$G\#$	A	$A\#$	B
C	0	${}^{1}/_{8}$	0	0	0	${}^{1}/_{8}$	0	0	0	0	0	${}^{6}/_{8}$
$C\#$	${}^{5}/_{10}$	0	${}^{5}/_{10}$	0	0	0	0	0	0	0	0	0
D	0	${}^{10}/_{11}$	0	${}^{1}/_{11}$	0	0	0	0	0	0	0	0
$D\#$	0	0	${}^{6}/_{8}$	0	${}^{2}/_{8}$	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0	0	0	0
$F\#$	0	0	0	0	0	1	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0	0	0	0
$G\#$	0	0	0	0	0	0	0	1	0	0	0	0
A	0	0	0	0	0	0	0	0	1	0	0	0
$A\#$	0	0	0	0	0	0	0	0	0	1	0	0
B	${}^{1}/_{3}$	0	0	0	0	0	0	0	0	0	${}^{2}/_{3}$	0

With this transition matrix, we will now compute the stationary distribution. By taking increasing powers of the above matrix it will converge to matrix with repeating rows. The row that repeats is the stationary distribution. By taking the above matrix to the 100^{th} power, we obtain the following matrix. To reiterate we chose to take the 100^{th} power simply because we knew, through testing, that the matrix would converge to yield the stationary distribution after raising to this power. Note that this is also guaranteed to us by Theorem 3. However to determine the convergence of the matrix such methods as the QR algorithm should be used.

$$\begin{array}{c} \\ C \\ C\# \\ D \\ D\# \\ E \\ F \\ F\# \\ G \\ G\# \\ A \\ A\# \\ B \end{array} \begin{array}{c} \begin{array}{cccccccccccc} C & C\# & D & D\# & E & F & F\# & G & G\# & A & A\# & B \end{array} \\ \begin{pmatrix} 0.1250 & 0 & 0.4167 & 0 & 0.3750 & 0 & 0.0833 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1250 & 0 & 0.4167 & 0 & 0.3750 & 0 & 0.0833 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1250 & 0 & 0.4167 & 0 & 0.3750 & 0 & 0.0833 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1250 & 0 & 0.4167 & 0 & 0.3750 & 0 & 0.0833 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The stationary distribution is

$$\pi = \begin{array}{c} \begin{array}{cccccccccccc} C & C\# & D & D\# & E & F & F\# & G & G\# & A & A\# & B \end{array} \\ \begin{pmatrix} 0.1250 & 0 & 0.4167 & 0 & 0.3750 & 0 & 0.0833 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{3.2}$$

where each number corresponds to the probability that the note will occur. Using this distribution we will now implement the Metropolis Hastings(MH) algorithm. We will use the transition matrix from the piece Fur Elise to represent our arbitrary density function to make note proposals. Below is its transition matrix.

	C	$C\#$	D	$D\#$	E	F	$F\#$	G	$G\#$	A	$A\#$	B
C	0	0	$^{1}/_{13}$	0	$^{6}/_{13}$	0	0	0	0	$^{4}/_{13}$	0	$^{2}/_{13}$
$C\#$	0	0	0	0	0	0	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0	0	0	0	0
$D\#$	0	0	0	0	1	0	0	0	0	0	0	0
E	$^{2}/_{22}$	0	0	$^{8}/_{22}$	$^{2}/_{22}$	0	0	0	$^{2}/_{22}$	$^{4}/_{22}$	0	$^{4}/_{22}$
F	0	0	0	0	0	0	0	0	00	0	0	
$F\#$	0.1250	0	0.4167	0	0.3750	0	0.0833	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0
$G\#$	0	0	0	0	0	0	0	0	0	0	0	1
A	$^{4}/_{10}$	0	0	0	$^{1}/_{10}$	0	0	0	0	0	0	$^{5}/_{10}$
$A\#$	0	0	0	0	0	0	0	0	0	0	0	0
B	$^{3}/_{13}$	0	$^{4}/_{13}$	0	$^{4}/_{13}$	0	0	0	0	$^{2}/_{13}$	0	0

