Simulating Music Using Markov Chains

Brijee Rana
BXR826

1 Introduction

This project will investigate using Markov chains to simulate music. I will begin with some motivations and history, then explain Markov chains. Markov chains holds a powerful assumption which I believe can model music as I think music holds that assumption. I will also discuss some interesting properties of Markov chains and how it can be applied. Then, I will simulate music using Markov chains and talk about my results as well as applying a powerful sampling technique called Markov chain monte carlo. In this project, the reader is assumed to have some knowledge in linear algebra and statistics.

I plan to provide further information and insight on Music and try answer questions like why certain chord progressions are arranged in a certain way or why Jazz uses certain notes compared to other styles of music. Technology in music production is continuing to develop more and more as AI is implemented. For example, GoogleAI's Magenta uses machine learning to create music and art. This inspired me to see if I can achieve the same results of synthesising music with using Markov Chains. I chose this statistical method because it has been successfully used in a wide range of topics. I hope to show why Markov chains are effective in simulation and analysis by the example of simulating music. Music composition software such as Max, SuperCollider and Csound use Markov chains in their algorithms.

1.1 History

In the early 1700s, Swiss mathematician Jakob Bernoulli refined the idea of expectation when he proved the law of large numbers. The theorem states that as the number of identically distributed, randomly generated variables increase, their sample mean approaches their theoretical mean. Bernoulli concludes that "If observations of all events could be continued for the entire infinity, it will be noticed that everything in the world is governed by precise ratios and a constant law of change.". In the early 1900s, Pavel Nekrasov and some philosophers, opposed the idea of a predetermined statistical fate. They thought that the law of large numbers might prevent us from having free will. Nekrasov made a famous claim that independence is a necessary condition for the law of large numbers where the outcome of previous events doesn't change the outcome of the current or future events. However, as we all can relate that most things in the physical world are clearly dependent on prior outcomes such as the chance of rain. This makes it so that the law of large numbers won't affect us and our free will as when the probability of some event depends on previous events, then we have dependent events or variables which goes against Nekrasov's claims.

Figure 1: Portrait of Andrei Andreyevich Markov (1856-1922)

Nekrasov's claim angered a Russian mathematician, Andrey Markov, as seen on figure 1, who disagreed with his claim. He states that "The law of large numbers can apply to dependent variables". He extends Bernoulli's results to dependent variables in which he introduces the idea of Markov chains. He proved that under certain conditions and properties, when you run a Markov chain, they reach an equilibrium. This means that regardless of how things begin, as the sequence continues, the distribution converges to some specific probability or ratio. This goes against Nekrasov's claim that only independent events could converge on predictable distributions.

1.2 Other applications

Markov chains are useful and applied in a variety of different topics such as computer science, physics, chemistry, maths, sports, economics, bioinformatics and more. Here are some more detailed examples:

- Natural language processing and speech recognition systems are based on hidden Markov models.
- Queuing theory, optimising the performance of telecommunications networks.
- Hidden Markov models are important in information theory.
- Finance is also another field where Markov chains can be modelled to the stock market.
- Bioinformatics, uses it for modelling evolution of biological populations and determining the population structure from genetic data.
- Computer vision uses Markov random field, which is a generalisation of a Markov chain with two or more dimensions so that the idea can be applied on pixels for edge detection and automated image analysis.
- Google pagerank algorithm uses Markov chains to rank websites based on the connections between them and is the central reason to Google's early success amongst other search engines.

2 Markov Chains

Definition 1. *Stochastic process: A stochastic process $X_t, t = 0, 1, ..., ,$ is a sequence of random variables with discrete states space S.*

A short example is that $X_t = i$ where $i \in S$ could be the state "rainy" on day t in a model about weather. Also, these random variables should not be independently and identically distributed.

