

Mean Beethoven

Dirk Schlingmann

Math/CS Division, University of South Carolina Upstate; schlingm@uscupstate.edu

Abstract

Can we extend the concept of the statistical mean to music? Can we find the mean of all 32 Beethoven piano sonatas? In this paper, I will investigate and provide possible answers to these questions.

Introduction

Computers, the Internet, and technology have advanced the way we learn and create. Playing sounds, displaying graphics, viewing moving images, and programming solutions have enhanced conceptual understanding and creative endeavors. With computing power, we can tackle real data and real problems. In this paper, I try to show how algorithmic music programming can provide musical solutions that are based on mathematical concepts and that expand our ideas of music.

Algorithmic Music

Music is an artistic composition of sounds. We often use instruments to create musical sounds, which are represented by notes in a musical score. If we want different instruments to collaborate on a piece of music, the instruments need to be designed in such a way that they produce sounds that are pleasing together. During the evolution of music, humans have recognized that certain ratios of frequencies of tones (pitches) produce favorable consonant sounds (chords). If the pitches (frequencies) adhere to a rigid common musical tone or tuning system, a whole orchestra with many different instruments and many different notes interacting with each other can play beautiful music together. Such a system is the 12-tone equal temperament, which is widely used in the Western world of today. With the introduction of computers and synthesizers, a Musical Instrument Digital Interface (MIDI) was engineered for this 12-tone equal temperament. MIDI has enhanced music and facilitated communication between MIDI devices. For a more detailed introduction to MIDI, please consult the information at reference [2].

In terms of mathematics, a music score is a list of notes, which are the instructions on how to perform the musical piece. A note is determined by its time-measure (the time when the note is turned on and the time when it is turned off), its pitch (frequency), the instrument (timbre), and the volume (dynamics or loudness). Each music note in a song can be described using the following notation or data structure: note = {{time-on, time-off}, pitch, instrument, volume}. A song is the set {note[1], note[2], ..., note[n]} of finitely many notes where note[i] = {{time-on[i], time-off[i]}, pitch[i], instrument[i], volume[i]}. This set of notes is stored in a MIDI file.

```
song = {
    {{time-on[1], time-off[1]}, pitch[1], instrument[1], volume[1]},
    ...,
    {{time-on[n], time-off[n]}, pitch[n], instrument[n], volume[n]}
}
```

In mathematics, we prefer numbers instead of music notation. For example, a standard grand piano has 88 keys (A0 through C8 in music notation) with corresponding MIDI numbers 21 through 108. Please note that the piano has the MIDI number 1, and the volume is a number between 0.0 and 1.0, where 0.0 is silence and 1.0 is maximum volume. For a more detailed list of MIDI instruments and their MIDI numbers,

please consult reference [3]. Using MIDI numbers, here are the first two and last two notes of the 1637 notes of Beethoven's *Adagio* from his famous *Pathetique* sonata.

```
adagio =
{
  {{0.0, 1.07083}, 56, 1, 0.27451}, {{0.0, 2.14375}, 44, 1, 0.321569},
  ...,
  {{292.038, 297.494}, 48, 1, 0.196078}, {{292.038, 297.494}, 56, 1, 0.196078}
}.
```

If we look more closely at the above MIDI data, we realize that a computer and a computer program might be helpful or even much more efficient in creating a set of individual notes and its associated music. The parameters for pitch, time-on, time-off, instrument, and volume can originate from mathematical objects, abstract concepts, or any appropriate data coming from almost anything imaginable. A computer algorithm (a carefully written computer program) can produce music that goes beyond traditional music. Music created by such computer algorithms is called algorithmic music. Algorithmic music is not limited to the MIDI environment. We can program our own unique sounds of any frequency and any timbre. Algorithmic music opens a door into a new world of music (references [4] and [5]).

Mean Beethoven

Mathematics, computing, and algorithmic music can also assist us in answering questions like “What would a mean Beethoven piano sonata sound like?” In terms of mathematics, the mean calculates the average. It might be easy to calculate the average number of a finite list of real numbers. We add up the individual real numbers and then divide the sum by the length of the list. The mean of the n real numbers x_1, x_2, \dots, x_n is $(x_1+x_2+\dots+x_n)/n$. Creating a mean Beethoven piano sonata is not that simple. Here are my attempts.

Beethoven wrote 32 piano sonatas. A complete list of MIDI files for all 32 piano sonatas can be found at *Kunst der Fuge* (reference [1]). Let us assume that all sonatas start at time 0.0. First, we organize the data of all individual notes from all piano sonatas. Let us assume that the notes of each individual piano sonata have already been put into a rigid order following these two rules. Notes with an earlier time-on are listed before notes with a later time-on. If notes share the same time-on value, then notes with a lower pitch are listed before notes with a higher pitch. Next, we create a spreadsheet-like layout of rows and columns where each row represents all the ordered MIDI notes of a piano sonata. If we introduce the notation, $\text{note}[i, j] = \{\text{t-on}[i, j], \text{t-off}[i, j], \text{p}[i, j], \text{instr}[i, j], \text{v}[i, j]\}$ to determine the j -th note of piano sonata i , then the described spreadsheet would look like the highlighted area of Table 1.