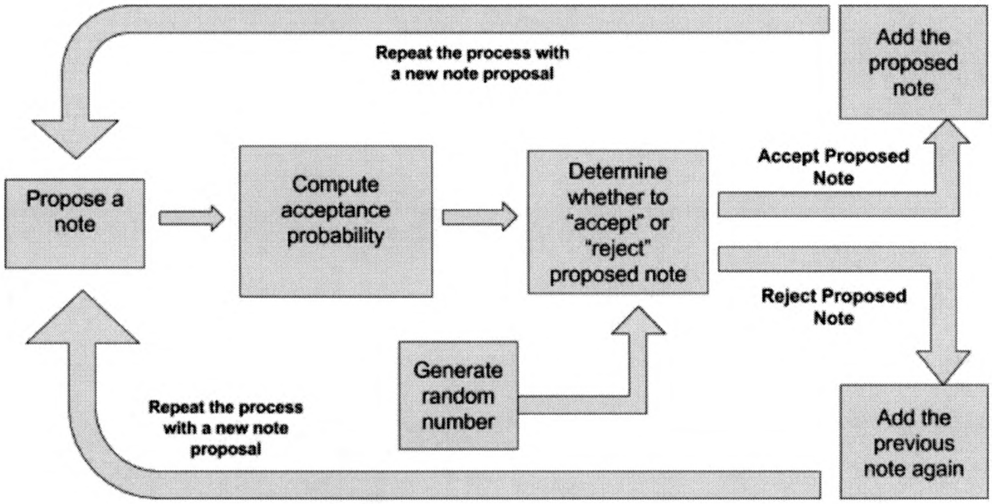


Figure 3.1: Illustration of the MCMC Protocol

We will choose the first note of the new piece to be D. We first start the algorithm by proposing a note. Note that in the following equations, $q(*|*)$ (from 3.1)will be values from the Fur Elise transition matrix. The conditional density function $s(*)$ used in equation 3.1 will be represented by the Perron-Frobenius vector π. The input of both of these functions will be either the proposed note or the previous note which refer to the states i and b listed in equation 3.1. The following calculations are the implementation of the MH Algorithm. The general process is outlined in Figure 3.1.

Iteration 1:

1. Propose Note: C

2. Compute the acceptance probability:

$$\begin{aligned}\alpha(C|D) &= min\{1, \frac{q(D|C)\pi(C)}{q(C|D)\pi(D)}\} \\ &= min\{1, \frac{{}^{1}/_{13} * 0.1250}{1 * 0.4167}\} \\ &= min\{1, 0.02307\}\end{aligned} \tag{3.3}$$

3. Generate a random number (Note: used MATLAB rand function)

$$u = 0.007 \tag{3.4}$$

4. Determine whether to accept or reject proposed note

$$0.007 = u < \alpha = 0.02307 \Rightarrow AcceptNote : C \tag{3.5}$$

5. Add note to new song

New Song = $[D, C]$

Iteration 2:

1. Propose Note: E
2. Compute the acceptance probability:

$$\begin{aligned}\alpha(E|C) &= min\{1, \frac{q(C|E)\pi(E)}{q(E|C)\pi(C)}\} \\ &= min\{1, \frac{{}^{2}/_{22} * 0.3750}{{}^{6}/_{13} * 0.1250}\} \\ &= min\{1, 0.59089\}\end{aligned} \tag{3.6}$$