Definition 2. *Markov property (1st order):*

$$\Pr(X_t = i|X_0, ..., X_{t-1}) = \Pr(X_t = i|X_{t-1}) \tag{1}$$

Markov property (nth order):

$$\Pr(X_t = i|X_0, ..., X_{t-1}) = \Pr(X_t = i|X_{t-1}, ..., X_{t-n}) \tag{2}$$

The Markov property is also a "memory-less property" because the future does not depend on the what happened before the present.For a 1st order process (1), only the current state in the sequence determines the next.

Definition 3. *A Markov chain is a stochastic process that has the 1st order Markov property with a discrete finite state space S and a transition probability P.*

Definition 4. *A homogeneous Markov chain has a transition probability matrix that does not depend on time t or step.*

I will mostly be discussing homogeneous discrete-time Markov chains in this project because I think this type of Markov chain will be the most useful at simulating music. This is because music has notes which are discrete which for my model is in the state space. When I refer to Markov chains I am referring to homogeneous discrete-time Markov chains.

2.1 Diagram

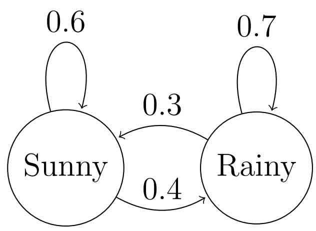

Figure 2: Markov chain diagram of weather

Example 1. *Figure 2 is a graphical representation of a Markov chain. It is a directed graph with states as the nodes and arrows that point to states. The arrows contain the probabilities of transitioning from one state to another. Figure 2 is a model of weather with only "rainy" and "sunny" states with probabilities of weather changing or staying the same shown in the arrows. Notice how there is a loop for each state so that just as in real life it is likely to rain the next day if it rained today. Every arrow coming out of a state has to have the probabilities adding up to 1.*

The probability of going from Rainy to Sunny is 0.3. Yet, if we want to calculate $Pr(X_0 = Rainy, X_1 = Sunny, X_2 = Sunny)$, we need to calculate $Pr(X_0 = Rainy)Pr(X_1 = Sunny|X_0 = Rainy)Pr(X_2 = Sunny|X_1 = Sunny)$. This new way of computing probabilities is due to the Markov property. To the find the initial probability that it will rain today we need to the initial probability of the state π_0, which is a row vector with the elements summing to 1.

2.2 Random walk

Definition 5. *(Random Walk) A random walk is when a model starts at a chosen state and randomly transitions to different states under the probability distribution of the model.*

Random walks have many applications in scientific fields because it is a useful model in recording stochastic activity. For my project, I will try to use a random walk to simulate music notes as I will record where the random walk goes in a Markov chain that construct from a dataset.

Example 2. *A random walk for figure 2 could be: rainy, sunny, sunny, rainy, rainy, sunny, rainy, ...*

2.3 Transition matrix

Definition 6. *(Transition matrices) The transition matrix P of a Markov chain X is a matrix with the probability of states transitioning between each other.*
The $(i,j)^{th}$ element in the transition matrix is the probability of state $X_t = i$ to $X_{t+1} = j$.
Every row of the transition must have its elements be non-negative and sum up to 1.

Theorem 7. *For any $t \in N$ and states $i, j \in S$, the state space, the matrix entry $(P_t \cdot P_{t+1})_{i,j} = \Pr(X_{t+2} = j | X_t = i)$.*

Proof. Denote $M = P_T \cdot P_{t+1}$. By matrix multiplication,
$M_{i,j} = \sum_{k=1}^{n} (P_t)_{i,k} (P_{t+1})_{k,j}$
$= \sum_{k=1}^{n} \Pr(X_{t+1} = k | X_t = i) \Pr(X_{t+2} = j | X_{t+1} = k)$
$= Pr(X_{t+2} = j | X_t = i)$
The final equality follows through from conditional probability.

