Random Music Generator*

Frank Chen

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 2.0†

Abstract

Random Music Generator

1 Random Music Generator

1.1 Introduction

Our original idea was to be able to generate random music once the instrument sounds were synthesized. Unfortunately, due to time constraints, the best we could do was to have a program randomly pick chords and note durations from an input library. The resulting output was, as expected, random chords and notes that most people would probably not consider as music. How does one teach a program how to write music? One technique is the Markov chain.

1.2 Markov Chains

Teaching a program music theory so that it can create music would be an extremely tedious task. You would have to teach chord structure, common progressions, the different musical styles, and so on. What if you could give the program examples of pieces you considered to be music and ask it, "write something like that for me." This is essentially how our Markov chain would work. The principle behind Markov chains in music is to generate a probability table to determine what note should come next. By feeding the program an example piece of music, the program can analyze the piece and create a probability table to determine which notes are more likely follow a given note. Below is an example of a first order Markov chain.

*Version 1.2: Dec 18, 2008 2:26 am +0000
†http://creativecommons.org/licenses/by/2.0/

Example of First Order Markov Chain Table for Music

	A	A#	B	C	D	E	F	G	G#
A	4/19		3/19		2/19	1/19		6/19	3/19
A#	1								
B	7/15		1/15	4/15		3/15			
C			6/15	3/15	6/15				
D				3/11	3/11	5/11			
E	4/19	1/19		3/19		5/19	4/19	1/19	1/19
F			1/5			1/5		3/5	
G	1/5		1/5	2/5				1/5	
G#			3/4			1/4			

Figure 1: The entries in each box indicate the probability or likelihood of the note in the column label to follow the note in the row label. Blank boxes indicate 0 probability, meaning that note will never be the next note given your current note. For example, if the current note is A#, the next note chosen will always be A.

With the probability table, one can generate random notes that still has some musical structure to it. By constructing a similar table for beats or note durations, one can complete the first order Markov chain for music generation. Increasing to a second order Markov chain simply means constructing a larger probability table where the row headings are now pairs of notes. The program will choose the next note according to the probability table, consequently updating the current note pair to include the newest note. The idea behind determining what order Markov chain to use is a balance between ensuring that the program has enough musical structure and allowing it enough freedom for creativity.