3. Generate a random number (Note: used MATLAB rand function)

$$u = 0.0442 \tag{3.7}$$

4. Determine whether to accept or reject proposed note

$$0.0442 = u < \alpha = 0.59089 \Rightarrow AcceptNote : E \tag{3.8}$$

5. Add note to new song

New Song = $[D, C, E]$

Above is the sheet music of the example just computed. We only showed the first two iterations to highlight the process, but the implementation of this method allows for us to specify the number of notes we want in our new solo. Thus if we specified the number of notes in our new solo to be 60, the software would keep applying the method until 60 notes were generated. Because we are only focusing on constructing new melodic phrases (i.e. new pitches), we chose to generate new solos equal in length to the original solo in order to use the note values of the original solo. In this example all of the notes that we proposed were accepted. However if $u > \alpha$, then we would reject the proposed note and just add the previous note. Thus for the first iteration, if we rejected the proposed note our new song after iteration 1 would have been $[D, D]$. Figure **??** shows a solo generated when the number of iterations is equal to 21. Observe how there is not much variability in the generated piece since there are not many notes that are available to propose (i.e. since alot of the notes have probability of 0 of occurring). This characteristic in the model will be elaborated upon in the analysis of the model.

Figure 3.2: MCMC generated solo after 21 Iterations

3.2 Analysis of Markov Chain Monte Carlo

The Jazz composition that we chose for this simulation was 'Anthropology' [15] by saxophonist Charlie Parker. This piece was chosen for the same reasons as the composition in the previous section. This piece also had the added advantage in that it uses all 12 tones. Using our new method we generated the invariant measure π. Below is the graph of Charlie Parker's original solo [15] overlayed with the new solo generated from our proposed method. The sheet music of Parker's original solo as well as the new solo can be found in the Appendix.

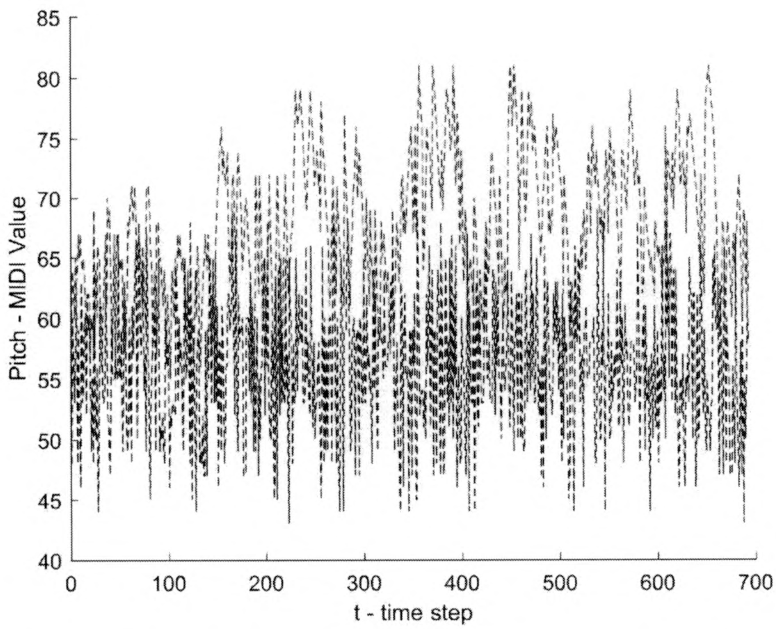


Figure 3.3: Charlie Parker(red) with Metropolis Hastings generated Solo(blue)