□

Theorem 8. *(Chapman-Kolmogorov theorem) $P_{ij}^{n+m} = \sum_{k=0} P^{(n)} P_{kj}^{(m)}$.*

Proof. By defiinition, zz $P_{ij}^{(n+m)} = \Pr(X_{n+m} = j | X_0 = i)$
$= \sum_{k=0} \Pr(X_{n+m} = j X_n = k | X_0 = i)$
$= \sum_{k=0} \Pr(X_{n+m} = j | X_0 = i X_n = k) \Pr(X_n = k | X_0 = i)$
$= \sum_{k=0} \Pr(X_{n+m} = j | X_n = k) \Pr(X_n = k | X_0 = i)$

□

Example 3. *Here is the transition matrix of figure 2. Every row must add up to 1.*

$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$$

We can use the transition matrix to calculate the probability of each state at time t or step. We can apply matrix multiplication so we do not have to calculate with probability trees and is clearer that using conditional probability.

If $\pi_0 = (0.2, 0.8)$ is the initial probability, meaning that today it there is a 20% chance of rain and 80% to be sunny.

Then the probability of being in a particular state the next day is done by the calculation:
$\pi_1 = \pi_0 P = (0.2, 0.8) \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} = (0.46, 0.54)$.

This means that at time t=1, the weather is 46% *to be rainy and* 54% *to be sunny.*

To calculate the probability for each state at time $t = 2$*, you have to calculate* $\pi_2 = \pi_1 P = \pi_0 P^2$.

Hence, to get the k-th step of a Markov chain, we would have to compute $P^0 T^k$.

Due to some properties of Markov chains and under certain conditions, the probability distribution of the states can converge.

At k=50, $\pi_3 = (0.5614, 0.4386)$
At k=50, $\pi_6 = (0.5711578, 0.4288422)$
At k=50, $\pi_8 = (0.571404202, 0.428595798)$
At k=10, $\pi_1 0 = (0.57142..., 0.42857...)$

As you can see, the greater the number of steps, the more the system seems to converge to some numbers. I will continue to explain this convergence and why it is so important further down the project.

2.4 Markov chain properties

In this section I will talk about the basic properties and interesting features of Markov chains which will help us better understand the behaviour of the model and what we can achieve with it. As the graphical representation of a Markov chain is a directed graph, there is an overlap with graph theory. Some of these definitions are from reference [7].

Definition 9. *(Connected graph) A graph is a connected graph when there is a path from any point to any other point in the graph.*

Figure 2 is an example of a connected graph.

Definition 10. *(Periodicity) A state has a period of k, where k is the greatest common divisor of all the possible return path lengths. If* $k > 1$ *than that state is periodic. If* $k = 1$*, then that state is aperiodic. A Markov chain is aperiodic if all of its states are aperiodic. For an irreducible Markov chain, if one state is aperiodic then all states are aperiodic.*

Figure 2 is an example of an aperiodic Markov chain.

Definition 11. *(Reducibility) A Markov chain is irreducible if it is possible to reach any state from any other state. It does not matter how many steps this takes. If the state space S is finite, then the Markov chain can be represented by a directed graph and the graph of an irreducible Markov chain is a connected graph.*

It can also be said that a Markov chain is irreducible if there exists a chain of steps between any two states that has a positive probability.

Definition 12. *(Transience) A state is transient if, when we leave this state, there is a non-zero probability that we will never return to it.*

Theorem 13. *In a Markov chain with a finite state space, not all the of the states can be transient.*

Proof. Suppose that this was not true, then the Markov chain will run out of states not to go to in an infinite number of times. This is a contradiction. □

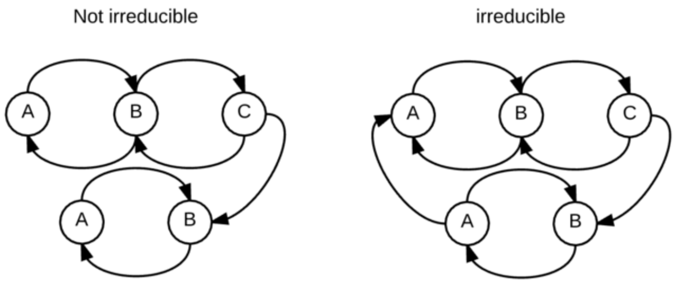

Figure 3: Graph on the left is reducible whereas the graph on the right is irreducible.

