MARKOV CHAINS AND MUSICAL COMPOSITION

PARTH WOKHLU

Abstract. In this paper we will provide a brief overview of discrete Markov chains, particularly concerning their long-time behavior. We will primarily be following Greg Lawler's exposition *Introduction to Stochastic Processes*. Then, we will study applications of this theory to stochastic music composition.

Contents

1. Introduction

The intersection of mathematics and music composition, while not an obvious one, has a long history. Throughout the mid-to-late 1900s, experimental composers like John Cage and Iannis Xenakis played around with the idea of leaving the choice of music notes up to chance in the midst of performance rather than being written by the composer. In this paper, we will utilize Markov chains to analyze musical compositions and ultimately generate new ones. First, we will discuss the various properties and behaviors of Markov chains, and then apply this to using Markov chains to randomly generate musical compositions.

2. Finite Markov Chains

2.1. **Definitions.** In this paper, we will be studying Markov chains: a type of stochastic process. A stochastic process $\{X_n\}_{n\in\mathbb{N}}$ is a sequence of random variables, and for our purposes, these random variables can only assume a finite number of states.

Date: DEADLINES: Draft AUGUST 14 and Final version AUGUST 28, 2021.

First, let's establish the notation that we'll be utilizing throughout this paper. We can consider the finite set of states $S = \{0, 1, 2, \ldots, N\}$ where each $n \in S$ denotes a state that the stochastic process X_n can land on. The probability of this process being unfolded in a particular way can be denoted by: $\mathbb{P}\{X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n\}$, where each i_k denotes a different state such that $i_k \in S$.

Every stochastic process begins somewhere. We can define the initial probability distribution, or the probability of a particular state at time zero, by $\phi(k) = \mathbb{P}\{X_0 = i_k \mid k \in S\}$. The rest of the process is dictated by the transition probability, or the probability of the chain moving from one particular state to some other given state. This can be written as $p_n(i_n \mid i_0, \ldots, i_{n-1})$, which formally is equivalent to $\mathbb{P}\{X_n = i_n \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}\}$. Thus, we can rewrite the cumulative probability of the system unfolding as $\mathbb{P}\{X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n\} = \phi(i_0)p_1(i_1|i_0)p_2(i_2|i_0, i_1)\ldots p_n(i_n \mid i_0, \ldots, i_{n-1})$

The Markov property of Markov chains dictates that the transition behavior of the system only considers the current state and disregards the history of the chain. Mathematically, this can be written as $\mathbb{P}\{X_n = i_n \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}\} = \mathbb{P}\{X_n = i_n \mid X_{n-1} = i_{n-1}\}$. Thus, we can write our transition probabilities as $\mathbb{P}\{X_n = i_n \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}\} = p(i_{n-1}, i)$.

In order to analyze, simulate, and extrapolate data from a Markov chain, we represent it with a transition matrix. A transition matrix $\mathbf{P}$ for a Markov chain is an $(N+1)\times(N+1)$ matrix whose (i, j) entry is $p(i, j)$, or the probability of the system moving from state i to state j. A transition matrix must be a stochastic matrix, which has the following two properties. First, for all i, j such that $1 \leq i, j \leq N$, $0 \leq \mathbf{P}_{ij} \leq 1$, as all the entries of the matrix are transition probabilities. Second, $\sum_{j=1}^{N} \mathbf{P}_{ij} = 1$ where $1 \leq i \leq N$. The summation of transition probabilities across a row must equal 1 because from each state i, the Markov chain must transition to *some* state with probability 1.

2.2. **Large-time behavior.** We can use the transition matrix of a Markov chain to study the behavior of the chain over multiple time steps instead of just one. We can denote the n-step probability as $\mathbf{P}_n(i, j)$, which is the probability that the stochastic process will transition from state i to state j in n steps, or more formally, $\mathbb{P}\{X_{k+n} = j \mid X_k = i\}$.

We first show that there is a easy way to compute these probabilities with the transition matrix.

Theorem 2.1. *$\mathbf{P}_n(i, j)$ is the (i, j) entry in the matrix $\mathbf{P}^n$. In other words, $\mathbf{P}^n = \mathbf{P}_n$*

Proof. This can be proven by induction. Take $n = 1$ as a base case, which is trivially true. Then, we establish our inductive hypothesis that $\mathbf{P}^n = \mathbf{P}_n$ for any given n and we want to show that this holds true for $n + 1$.