Before we calculate the j -th note ($\text{note_meanB}[j]$) of Mean Beethoven, we need to be aware that different piano sonatas have different amounts of notes. For example, piano sonata 1 has 7331 notes. Piano sonata 2 has 9340 notes, and so on. Piano sonata 20 has 2937 notes, which is the least amount of notes of all 32 piano sonatas. Piano sonata 19 contains 3362 notes, which is the second smallest amount. Piano sonata 23 contains 15575 notes, which is the second highest amount; and piano sonata 29 has 21579 notes, which is the highest amount of all 32 piano sonatas. When we compute $\text{note_meanB}[j]$, we must avoid adding data items that do not exist.

Now, let us produce a version of Mean Beethoven. The j -th note of Mean Beethoven can be calculated using the usual data structure

$$\text{note_meanB}[j] = \{\text{t-on_meanB}[j], \text{t-off_meanB}[j]\}, \text{p_mean}[j], \text{instr_meanB}[j], \text{v_meanB}[j]\}.$$

If $1 \leq j \leq 2937$, then

$$\text{t-on_meanB}[j] = (\text{t-on}[1, j] + \dots + \text{t-on}[32, j])/32, \text{t-off_meanB}[j] = (\text{t-off}[1, j] + \dots + \text{t-off}[32, j])/32,$$

$$\text{p_meanB}[j] = \text{Round}[(\text{p}[1, j] + \dots + \text{p}[32, j])/32], \text{instr_meanB}[j] = 1, \text{ and}$$

$$\text{v_meanB}[j] = (\text{v}[1, j] + \dots + \text{v}[32, j])/32.$$

If $2937 < j \leq 3362$, then

$$\begin{aligned}
 t\text{-on_meanB}[j] &= (t\text{-on}[1, j] + \dots + t\text{-on}[19, j] + t\text{-on}[21, j] + \dots + t\text{-on}[32, j])/31, \\
 t\text{-off_meanB}[j] &= (t\text{-off}[1, j] + \dots + t\text{-off}[19, j] + t\text{-off}[21, j] + \dots + t\text{-off}[32, j])/31, \\
 p\text{-meanB}[j] &= \text{Round}[(p[1, j] + \dots + p[19, j] + p[21, j] + \dots + p[32, j])/31], \\
 \text{instr_meanB}[j] &= 1, \text{ and} \\
 v\text{-meanB}[j] &= (v[1, j] + \dots + v[19, j] + v[21, j] + \dots + v[32, j])/31.
 \end{aligned}$$

If $j > 3362$ notes, then we continue in the above fashion. The bottom row of Table 1 shows $\text{note_meanB}[1]$ and $\text{note_meanB}[2]$.

Table 1: Organization of the Notes of Beethoven’s 32 Piano Sonatas with Mean

	1 st note	2 nd note	...
sonata 1	{{0.0, 0.165593}, 60, 1, 0.266667}	{{0.256417, 0.420606, 65, 1, 0.282353}	...
sonata 2	{{0.0, 0.110294}, 57, 1, 0.470588}	{0.0, 0.110294}, 69, 1, 0.470588}	...
...
sonata i	{{t-on[i, 1], t-off[i, 1], p[i, 1], 1, v[i, 1]}	{{t-on[i, 2], t-off[i, 2], p[i, 2], 1, v[i, 2]}	...
...
sonata 32	{{0.0, 0.184606}, 51, 1, 0.737255}	{0.0, 0.184606}, 63, 1, 0.737255}	...
mean	{{0.0, 0.686135}, 55, 1, 0.426471}	{{0.0806185, 0.854284}, 55, 1, 0.411275}	...

We will end up with the following excerpt (first two notes and last two notes) of Mean Beethoven, where the last note is $\text{note_meanB}[21579]$.

$$\begin{aligned}
 \text{Mean Beethoven} &= \\
 &\{ \\
 &\quad \{\{0.0, 0.686135\}, 55, 1, 0.426471\}, \{\{0.0806185, 0.854284\}, 55, 1, 0.411275\}, \dots, \\
 &\quad \{\{2302.65, 2311.33\}, 77, 1, 0.92549\}, \{\{2302.65, 2311.33\}, 82, 1, 0.941176\} \\
 &\}.
 \end{aligned}$$

When we listen to Mean Beethoven (supplemental MIDI-file MeanBeethoven.mid), it does not sound grandiose. It did not turn out—what some of you intuitively were hoping for—to be another beautiful piano sonata similar to the 32 existing piano sonatas. However, in a mathematical sense, we should have expected a monotonous song, at least at the beginning of the song. Because the mean calculates the center of a data set, the mean pitches hover around the center. Of course, at the end of Mean Beethoven, from the 15576th note through the 21579th note, Mean Beethoven sounds like the end of piano sonata 29.