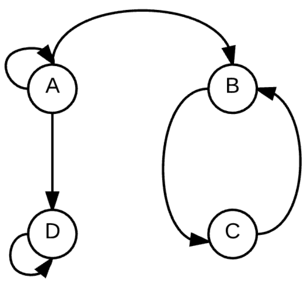

Figure 4: Graphs showing transience and recurrence

Definition 14. *(Recurrence) A state is recurrent if we know that we will return to that state, in the future, with probability 1 after leaving it.*

In figure 4, the state A is transient. States B, C, D are recurrent.

Definition 15. *(Ergodicity) A state is known as ergodic if it is positive recurrent and aperiodic. A Markov chain is ergodic if all its states are.*

Theorem 16. *Let X be an ergodic Markov chain with states 1,2,...,n and stationary distribution* $(x_1, x_2, ..., x_n)$*. If the process begins at state k, the expected number of steps* E_k *to return to the state k is* $E_k = \frac{1}{x_k}$*.*

Theorem 17. *An irreducible Markov chain has a stationary distribution if and only if the Markov chain is ergodic. If the Markov chain is ergodic, the stationary distribution is unique.*

Definition 18. *(Absorbing state) An absorbing state i is a state for which* $P_{i,i} = 1$*.*

It is important to understand absorbing states because they can exist when constructing a Markov chain, if a state is an absorbing state then the Markov chain is not irreducible and thus cannot converge. More information is given on the next section.

2.5 Convergence of Markov Chains

In this section I will further discuss what it means for a Markov chain to have an equilibrium, what conditions are required and how we can find a stationary distribution.

Definition 19. *(Stationary distribution) A stationary distribution of a Markov chain over the state space is a probability distribution that remains unchanged in the Markov chain as time progresses. Normally, it is represented as a row vector π whose entries are probabilities summing to 1, and given transition matrix P, it satisfies $\pi = \pi P$. This means that π is invariant by the matrix P.*

Theorem 20. *A Markov chain that is irreducible and aperiodic will have a probability distribution that will converge to the stationary distribution.*

Theorem 21. *An irreducible Markov chain has a stationary distribution if and only if the Markov chain is ergodic. If the Markov chain is ergodic, the stationary distribution is unique.*

Theorem 22. *An absorbing Markov chain has stationary distributions with nonzero elements only in absorbing states.*

Theorem 23. *Average time to return to State(X)* $= \frac{1}{Stationarydistributionatstate(X)}$

Here I will show some Markov chains that do not converge.
Case 1:

$$P = \begin{pmatrix} 0.7 & a & a & 0.3 \\ 0.4 & a & a & 0.6 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\pi_0 = (0.6, 0.2, 0.2, 0)$

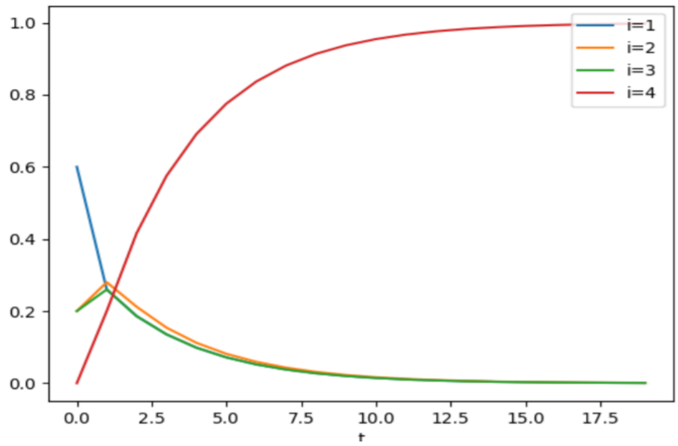

Figure 5: Convergence of Markov chain with an absorbing state

As you can see by the transition matrix, the system is bound to stay the the 4th state.

