GP Assignment - Markov Analysis and Algorithmic Composition

Abstract

This Assignment is to practice Markov Analysis and Algorithmic Composition with Beethoven's Ode to Joy and Turk's March. Also contain discussion questions/short essay.

Beethoven Task

Working by analogy, do for the Beethoven fragment what was modeled in the classroom activity that featured the Kabalevsky melody. That is, for the Beethoven fragment:
1. Create the state transition count matrix – being sure to consider the melody to be a "wrap around" melody. Do your best to render an accurate, professional looking count matrix.

from/to	**C**	**CH**	**CI**	**D**	**DH**	**DI**	**DQ**	**E**	**EQ**	**F**	**G**
C	2			2							
CH								1			
CI		1									
D	2							2			
DH								1			
DI					1						
DQ			1								
E				2			1	2	1	2	
EQ						1					
F								2			2
G										2	2

E E F G G F E D C C D E EQ. DI DH E E F G G F E D C C D E DQ. CI CH -> E

2. Create the state transition probability matrix. Do your best to render an accurate, professional looking probability matrix.

from/to	**C**	**CH**	**CI**	**D**	**DH**	**DI**	**DQ**	**E**	**EQ**	**F**	**G**
C	0.5	0.00	0.00	0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.0	0.00	0.00	0.00
CI	0.00	1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
D	0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.5	0.00	0.00	0.00
DH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.0	0.00	0.00	0.00
DI	0.00	0.00	0.00	0.00	1.0	0.00	0.00	0.00	0.00	0.00	0.00
DQ	0.00	0.00	1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E	0.00	0.00	0.00	0.25	0.00	0.00	0.125	0.25	0.125	0.25	0.00
EQ	0.00	0.00	0.00	0.00	0.00	1.0	0.00	0.00	0.00	0.00	0.00
F	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.5	0.00	0.00	0.5
G	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.5	0.5

3. Create the state transition distribution matrix. Do your best to render an accurate, professional looking distribution matrix.

from/to	**C**	**CH**	**CI**	**D**	**DH**	**DI**	**DQ**	**E**	**EQ**	**F**	**G**
C	0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
CH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
CI	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
D	0.50	0.50	0.50	0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00
DH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
DI	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DQ	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
E	0.00	0.00	0.00	0.25	0.25	0.25	0.375	0.625	0.75	1.00	1.00
EQ	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
F	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.50	1.00
G	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	1.00

4. Manually run the simulation algorithm presented in class on the following sequence of random numbers in order to obtain a melodic fragment in the spirit of Ode to Joy. (Like the real Ode, your fragment will contain 30 notes. Unlike the real Ode, your result won't sound magical. Still it should sound like a sickly ghost of the real thing.

0.571 0.790 0.693 0.359 0.009 0.252 0.134 0.839 0.846 0.863
0.396 0.213 0.540 0.976 0.351 0.619 0.227 0.798 0.595 0.438
0.278 0.235 0.138 0.212 0.202 0.309 0.689 0.829 0.060 0.725

E E F G F E DQ CI CH E F
E D E F E E D E E E
DQ CI CH E D C D E D E

Turk Task

Working by analogy, do for Turk's march what was modeled in the classroom activity that featured the Kabalevsky melody. That is, for the Turk fragment:

1. Create the state transition count matrix – being sure to consider the melody to be a "wrap around" melody. Do your best to render an accurate, professional looking count matrix.

from/to	**C**	**CH**	**D**	**DH**	**E**	**EH**	**FH**	**G**
C	2		1					1
CH	1	1	1					
D		1	5					2
DH					1			
E	1		1		3			
EH		1						
FH				1				
G					1	1	1	

C C C G EH CH D D D G FH DH E E E C D D D G E E D D CH CH

2. Create the state transition probability matrix. Do your best to render an accurate, professional looking probability matrix.