Consider the following probability: $\mathbb{P}\{X_{n+1} = j \mid X_0 = i\}$ (the probability that given a starting state at i, the system will reach state j in $n + 1$ steps). Given the state set S, this probability is equivalent to $\sum_{k\in S} \mathbb{P}\{X_n = k \mid X_0 = i\}\mathbb{P}\{X_{n+1} = j \mid X_n = k\}$. Summing up the probabilities over the entirety of the state set reveals the probability that state i moves to any other state in n steps and moves to state j in 1 step.

Mathematically, this is equal to $\sum_{k\in S} p_n(i,k)p(k,j)$. Ultimately, because $p_n(i,k)$ is the (i,k) entry of the $\mathbf{P}^n$ matrix as per the induction hypothesis, the last summation over the state set is the (i,j) entry of $\mathbf{P}^n\mathbf{P} = \mathbf{P}^{n+1}$, proving the theorem by induction. □

In general, we can understand the large-time behavior of Markov chains by considering $\mathbf{P}^n$ for large values of n. This can be represented as the limit matrix $\prod = \lim_{n\to\infty} \mathbf{P}^n$. With this in mind, define the limiting probability vector $\overline{\pi}$ as the vector such that for any probability vector $\overline{v}$, $\lim_{n\to\infty} \overline{v}\mathbf{P}^n = \overline{\pi}$. This introduces the notion of an invariant probability distribution for $\mathbf{P}$, where there exists some limiting probability vector $\overline{v}$ such that $\overline{\pi} = \lim_{n\to\infty} \overline{v}\mathbf{P}^{n+1} = (\lim_{n\to\infty} \overline{v}\mathbf{P}^n)\mathbf{P} = \overline{\pi}\mathbf{P}$. This probability vector $\overline{\pi}$ is called the invariant probability distribution.

Also known as the stationary probability distribution, intuitively, the vector $\overline{\pi}$ represents the probability of being in any given state at any point in time. Importantly, it can also be described as the left eigenvector of $\mathbf{P}$ with eigenvalue 1.

2.3. **Reducibility.** In our analysis of Markov chains, it's interesting to observe how the events within the system interact with each other. The concept of "reducibility" deals with the idea of how different states within the state set are grouped together by their chances of transitioning between each other.

Definition 2.2. Two states i and j of a Markov Chain "communicate" (denoted by $i \leftrightarrow j$) if there exists some $m, n \geq 0$ such that $p_m(i,j) > 0$ and $p_n(j,i) > 0$.

In other words, i and j communicate if and only if there is a non-zero probability of being able to start at either i or j and reach the other state within a finite number of steps/transitions.

Lemma 2.1. *Communication is an equivalence relation.*

Proof. Reflexivity follows from the fact that $p_0(i,i) = 1$, which is a positive probability over a finite number of steps. Symmetry follows from the definition, as in order for i to communicate with j, j must communicate with i.

For transitivity, we want to show that if $i \leftrightarrow j$ and $j \leftrightarrow k$, then $i \leftrightarrow k$. We know that for some n and m, $p_n(i,j) > 0$ and $p_m(j,k) > 0$. Then, $p_{n+m}(i,k) = \mathbb{P}\{X_{n+m} = k \mid X_0 = i\}$. Because this describes a general path from state i to k in $n+m$ steps, we can say that it is greater than or equal to the following probability: $\mathbb{P}\{X_{n+m} = k, X_n = j \mid X_0 = i\} = \mathbb{P}\{X_n = j \mid X_0 = i\}\mathbb{P}\{X_{n+m} = k \mid X_n = j\} = p_n(i,j)p_m(j,k)$. By the assumption, we know that this product is non-zero, which means that the intial probability $p_{n+m}(i,k) = \mathbb{P}\{X_{n+m} = k \mid X_0 = i\} > 0$, and transitivity holds. Thus, because the communication relation is reflexive, symmetric, and transitive, it is an equivalence relation. □

We call the equivalence classes induced by this equivalence relation *equivalence classes*. Essentially, two states are in the same communication class if and only if they can communicate with each other. We call a Markov chain *irreducible* if there is only 1 communication class, or in other words, if every state within the Markov chain has a communication relation with another.