Case 2:

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$\pi_0 = (0.6, 0.2, 0.2)$

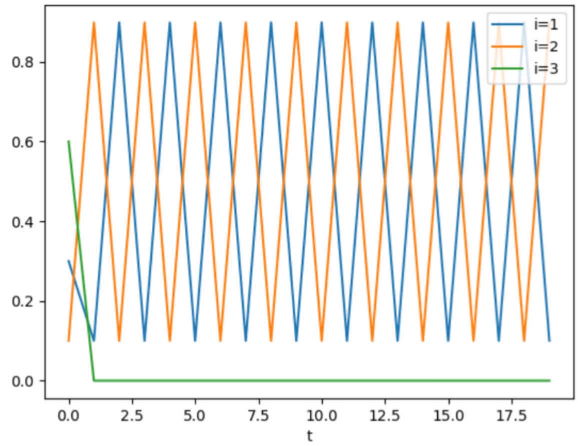

Figure 6: When there are no links to a state

As you can see by the transition matrix there is no link to state 3 therefore the system oscillates from state 1 to 2.

2.6 Linear algebra approach

To find the stationary distribution of a Markov chain given the conditions are met. We can take a linear algebra approach. From the definition, we need the row vector π to be invariant to the transition matrix i.e. this equation: $\pi P = \pi$. Notice how this equation is similar to the eigenvector equation, $Pv = \lambda v$ where the eigenvalue, $\lambda = 1$. π is a left eigenvector of the matrix P, instead how it is regularly in the equation. The elements of π must add up to 1, so it must be normalised if it is not already when finding the corresponding eigenvector to the eigenvalue 1. The eigenvector equation is expressed in column vectors, therefore I have to transpose the matrices.
$(\pi P)^T = \pi^T \implies P^T \pi^T = \pi^T$.

When there are multiple eigenvectors associated to an eigenvalue of 1, each such eigenvector gives rise to an associated stationary distribution. However, this can only occur when the Markov chain is reducible, i.e. has multiple communicating classes.

Example 4. *As an example, I will find the stationary distribution of figure 2 using this approach because figure meets our conditions for convergence.*

$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$$

. *When I transpose the matrix P, I can put into the eigenvector equation where I can find the eigenvalues.*
$P^T v = \lambda v$
Then: $(P^T - \lambda I) \cdot v = 0$
The equation has a nonzero solution if and only if $|P^T - \lambda I| = 0$. *det* $\begin{pmatrix} 0.7-\lambda & 0.4 \\ 0.3 & 0.6-\lambda \end{pmatrix} = 0$.

Thus, we have the equation: $\lambda^2 - 1.3\lambda + 0.3$. *From this we find that* $\lambda_1 = 0.3 and \lambda_2 = 1$. *It is vital that there is an eigenvalue = 1.*
The eigenvalue of 1 gives us the eigenvector $(\frac{4}{3}, 1)$.
Then, this means that when normalised so the elements sum up to 1, the stationary distribution must be $\pi = (\frac{4}{7}, \frac{3}{7})$.

3 Simulating Music

In this section, I will go through how I simulated music. As I am a guitarist, I will be constructing the Markov chain with chords. I have written my code which is based on an algorithm similar to reference [8]. The data I use from a certain genre or artist will make it so that the model is more likely to generate chord sequences from that style or musician. The data sample I used are chords from The Beatles. I code these algorithms in python and I use basic libraries to help with the numerical operations.

To simulate music I have to:

1. Take a large data sample of chords
2. Calculate the probability distribution.
3. Do a random walk on my Markov Chain model.

I will go through those steps in further details in these next parts.