As in the analysis of the previously discussed model, we seek to determine the effectiveness of this new construction in generating solos which mimic the improvisational techniques utilized by Charlie Parker. A particular technique that we tested for was that of chromatic playing (i.e. playing which consists primarily of semitone movement). Because of the similarity of the test to the one used for the Miles Davis model, we used the same logic statement used in that analysis. However since we consider first the pitch classes in constructing our transition matrix and then the octave later in the final construction of the solos, we had to create a multi-layered model for correct representation in PRISM. This led to creating a pair of synchronized PRISM modules in which one simulated the transitions between the pitch classes and the other determined the choice of which octave the note will occur. The simulations showed that chromatic playing rarely occurred and only of 2 note length. For example, the two note sequence $B \rightarrow C$ would occur with probability 1. However considering the probability of either C or B (The only two notes playable to be considered chromatic playing) immediately after the following sequence would yield a probability of 0. Aurally, the generated solo sounds similar to about the initial 20 notes of Charlie Parker's solo. However as it continued, it seemed to deviate completely in theme. We will consider several reasons for this outcome. One initial cause for this is the generalization of the notes to the pitch classes. The model discussed in Chapter 2, would differentiate between say $C4$ and $C3$. So for example in a particular solo if the note $C4$ only occurs once and the note $C3$ occurs five times then the first method will assign more importance to (via having a higher probability) to $C3$ than to $C4$. However, the MCMC method views these as the same note and therefore it can be viewed as assigning from one perspective the same probability to $C4$ and $C3$ and from another perspective increasing the probability of C occurring in the generated solo. Thus its providing a slight normalizing affect to notes that are not as popular and simultaneously increasing the probability of a particular pitch class occurring in the generated solo. Another, much more significant issue that could yield to a less accurate representation of an artist by our model is the "burn in" [3] period of the MH algorithm. Because the MH algorithm (and MCMC methods in general) relies upon the need for an initial distribution to converge to the stationary distribution, the first few initial notes are not indicative of the nature of the stationary distribution. The "burn in" is thus thought of as the number of iterations needed to be discarded starting from the initial execution of the MH algorithm. Because we did not consider the "burn in" period in our implementation of the MH algorithm we therefore included in our solo a sequence of notes which may not have been yielded from the Charlie Parker transition matrix. In [3], Brooks et al recommends removing the first 100 outputs from the algorithm. However to determine the number of notes to discard from the solo requires further testing. Having to consider a burn in period can also be avoided if the initial distribution is close to the stationary distribution. The last reason that may cause inaccurate solos is the choice of the initial distribution (represented by a transition matrix) with regards to the probabilities between the notes. If majority of the entries (i.e. the probability of going from one note assuming a certain note has occurred) are 0 in the transition matrix then computing the Hastings ratio (Equation 3.1) could become difficult. The problem lies in that if a significant number of the entries in the transition matrix are zero then it limits the possibility of states that is possible for successful note proposals in the algorithm. Consider once again the example in Section 3.1. If we start from the first iteration where the starting note was D it is impossible to have a proposed note of E since in the Hastings ration $q(E|D) = 0$ since the $[D, E]$ entry in the Fur Elise transition matrix is 0. However if a better transition matrix was used, it will allow for more of a variety of note proposals and therefore more note variety in the solos generated. Thus an interesting line of study would be to determine an effective way of recognizing songs which are similar and would therefore yield similar distributions. This could possibly lead to a style classification of artist.

Chapter 4

Conclusion

In this work we considered the use of Markov Chains to construct software that mimics the improvisation styles of Jazz artists. We first analyzed the use of 1^{st} and ultimately n^{th} order Markov Chains in fulfilling this objective. We observed from the analysis that some of the models constructed yielded replication of improvisation techniques that the artists was known to employ. We then proposed the construction of Markov Chain using pitch classes and showing how it yielded a compact representation of the artist's tendencies. It was clear from the analysis that there was a clear trade-off between the accuracy of the model in representing the artist's style and the memory-efficient compact representation of the artist. Further analysis is needed in determining whether the given models captured more complex improvisational techniques used by the artist. An example of such complex techniques would be using chord changes to construct solos. Determining chords from a MIDI file is quite challenging and thus a different approach may be needed to retrieve this level of musical data from the file. As stated previously, it is for this reason that we focused on saxophone and trumpet musicians since the solos they generate are melodic in nature (i.e. as opposed to harmonic). A possible alternative approach worth exploring is the distinguishing of chords from the perspective of frequency. Establishing an effective method of extracting chord data from a piece can lead to modeling and possibly replicating the improvisation styles of pianist like Oscar Peterson and Bill Evans. Continuation of this work also includes the improvement of our pitch class model in giving a more accurate representation of the artist while maintaining a relatively compact representation.