from/to	**C**	**CH**	**D**	**DH**	**E**	**EH**	**FH**	**G**
C	0.500	0.000	0.250	0.000	0.000	0.000	0.000	0.250
CH	0.333	0.000	0.666	0.000	0.000	0.000	0.000	0.000
D	0.000	0.125	0.625	0.000	0.000	0.000	0.000	0.250
DH	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
E	0.200	0.000	0.200	0.000	0.600	0.000	0.000	0.000
EH	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
FH	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
G	0.000	0.000	0.000	0.000	0.333	0.333	0.333	0.000

3. Create the state transition distribution matrix. Do your best to render an accurate, professional looking distribution matrix.

from/to	**C**	**CH**	**D**	**DH**	**E**	**EH**	**FH**	**G**
C	0.500	0.500	0.750	0.750	0.750	0.750	0.750	1.000
CH	0.333	0.333	1.000	1.000	1.000	1.000	1.000	1.000
D	0.000	0.125	0.750	0.750	0.750	0.750	0.750	1.000
DH	0.000	0.000	0.000	0.000	1.000	1.000	1.000	1.000
E	0.200	0.200	0.400	0.400	1.000	1.000	1.000	1.000
EH	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
FH	0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000
G	0.000	0.000	0.000	0.000	0.333	0.666	1.000	1.000

4. Manually run the simulation algorithm presented in class on the following sequence of random numbers in order to obtain a melodic fragment in the spirit of March. (Like the real March, your fragment will contain 26 notes. Unlike the real March, your result won't sound completely coherent. Still it should sound like a sickly ghost of the real thing.
0.033 0.932 0.257 0.925 0.796 0.267 0.401 0.721 0.604 0.430
0.587 0.334 0.418 0.229 0.286 0.438 0.569 0.606 0.721 0.528
0.763 0.853 0.383 0.708 0.081 0.565

C C G E E E D D D D D
D D D D D D D D D D
G FH DH E C D

Short Essay

In this assignment, we focused on two melodies; Beethoven's Ode to Joy, and Turk's March. Each melody had their own distinct differences, for example in Beethoven's piece, the listener might notice the changes between each note to be more gradual, smooth but also variant, while in Turk's piece, there are longer same notes with sudden and greater note changes that are consistent throughout the whole thing. Another distinct difference between the two pieces is that in Ode to Joy, you can imagine yourself walking up and down several hills, while in March, it seems like you are, well, marching.

We also generated two of our own melodies by using the Markov process. A Markov process is a stochastic process that possesses the "Markov property", which means that one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history. The Markov process is useful for being able to capture the essence and behavior of something by using the probability of something happening after one another. So in order to create a Markov model, I started with filling out a state transition count matrix, which was done by counting the number of times a state (music note) transitioned to another state. Then I created a state transition probability matrix by using the data from the previous count matrix, and this probability matrix allowed us to see how probable it is to transition from one state to another. I believe that the state transition probability matrix does in some sense capture the spirit of both Ode to Joy and March because it gives us a good idea of how likely each state to another state will happen. So in the probability matrix, there are a few 1, which means there is only one possible transition from that state to another, and some part of this is true because in the real one, there are certain notes that follow one another more likely than others. Next, using the probability matrix, I created a state transition distribution matrix, which will be used to generate our "fake" melody. With the given number that was obtained from a simulation algorithm, we looked over the distribution matrix and repeated this until all the numbers were used. Then when we have our full melody, we type it into the SimplePlayer program that generates a midi file out of it, which can later be converted to mp3 file through a converter online.

This also proves that JFugue is an executable music knowledge representation since we did it in this assignment by generating our JFugue and then inputting it in SimplePlayer to get a midi file and converting it to mp3 for everyone to hear. JFugue is a knowledge representation because it's presenting music in strings. And this overall assignment allowed us to improvise our own version of the melodies. An improvisation is creating or inputting something into an original piece to make it unique while also maintaining some of the spirit of the original piece. The approach we did to generate the new melodies can be viewed as an improvisational process because it's creating something unique but at the same time it also has some of the traits of the original piece.