3.1 Finding the probability distribution

Here I describe how I create the Markov chain model using the data that I found.

1. Get the sequence of chords. e.g. 'F', 'A7sus4', 'A7', 'Dm', 'Gm6'
2. Make bigrams of chords next to each other. ('F', "A7sus4"), ('A7sus4', 'A7'), ('A7', 'Dm'), ('Dm,'Gm6')
3. Pick a chord as the initial chord in a sequence, and find the bigrams from the initial chord to another chords. e.g. We pick F, ('F', "A7sus4"), ('F', 'A7'), ('F', 'Dm'), ('F,'Gm6')
4. Then calculate the frequency of each unique bigram to appear in the sequence.
5. Normalizing the result, we will get probabilities of each transition.
6. Then, from the initial chord then repeat the process with the nodes connected to the initial one.

Now from the probability distributions that I have build I have Markov chain that I can preform random walks upon to obtain musical chords. I can also have a transition probability matrix to use for statistical calculations.

3.2 Monte Carlo approach

Markov Chain Monte Carlo (MCMC) is a statistical method for random sampling. By constructing a Markov chain, I can sample from the distribution by recording states from the chain.The more the number of steps recorded, the closer the distribution of the sample matches the actual desired distribution. MCMC is has a lot of algorithms, such as the Metropolis–Hastings algorithm, and these algorithms can all be adjusted to achieve specific tasks. Furthermore, this method is more computationally inexpensive than other techniques to achieve the same tasks, sometimes not possible to do computationally. Sampling is important because most of the time it is not difficult or impossible to do. The probability distribution can be changed easier and sampling is unbiased and the variance can be changed.

Generally there are 4 steps that are most common in MCMC:

1. Finding or optimising a probability.
2. Repeatedly select random data points.
3. Perform a deterministic computation.
4. Combine the results.

Furthermore, using MCMC can provide more analytical data of my Markov chain. For this section, I will try to find the stationary distribution using this approach as the greater the number of steps I sample, the closer that distribution will be to the actual stationary distribution with its ratio between states (due to the Law of large numbers). It provides information about certain styles of music which can be used to compare with different artists or genres. This could provide useful information for music producers and musicians.

To do this I will:

1. Simulate a random walk with a very high number of steps. (n = 1000000)
2. Count the number of times each state is appearing in our walk.
3. Divide the frequency of each state by n and thus find the stationary distribution.

This method works if the number of steps is huge due to the law of large numbers as the answer converges to the expected value.

Alternative approaches:
•Matrix multiplication, where the transition matrix is calculated to the power of a very high number n.
•Linear algebra approach, by finding and normalising the left eigenvector when $\lambda = 1$, as seen before.

3.3 Problems

When creating the Markov chain, we may not reach an equilibrium because the Markov chain does not satisfy the conditions to do so, namely being aperiodic and irreducible. This could happen with some cases of data samples. The cause could be from a state that has no transition probabilities to it and a state that does not have transition probabilities to other states (an absorbing state) which means the Markov chain is not irreducible. To solve this issue, I am am going to apply 2 methods that are also used in Google's pagerank algorithm which is also a Markov chain but it ranks websites.

A page with no incoming links means a column of zeros in the transition matrix. This means that the state will no be reachable (irreducible) and the Markov chain will not converge. One solution is to introduce a damping factor. It is implemented by adding a constant uniformly to all states on every step. This will give each state a non-zero probability in the probability distribution and prevent states flip-flopping.

State with no outgoing links, is an absorbing state which again does not allow the Markov chain to be irreducible. This means there is a row of zeros in the transition matrix, also means the row does not sum to 1. The solution is to create equal transition probabilities with that state with every other state, meaning the probability to any state from that state is the same.

However, applying this onto our Markov chain model means that the probabilities are different and therefore do not represent the original model that well.