Chapter 5

Appendix

So What Solo

Miles Davis

So What Solo

John Coltrane

So What Solo

George Benson

First Order Miles Davis Solo

[Composer]

Second Order Miles Davis

[Composer]

First Order John Coltrane

[Composer]

Second Order John Coltrane

[Composer]

First Order George Benson

[Composer]

Second Order George Benson

[Composer]

Anthropology

Charlie Parker


3
3
3
3
3
3
3

Metropolis Hastings Solo

[Composer]


3

Bibliography

[1] Bergomi, Mattia G., and Alessandro Portaluri. "Modes in Modern Music from a Topological Viewpoint." [1309.0687] Modes in Modern Music from a Topological Viewpoint. N.p., 03 Sept. 2013. Web. 17 Apr. 2017. ¡http://arxiv.org/abs/1309.0687¿.

[2] Dave Benson, "Music: A Mathematical Offering",United Kingdom, 2008.

[3] Brooks, Steve, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng. Handbook of Markov Chain Monte Carlo. Boca Raton, Fla: CRC, Taylor Francis Group, 2011. Print.

[4] Burgoyne, John Ashley, Jonathan Wild, and Ichiro Fujinaga. "Compositional Data Analysis of Harmonic Structures in Popular Music." Mathematics and Computation in Music Lecture Notes in Computer Science (2013): 52-63. Web.

[5] Karlin, Samuel, and Howard M. Taylor. A First Course in Stochastic Processes. New York: Acad., 2011. Print.

[6] Dudley, Richard M. Real Analysis and Probability. Cambridge: Cambridge U, 2011. Print.

[7] Martin Hairer, "Ergodic Properties of Markov Processes", The University of Warwick, 2006.

[8] "KenSchutte.com." Ken Schutte. N.p., n.d. Web. 16 Apr. 2017.

[9] "Midi Toolbox." Midi Toolbox - Humanistinen Tiedekunta. N.p., 20 May 2016. Web. 17 Apr. 2017.

[10] MATLAB. Natick, MA: MathWorks, 2006. Computer software.

[11] REAPER. Computer software. Vers. Trial. Cockos Incorporated, 2004. Web. ¡http://www.reaper.fm/¿.

[12] Mazzola, Guerino. Flow, Gesture, and Spaces in Free Jazz. Place of Publication Not Identified: Springer, 2014. Print.

[13] Milmeister, GeÌrard, and Guerino B. Mazzola. The Rubato Composer Music Software: Component-based Implementation of a Functorial Concept Architecture. Berlin: Springer, 2009. Print.

[14] Marom, Yuval. "Improvising Jazz With Markov Chains." CiteSeerX. N.p., n.d. Web. 17 Apr. 2017. ¡http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.721¿.

[15] Parker, Charlie. Charlie Parker Omnibook: For All Bass Clef Instruments: Transcribed from His Recorded Solos: Transported to Concert Key. Lynbrook, NY: Atlantic Music, 1978. Print.

[16] Agmon, Eytan. "A Mathematical Model of the Diatonic System." Journal of Music Theory 33.1 (1989): 1. Web.

[17] Muller, Stefan. "Musical Gestures Archives." Robotized. N.p., n.d. Web. 17 Apr. 2017. ¡https://robotized.arisona.ch/tag/musical-gestures/¿.

[18] Maurer, W. Douglas. "The Mathematics of Jazz." Bridges Proceedings (2004): 273-81. Web.

[19] "PRISM - Probabilistic Symbolic Model Checker." Www.prismmodelchecker.org. N.p., n.d. Web. 21 Apr. 2017. ¡http://www.prismmodelchecker.org/¿.

[20] Ames, Charles. "The Markov Process as a Compositional Model: A Survey and Tutorial." Leonardo 22.2 (1989): 175-87. JSTOR. Web. 23 Apr. 2017.