Mode Beethoven

In mathematics, the modes of a list of numbers are the numbers that occur most frequently in the list and that occur at least twice. If each number in the list of numbers occurs only once, then the mode does not exist. The mode can also be used as a measurement of the central tendencies of a set of numbers. In this paper, let us slightly modify the correct mathematical definition of the mode to also allow the mode to be each number in a list of numbers if each number occurs only once. What happens if we use this modified definition of the mode instead of the mean in the creation of a central Beethoven sonata, which we call Mode Beethoven? If there is more than one mode, then we take the average of the modes. We create Mode Beethoven by replacing $\text{note_meanB}[j]$ with

$$\text{note_modeB}[j] = \{\{t\text{-on_modeB}[j], t\text{-off_modeB}[j]\}, p\text{-mode}[j], \text{instr_modeB}[j], v\text{-modeB}[j]\},$$

where

$$\begin{aligned}t\text{-on_modeB}[j] &= \text{Mean}[\{\text{Mode}[\{t\text{-on}[1, j], \dots, t\text{-on}[32, j]\}]\}], \\ \text{duration_modeB}[j] &= \text{Mean}[\{\text{Mode}[\{t\text{-off}[1, j] - t\text{-on}[1, j], \dots, t\text{-off}[32, j] - t\text{-on}[32, j]\}]\}], \\ t\text{-off_modeB}[j] &= t\text{-on_modeB}[j] + \text{duration_modeB}[j], \\ p\text{-modeB}[j] &= \text{Round}[\text{Mean}[\{\text{Mode}[\{p[1, j], \dots, p[32, j]\}]\}], \\ \text{instr_modeB}[j] &= 1, \text{ and} \\ v\text{-modeB}[j] &= \text{Mean}[\{\text{Mode}[\{v[1, j], \dots, v[32, j]\}]\}]\end{aligned}$$

if $1 \leq j \leq 2937$, for example.

When I listen to Mode Beethoven (supplemental MIDI-file ModeBeethoven.mid), it already sounds more appealing to me in comparison to Mean Beethoven. Using the modified mode adds more diversity.

We even add more diversity if in the formulas for note_modeB[j], we randomly pick (abbreviated by RP) a mode if there is more than one mode instead of calculating the mean of the modes. We create Random Mode Beethoven by replacing note_modeB[j] with

$$\text{note_rmodeB}[j] = \{\{t\text{-on_rmodeB}[j], t\text{-off_rmodeB}[j]\}, p\text{-rmode}[j], \text{instr_rmodeB}[j], v\text{-rmodeB}[j]\},$$

where

$$\begin{aligned}t\text{-on_rmodeB}[j] &= \text{RP}[\{\text{Mode}[\{t\text{-on}[1, j], \dots, t\text{-on}[32, j]\}]\}], \\ \text{duration_rmodeB}[j] &= \text{RP}[\{\text{Mode}[\{t\text{-off}[1, j] - t\text{-on}[1, j], \dots, t\text{-off}[32, j] - t\text{-on}[32, j]\}]\}], \\ t\text{-off_rmodeB}[j] &= t\text{-on_rmodeB}[j] + \text{duration_rmodeB}[j], \\ p\text{-rmodeB}[j] &= \text{RP}[\{\text{Mode}[\{p[1, j], \dots, p[32, j]\}]\}], \\ \text{instr_rmodeB}[j] &= 1, \text{ and} \\ v\text{-rmodeB}[j] &= \text{RP}[\{\text{Mode}[\{v[1, j], \dots, v[32, j]\}]\}]\end{aligned}$$

if $1 \leq j \leq 2937$, for example.

When I listen to Random Mode Beethoven (supplemental MIDI-file RandomModeBeethoven.mid), it sounds even more appealing to me than Mode Beethoven.

If we transpose all sonatas into the same key, we might further enhance the appeal. You be the judge. For Random Mode Key C Beethoven (supplemental MIDI-file RandomModeKeyCBeethoven.mid), I have transposed all sonatas into the key of C major (A minor).

Summary and Conclusions

The algorithmic music Mean Beethoven, Mode Beethoven, Random Mode Beethoven, and Random Mode Key C Beethoven are my first attempts at finding an average Beethoven sonata. For me, the songs increase in attractiveness when I switch from Mean to Mode to Random Mode to Random Mode Key C. I believe this music has its place in the world of applications of mathematical ideas.

References

- [1] *Kunst der Fuge*. <https://www.kunsterfuge.com/>.
- [2] MIDI Association. <https://www.midi.org/>.
- [3] MIDI Association. General MIDI 1 Sound Set. <https://www.midi.org/specifications-old/item/gm-level-1-sound-set>.
- [4] D. Schlingmann. Algorithmic music albums. <https://open.spotify.com/search/Dirk%20Schlingmann/albums>.
- [5] D. Schlingmann. *Music via Math*. <https://www.amazon.com/Music-via-Math-Dirk-Schlingmann-ebook/dp/B07VTR4DKX/>.