4 Results

Here are some results of doing random walks on my Markov chain model:

['Bb', 'Dm', 'C', 'Dm7', 'Bb', 'C7', 'F', 'F', 'A7sus4', 'A7', 'Dm', 'Gm6', 'C7', 'F', 'Em7', 'A7', 'Dm', 'C', 'Dm7', 'G7', 'Bb', 'C7', 'F', 'F', 'A7sus4', 'A7', 'Dm', 'C', 'Dm7', 'G7']

['Dm7', 'G7', 'Bb', 'C7', 'F', 'F', 'A7sus4', 'A7', 'Dm', 'C', 'Dm7', 'Bb', 'F', 'C', 'Bb', 'C7', 'F', 'Fsus4', 'F', 'C', 'Bb', 'F', 'Em7', 'A7', 'Dm', 'C', 'Dm7', 'G7', 'Bb', 'C7']

['Dm7', 'G7', 'Bb', 'F', 'G7', 'Bb', 'Dm', 'Gm6', 'C7', 'F', 'Em7', 'A7', 'Dm', 'C', 'Bb', 'F', 'C', 'Dm7', 'G7', 'Bb', 'F', 'A7sus4', 'A7', 'Dm', 'Gm6', 'C7', 'F', 'A7sus4', 'A7', 'Dm']

As you can see, each random walk generated creates a sequence of unique chords that are similar in style to the chord progressions from my data. It is difficult to show that these chords do resemble the original artist as I cannot play music out of this project but I have played them. Some parts of the sequences are incoherent to others but it is difficult to tell as music is subjective.

Here is the stationary distribution for my model with a million steps using the Monte Carlo approach:
('Dm7': 0.084990, 'Bb': 0.138257, 'Dm': 0.127625, 'Gm6': 0.042460, 'C7': 0.085074, 'F': 0.191686, 'Em7': 0.042606, 'A7': 0.085285, 'C': 0.084892, 'Fsus4': 0.021176, 'G7': 0.053270, 'A7sus4': 0.042679)

As you can see, we can see the probability distribution of the chords simulated by the Markov chain if it supposedly ran forever. From this, we can deduce that 'F','Bb' and 'Dm' are the most played chords and 'Fsus4', 'Gm6', and 'A7sus4' are the least played chords. There are some sub-sequences of the random walk that are the same to one another or to the original data which I think is expected but we can lower the likelihood of that by maybe adding more data. We can then apply the same method to different types of data with different artist or genres of music and compare the similarities and differences.

4.1 Comparisons

In this section, I will be discussing the advantages and disadvantages of using Markov chains to simulate music.

The advantages I found in using Markov chains is that, firstly, it is very computationally inexpensive to run. Compared to using a big software which uses large amounts of memory and requires a powerful computer to execute, simulating music with Markov chains was faster and did not affect my CPU as much. Depending on the data used to build a model, the model could replicate certain genres and musicians very well. I find that using a lot of data to build the model makes it so that it sound less random. There are many different machine learning methods and models, so it can be difficult to chose the right method for the intended task whereas

Markov chains are comparatively straightforward and can be manipulated easier. Finding the stationary distribution made it so that I could analyse the music and be able to compare it to other types of music.

The disadvantages are that, the larger the size of the data or the more states there are, the more time it takes for the Markov chain model to be constructed. Further, the effectiveness of the Markov chain depends on the data used to be good, so if I cannot find a good source of data then model itself with perform poorly. Furthermore, collecting and processing the data can take longer and may be more tedious. This is because there are many sources of data for music but they may be in a different format or contain unique symbols that are difficult to understand. Since I am using random walks, there are some cases where the music does sound random and incoherent with each other, hence there is not guarantee the music will make any sense. This project is based on music therefore for some readers, there is musical complexity as they cannot read or recreate the music. Furthermore, when finding data, some may be from a different key which is not cohesive and sometimes may need to be transposed to one specific key, this adds another layer of difficulty. Also, the model only generates chords, it does not generate the tonality, timbre and other musical qualities that would improve the model but also severely complicates it.

