Music and Mathematics Harmonizing Together
Cristian Cardona
Fall 2016 STEC 4500 Project

Abstract

This research was carried out with the intent to bridge a gap between music composition and how songs relate to one another in terms of their note sequence. As we know, musical preferences are quite varied, and sometimes we tend to like a certain artist over another and sometimes there seem to be certain patterns underneath our favoritism. By using music processes, we try to quantify this relationships between songs and artist with hopes of obtaining a relationship between them. As such, mathematicians look for tools to describe the artistic creativity and talent in this music we enjoy. In this project, Markov chains are used to compare and contrast songs from different artists. Given a set of states(in this project, notes will represent the states). A Markov chain will represent the the song or sequence of notes which make up the songs melody or most popular and iconic part of the song.

Introduction

The experimentation of random music has been around for a long time. One good example is one of the first formal types of algorithms in music history is the Musical Dice Game, or *Musikalisches Wurfelspiel*. The idea behind this game involved composing a series of pieces of music which were still pleasing to the ear, whom are going to be recombined in many different ways [1]. The dice come into play when you roll the dice and the result will be the number of composition which be used, this process gets carried out until there are no more pieces of music left to use or the composer has reached a composition that is suitable for his/her needs.

In the Virtual Music book by David Cope, he talks about many different programs which have been created to make random music in the style of popular composers like Bach, Mozart and Beethoven. This programs have been around for along time and do not provide explanations or insights as to what this similarities mean. David Cope himself was able to create some algorithms which take music as input and by the use of Generic algorithms

and what seems to be some sort of Artificial Intelligence, his algorithms learn the style of the composer being inputted into the algorithm. The more music the algorithm could analyze, the more it would learn about styles and were the patters being used by given artist. This algorithm took about 30 years to finalize.

The main purpose of this project is to understand the randomness of music and to make an attempt at creating a process by which one can compare two pieces of music and obtain information that informs us whether this two pieces of music are similar to each other and what does this similarity mean. The following are the research questions which directed the project and lead to some conclusions about the similarity of music. First question asks if whether an artist have a similar Markov chain across all of his/her songs. Second, we thought it would be important to know the differences and similarities of the Markov chain between two separate artists. Lastly, we would like to know if it is possible to distinguish a randomized song produced by the Markov chain of the original song itself.

Methodologies Used

In order to analyze the songs in a way that was organized and that would be reasonably easy, the songs were organized in matrices of $12x12$, in which both rows and columns are denoted by the 12 note system, in this case we used the notes: A, A♯, B, C, C♯, D, D♯, E, F, F♯, G, G♯. Since the songs were being organized into matrices, the next idea was to find the difference between two matrices, this would then tell us how far apart two songs are. In order for this to happen, both songs need to be in a matrix of the same size. This is already accounted for since all the songs, as previously mentioned were set to be $12x12$. The math that makes this is known by the name of Linear Algebra which deals with computations related to matrices.

The program used for computing and storing the data (songs as matrices) was $\mathbb{R}$. This program is composed of a language and environment for statistical computing and graphics. $\mathbb{R}$ provides a wide variety of statistical(linear and nonlinear modeling, classical statistics test, ...) and graphical techniques, and it is highly extensible. [2]

$\mathbb{R}$ comes with many built-in functions which were of some use for us to obtain results and plot them in order to provide visual representation. One main function used in order to obtain the distance between two matrices was

the $dist(x, method = "", ...)$ function which would give you the distance for the given argument by using the specified method. What we did exactly was hard code the Euclidean method and the Manhattan/City Block Distance as well as using the $dist(x, ...)$ function using the following methods: Minkowski, Euclidean, Manhattan/City Block Distance, Maximum, Canberra.

Here is an explanation of the most important methods used, as they help give understanding why they were chosen as a driving force as the mathematics behind this project. The Euclidean distance between two points p and q is the length of the line segment connecting them. In Cartesian coordinates, if $p = (p1, p2, ..., pn)$ and $q = (q1, q2, ..., qn)$ are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by the Pythagorean formula [3]:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_n - p_n)^2}$$

Next we have the Manhattan method: in which there exist many equal-length paths between two points in city-block space. Here is what the formula look like [4]:

$$d(a,b) = d(b,a) = \sum_{i=1}^{n} |b_i - a_i|$$

The last important method which we noticed us very useful because it can help derive many of the other methods, including the ones we previously displayed. Euclidean distance and city-block distance are special cases (different values of p) for the Minkowski metric. Note: $p = 1$ gives city-block distance. $p = 2$ gives Euclidean distance. As p increases, increasing emphasis is given to large differences in individual dimensions. Here is the Minkowski formula [5]:

$$d(x,y) = d(y,x) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

We can see now that the Minkowski can be seen as a parent function to many of the distance formulas around, because as we change the value of p we get different formulas which help analyze distances in different ways which can be useful depending on the problem domain.