This also introduces the idea of different types of classes. Take the example of a Markov chain with a state set of 5 states, where states $2, 3, 4$ have a $\frac{1}{5}$ chance of transitioning to each of the 5 states, while $p(1,1) = p(0,0) = 1$. In this case, there

are three communication classes: $\{1\}, \{2, 3, 4\}$, and $\{5\}$. We can call the middle class $\{2, 3, 4\}$ a transient class because there is a certain probability that the system will leave the class for good and not return (because states 1 and 5 stay in the same state with probability 1). Thus, we can call states 1 and 5 recurrent states.

2.4. **Periodicity.** For the sake of definitions, suppose that $\mathbf{P}$ is the matrix for an irreducible Markov chain. We can define the period of some state $d = d(i)$ to be the greatest common divisor of the set $J_i = \{n \geq 0 \colon p_n(i, i) > 0\}$. In other words, the state i has the period k if any return to state i must occur in multiples of k time steps.

We call an irreducible matrix *aperiodic* if its period is equal to 1. This grouping of all the states is possible because of the following theorem.

Theorem 2.3. *All states within a communication class have the same period.*

Proof. We want to show that periodicity is a class property. Suppose that state i has a period d and state j has a period of d'. We want to show that $d = d'$. Because $\mathbf{P}$ is irreducible, we know that $i \leftrightarrow j$, so there exists some finite numbers m and n such that $p_m(i, j) > 0$ and $p_n(j, i) > 0$, so then $p_{m+n}(i, i) > 0$. Thus, $m + n$ is divisible by d. Now, suppose that $p_k(j, j) > 0$ for some k. Thus, we know that $p_m(i, j)p_k(j, j)p_n(j, i) > 0$. It follows that i can return to state i in $(m + k + n)$ steps, so $p_{m+k+n}(i, i) > 0$. Thus, we have that $(m + k + n)$ is divisible by d. We already know that $(m + n)$ is divisible by d. Thus, it must follow that k is divisible by d.

By the definition of periodicity, d' is the greatest possible divisor of k, so $d' \geq d$. Analogously, through the same logic while flipping the states, we can see that $d \geq d'$. Thus, $d' = d$. □

Now that we have established reducibility, periodicity, and the relationship between them, we can further extrapolate theorems from these properties.

Theorem 2.4. *If $\mathbf{P}$ is irreducible and aperiodic, then there exists an $M > 0$ such that for all $n \geq M$, $\mathbf{P}^n$ has all entries strictly positive.*

Proof. Take any two states i and j. Since $\mathbf{P}$ is irreducible, i and j can communicate with each other. Thus, there is some integer m such that $p_m(i, j) > 0$.

Since $\mathbf{P}$ is aperiodic, the greatest common divisor of the number of steps to bring the state i back to i is 1. Thus, there exists some integer L such that for all $n \geq L$, $p_n(i, i) > 0$.

It follows that for all $n \geq L$, $p_n(i, i)p_m(i, j) > 0$, which implies that $p_{n+m}(i, j) > 0$. Let the integer M be equal to $L + m$. Then, for all $n \geq M$, $p_n(i, j) > 0$.

Since i and j were arbitrary, this applies to every pairing of states within the finite state set. □

It is also interesting to observe the properties of $\mathbf{P}^n$ when $\mathbf{P}$ is reducible and can be broken down into communication classes. We can denote the recurrent classes of $\mathbf{P}$ as $R_1, \ldots, R_r$, and transient classes $T_1, \ldots, T_s$.

By definition, recurrent classes are self-contained, so each acts as a sub-chain of the whole Markov system, where the transition sub-matrix simply consists of the probabilities associated with moving within the recurrent class. As a result, each recurrent class has a different invariant probability vector associated with it such that if some state i is not in recurrent class R_k (where $1 \leq k \leq r$), then the class'

associated invariant probability distribution π^k has no chance of landing on i. In mathematical notation, $\pi^k(i) = 0$ if $i \notin R_k$.

Furthermore, assume that the submatrix P_k for each recurrent class R_k is aperiodic, such that each state returns to itself in an irregular/non-divisible number of steps. Then, if $i \in R_k$, $\lim_{n\to\infty} p_n(i,j) = \pi^k(j)$ when $j \in R_k$ and $\lim_{n\to\infty} p_n(i,j) = 0$ when $j \notin R_k$.

This is due to the following theorem concerning the calculation of invariant probability

Theorem 2.5. *π can be computed by* $\lim_{n\to\infty} p_n(i,j) = \pi(j)$

Proof. This proof is tedious and long, but can be found in a past REU author's paper under Theorem 4.5 [**3**]. □

It follows that if states i and j are members of transient classes, because a chain will always end in a recurrent class, $\lim_{n\to\infty} p_n(i,j) = 0$.