4.2 Alternative simulation methods

The theory of Markov chains are important because so many things can be modelled by it as they satisfy the Markov property, yet there are some common examples of stochastic properties that do not satisfy the Markov property. Ideas behind chaos theory would be modelled well by Markov chains because it is based on the idea of one small action can have a ripple effect and become something great.

Google's MagentaAI uses neural networks and machine learning methods that provide people the tools to manipulate music and art. The AI require greatly more computational power and is still an area of great research. There are a plethora of demos that provide people with interesting and creative tools for creating music. There is even a demo called "MidiMe" which trains a small model to sound like you. These sort of experiments use Tensorflow which is deserves its own dissertation to talk about its marvelous ingenuity. However, some machine learning models use Markov chains in itself.

Another simulation approach can be generating random variates. Simulation approaches in inference are randomisation and permutation procedures and bootstrapping.

5 Conclusion

In conclusion, I believe Markov chains are a powerful simulation technique for stochastic processes that satisfy the Markov property, especially for time-series data like music. Compared to other simulation and modelling techniques, it is not computationally expensive. Although music itself is subjective as it is an art, it was interesting to recreate and analyse. This project has provided me an understanding in how stochastic processes work and has given me a great appreciation for them because of their application and usefulness. It was interesting to analyse the data behind the music to try and further understand why some music sounds the way it does. If were to do this project again, I would experiment with higher order Markov chains and see if that is a better way to model music.

References

[1] Lothar Breuer (2014) Introduction to Stochastic Processes. University of Kent. Available at https://www.kent.ac.uk/smsas/personal/lb209/files/sp14.pdf (Accessed: 3 August 2020).

[2] JJ O'Connor and EF Robertson (2016) Andrei Andreyevich Markov. University of St. Andrews, Scotland. Available at https://mathshistory.st-andrews.ac.uk/Biographies/Markov/ (Accessed: 4 August 2020).

[3] Erhan Cinlar (2013) Introduction to Stochastic Processes [Dover books on Mathematics]. Dover Publications, Inc; Reprint edition (20 Feb. 2013). (Accessed 9 August 2020).

[4] Steven Thornton, (25 October, 2016) Markov Chains in LaTeX. Available at: http://steventhornton.ca/blog/Markov-chains-in-latex.html (Accessed: 14 October 2020).

[5] Joseph Rocca, (24 February, 2019 Introduction to Markov Chains. Available at: https://towardsdatascience.com/brief-introduction-to-Markov-chains-2c8cab9c98ab (Accessed: 13 February 2021).

[6] Brilliant, Markov Chains. Available at: https://brilliant.org/wiki/Markov-chains/see-also (Accessed: 4 March 2021).

[7] Dani Gamerman and Hedibert F. Lopez, (25 October, 2016) Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. Available at: http://steventhornton.ca/blog/Markov-chains-in-latex.html (Accessed: 5 March 2021).

[8] Paul K. Newton and Kamran Aslam, (2009) Journal of Quantitative Analysis in Sports. Available at: https://cpb-us-e1.wpmucdn.com/sites.usc.edu/dist/5/476/files/2020/04/JQA2009.pdf (Accessed: 5 April 2021).

[9] Brit Cruise, (28 April 2014) Origin of Markov Chains. Available at: https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/markovchains (Accessed: 7 April 2021).

[10] Richard Weber, (Autumn 2012) Markov Chains. Available at: http://www.statslab.cam.ac.uk/ rrw1/markov/index.html (Accessed: 9 April 2021).

[11] Victor Powell and Lewis Lehe, (2015) Markov Chains visually explained. Available at:https://setosa.io/ev/markov-chains/ (Accessed: 15 April 2021).

[12] John McCormack, (1996) Grammer Based Music Composition. Available at:http://users.monash.edu/ jonmc/research/Papers/L-systemsMusic.pdf (Accessed: 16 April 2021).