The data used comes from songs that are familiar to me, and were also available from music books and sheet music. Some songs were directly transcribed from the books and or sheet music as well as other songs that I played on the guitar in order to create the note sequence that I wanted to examine. The data come from artists such as Muse, Led Zeppelin, Spirit, Johann Sebastian Bach, Ludwig Van Beethoven. Over all we have a pool of 10 songs which we can use to find the difference between each other and gather some data from that. Since we did a combination computation, we have 10 and we chose 2, which yields 45 different combinations which were then ran by each of the 7 methods previously described.

Results

As previously mentioned, we used seven different computations to find the distance between the songs we used in the project. We found out that the $dist(x, ...)$ and the hard coded equivalents of the same methods do not yield the same output which then led us to only use the hard coded versions of the Euclidean and Manhattan methods.

0.1 Question 1

In the first question we were looking for a similarity among songs of a given artist. We discovered that by gathering all the songs we had from one artist, and taking the standard deviation from all the difference among the songs from Muse, we obtain $0 : 9771271$ which lands between the 25% and 50% percentile, meaning that on average the songs we tested are going to be 50% to 75% similar and that is quite significant and could be seen as a trivial result since its from the same artist.

The second artist we test was Led Zeppelin, which after taking the standard deviation from the results we had, we obtained: $0 : 5117152$ which for this case according to the percentile puts this songs between 0% and 25% which means that the songs were 75% to 100% similar to each other, which is even more impressive than the results from Muse.

0.2 Question 2

The second question for the research looks into the differences and similarities between two different artists.

One case which gets its own section will be the be the one between Led Zeppelin and Taurus because of their legal battle over the similarity of the song Stairway To Heaven by Led Zeppelin to that of Taurus's Spirit. The difference once calculated come out to be: 2.211234. Then we went on to calculate the difference between the song Spirit by Taurus v. Heartbreaker by Led Zeppelin. The difference came out to be 2.660298 which shows that the song from Taurus is more similar to Stairway To Heaven than it is to Heartbreaker.

When comparing Muse to Led Zeppelin, the standard deviation between the two groups comes out to be: 0.2510109 among 12 songs analyzed. This shows us that songs by Muse and Led Zeppelin are more related to each other in terms of note sequence than the song from Spirit v. Led Zeppelin.

0.3 Question 3

The last question looks into the ability to tell whether a song that is being compared to an original song is in fact just a randomized version of itself. For this, we created randomization for 4 different songs. For each, 4 songs were created from each of the original. The overall consensus is that yes in fact we can tell if two songs are very similar to each other, than this could mean that one of the songs is just a randomization of the original song.

Here is the data for part 3:

Song A, Standard deviation is 0.1104401:

0%	25%	50%	75%	100%
1.535138	1.582441	1.630327	1.695206	1.793490

Song B, Standard deviation is 0.1265098:

0%	25%	50%	75%	100%
0.3604572	0.4616551	0.4958364	0.5395077	0.6691757

Song C, Standard deviation is 0.1857159:

0%	25%	50%	75%	100%
1.272927	1.347967	1.474408	1.601899	1.680093

Song D, Standard deviation is 0.5665721:

0%	25%	50%	75%	100%
0.2292682	0.3034414	0.4168350	0.7451700	1.4641674

Song D is Layla by Eric Clapton which was the only song among the 4 tested to show that its randomized songs can still come up to be significantly different.

Conclusions and Implications

What did you learn from this study? What are the implications of your work to practice (I call this the so what section: why should someone that does not know you want to read your work)? Future research questions and suggestions for treating those.

This study showed us the difficulties of gathering and organizing data, as well as finding suitable mathematical tools to compute and make sense of the data. As well as to how to present this in a formal way. Even thought the music picked was of a certain genre or restricted to what songs were easily accessible, we are confident to say that anyone interested in music could use our approach and find similar outcomes when analyzing music of their taste. The important take away from the project is not how similar Muse is to Led Zeppelin or how different other artist are but to see that this could be the start of a general process to determine if artists are plagiarizing and up to what extend.

Future research questions would be to create more data and to compare it to different genres like much popular music today comes from sampling old popular tunes, we would like to see which era has been more heavily plagiarized. Also, streamlining the process of gathering songs and transcribing would be something to look into in order to make the process faster and it would help reduce human mistakes.

References

[1] Cope, D., & Hofstadter, D. R. (2001). *Virtual music: Computer synthesis of musical style*. Cambridge, MA: MIT Press.

[2] What is R? (n.d.). Retrieved December 09, 2016, from https://www.r-project.org/about.html

[3]Euclidean distance. (n.d.). Retrieved December 01, 2016, from https://en.wikipedia.org/wiki/Euclidean-distance.

[4]Taxicab geometry. (n.d.). Retrieved December 01, 2016, from https://en.wikipedia.org/wiki/taxicab-geometry.

[5]Minkowski distance. (n.d.). Retrieved December 01, 2016, from https://en.wikipedia.org/wiki/Minkowski-distance.