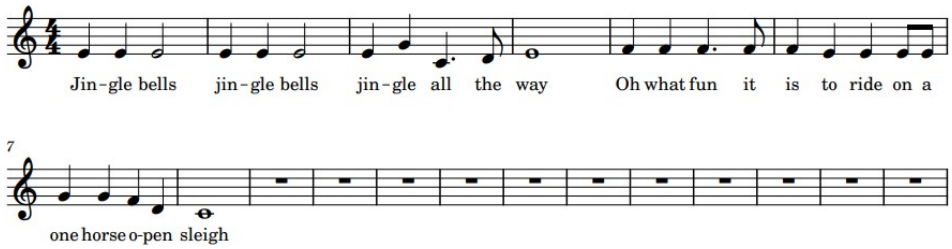
3. Music Generation

Now that we have established the concept and property of Markov chains, we can utilize them to create musical compositions. The goal of this section will be to create a stochastic matrix based on the collection of notes of "Jingle Bells" and return a musical score filled with notes generated by the matrix.

All of the referenced code and files can be found on Github:

3.1. **Matrix creation.**

First, we have to create a stochastic matrix that contained the data we needed to store from the following arrangement of "Jingle Bells".



In order to create our matrix, we need to find the transition probabilities between all of the "sound objects" in the score. Each sound object is a musical pitch and duration. For example, the first note in the piece is the sound object E4; it has a pitch of E and has the duration of a quarter note.

In total, there are 12 sound objects in the piece, so we must create a 12x12 matrix where entry (X, Y) is the probability of sound object X being immediately followed by sound object Y.

After calculating these transition probabilities by counting the number of individual transitions and dividing by 12, we have the following stochastic matrix which we can call **P**. We can confirm that this is a stochastic matrix because all of the rows add up to a value of 1.

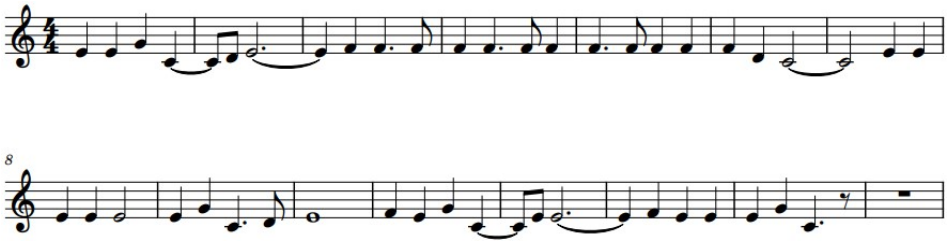
	E4	E2	G4	C3	D8	E	F4	F3	F8	E8	D4	C
E4	0.43	0.29	0.14	0	0	0	0	0	0	0.14	0	0
E2	1	0	0	0	0	0	0	0	0	0	0	0
G4	0	0	0.5	0.5	0	0	0	0	0	0	0	0
C3	0	0	0	0	1	0	0	0	0	0	0	0
D8	0	0	0	0	0	1	0	0	0	0	0	0
E	0	0	0	0	0	0	1	0	0	0	0	0
F4	0.25	0	0	0	0	0	0.25	0.25	0	0	0.25	0
F3	0	0	0	0	0	0	0	0	1	0	0	0
F8	0	0	0	0	0	0	1	0	0	0	0	0
E8	0	0	0.5	0	0	0	0	0	0	0.5	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	1
C	1	0	0	0	0	0	0	0	0	0	0	0

3.2. **Markov chain simulation.** Now that we have our stochastic matrix **P**, we can use run through it to generate new music with the same underlying transition properties as Jingle Bells. The code referenced in this section can be found under the file *mcSimulation.py*

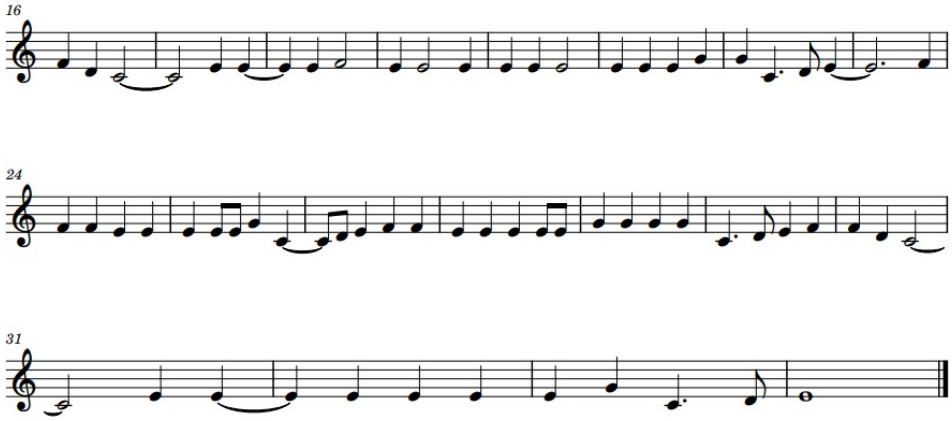
We run through the chain utilizing the transition probabilities to guide our choices in sound objects. Given that most of our sound objects are quarter notes, if we go through 48 steps of our transition matrix, we should be able to generate about two lines of music. We can also start with two different initial states to observe the impact of the initial probability distribution: one chain where we start with the same state as the original training data of "Jingle Bells" (E4), and one chain where there is an even chance of starting in 3 of the primary sound objects in the C Major key: E4, G4, and F4. From an initial sampled point, we then generate subsequent sound objects as dictated by the transition matrix.

We eventually get the following two series of sound objects (one for each aforementioned initial probability distribution).

3.3. **Results.** The following musical score is the first result, with an initial probability distribution of E4:



The next score is the second result which starts on an F note rather than an E note.



Audio recordings are available here.

Audibly and visually, we can see how these songs have a similar flavor to Jingle Bells.

3.4. **Invariant probability distribution calculation.** Now that we have generated a musical score from our stochastic matrix, we can take our mathematical analysis a step further by finding the invariant probability distribution (the probability of being in any given state at any point in time) of the sound objects of "Jingle Bells" utilizing three different methods. Each of the three referenced algorithms can be found in the file *invarProbDistrib.py*

First, we can leverage our knowledge of large-time behavior from Section 2.2, where it was established that multiplying $\mathbf{P}$ an extremely large amount of times ($\mathbf{P}^n$ where n approaches infinity) reveals the invariant probability distribution. In this case, we multiply $\mathbf{P}$ by itself 1000 times. Using this method, we get $\pi = [0.2414$ 0.069 0.1379 0.069 0.069 0.069 0.1379 0.0345 0.0345 0.069 0.0345 $0.0345]$

By the same intuition, we can use a Monte Carlo method, where we repeat our simulation for an extremely large number of steps, count the number of times each state was landed on, and divide our ending result by the number of steps we took to get there.

This returns the following probability distribution: $\pi = [0.2406$ 0.0686 0.1382 0.0692 0.0692 0.0692 0.1381 0.0344 0.0344 0.0691 0.0344 $0.0344]$, which is roughly the same as our previous result.

The final method we can utilize to find the invariant probability distribution is find the left eigenvectors of $\mathbf{P}$. As established in Section 2.2, the left eigenvector corresponding to the eigenvalue of 1 returns the invariant probability distribution. Using Python to calculate this vector gives us the following: $\pi = [0.2414,$ $0.069,$ $0.1379,$ $0.069,$ $0.069,$ $0.069,$ $0.1379,$ $0.0344,$ $0.0344,$ $0.069,$ $0.0344,$ $0.0344]$.

All three methods remain consistent, corroborating each other's merit.

4. Bibliography

I would like to thank my mentor Phillip Lo for encouraging my research in this topic and helping me throughout the process. I would also like to thank Greg Lawler for his accessible and in-depth writing about stochastic processes and Markov Chains. I'd also like to thank Sujan Dutta for his work on YouTube with

helping me bring my math to Python. Finally, I'd like to thank Professor Peter May for helping me find funding and organizing my amazing experience at the REU.

References

[1] Lawler, G.F.(1995). Introduction to Stochastic Processes. USA: Chapman Hall
[2] Datta, Sujan. Markov Chains: Simulation in Python — Stationary Distribution Computation — Part - 7. Youtube, 31 Mar. 2021, https://www.youtube.com/watch?v=G7FIQ9fXl6U.
[3] Wu, Botao. Invariant Probability Distributions. UChicago Mathematics REU, http://www.math.uchicago.edu/ may/VIGRE/VIGRE2011/REUPapers/WuB.pdf. Accessed 29 July 2011.