

Music Classification - Pick a better title later

Christian Johnson

Electrical Engineering Department
United States Coast Guard Academy
New London, Connecticut 06320

Email: Christian.S.Johnson@USCGA.edu

Daniel Nusraty

Electrical Engineering Department
United States Coast Guard Academy
New London, Connecticut 06320

Email: Daniel.Y.Nusraty@USCGA.edu

Joshua West

Electrical Engineering Department
United States Coast Guard Academy
New London, Connecticut 06320

Email: Joshua.C.West@USCGA.edu

Abstract—This is an abstract

I. INTRODUCTION

A. Motivation

With the advent of music subscription services, many people have become accustomed to automatically generated playlist, tailor-made to their own personal taste. These services provide such playlists at the push of a button, analyzing the user's listening history to continuously recommend similar songs. Providers such as Pandora, Spotify, and LastFM maintain massive databases containing context information on millions of songs in order to present the best recommendations. For those who prefer to maintain their own local music library however, the options for tailored music recommendations are dramatically reduced. In this paper, we seek to explore the application of digital signal processing techniques in order to implement similar music recommendation functionality that will work with local music files.

B. Similar Work

II. THEORY

A. Background

The majority of music analysis is based on a family of functions known as Fourier Transforms. A Fourier Transform is responsible for transforming time-domain audio information into a frequency-domain representation. In the time domain, sound is represented as amplitude as a function of time. In the frequency domain, sound is instead represented as a function of magnitude based on frequency. What is important to recognize, is that in both cases the signal is still continuous, meaning unbroken. A continuous signal contains a great deal of information that makes it difficult to perform operations on. In order for a computer to be able to process such a signal, that amount of information must be reduced through an operation known as *sampling*. Sampling replaces the continuous signal with discrete representation; an array of values at (typically) evenly spaced indices of time. The normal Fourier Transform cannot operate on discrete signals however, which is where a specific type of Fourier Transform known as the Discrete Fourier Transform (DFT) comes in. The DFT is comprised of

a single summation operation, which will produce an array of complex number representations.

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} kn} \quad (1)$$

Each value in the DFT output $S[k]$ is a complex number that represents a point in an N dimensional vector space. The magnitude and phase of that point represent the content of the time-domain signal at frequency $\frac{2k\pi}{N}$, where N represents the length of $S[k]$. These values depend a great deal on a concept known as *sampling frequency*, which refers to the time between each sample taken of the original time-domain signal. Each 'bin' of $S[k]$ represent a collection of frequencies, $\frac{-F_s}{2N} + \frac{k*F_s}{N} < f < \frac{F_s}{2N} + \frac{k*F_s}{N}$. As such, it is important to recognize that $S[1]$ does not directly correspond to a specific frequency value, and in order to find the DFT output for, say, 20Hz, one must determine which bin this frequency will fall into.

1) *The FFT*: In its original form, the DFT is quite computationally expensive and complex, comprised of $\mathcal{O}(n^2)$ operations. An algorithm known as the Fast Fourier Transform (FFT) dramatically simplifies this complexity, reducing the expense to $\mathcal{O}(n \log(n))$ operations. It does this by exploiting symmetry within the DFT. Equation (1), the basic representation for the DFT, is periodic about N , i.e. $X_{k+l*N} = X_k$ for any integer l . The FFT uses this relationship to break the DFT into even and odd components

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N/2-1} x(2m) * e^{-j2\pi km/(N/2)} \\ &\quad + \sum_{m=0}^{N/2-1} x(2m+1) * e^{-j2\pi km/(N/2)} \end{aligned} \quad (2)$$

B. Analyzing Music

How music is structured FFT output

III. EXPERIMENTAL PROCEDURE

A. *Taking the FFT of an MP3 File*

B. *Shrinking our Sample Size - CORRECT terminology?*

C. *Data Storage and Access*

IV. RESULTS

V. FAKE BIBLIOGRAPHY

Move all these sources to a .bib file later and import using BibTex

A. *Similar Work*

<https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022>
<https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition>

B. *Theory*

<https://dsp.stackexchange.com/questions/5915/what-is-the-meaning-of-the-dft> <https://dsp.stackexchange.com/questions/26927/what-is-a-frequency-bin> <https://stackoverflow.com/questions/10754549/fft-bin-width-clarification>

C. *FFT*

<https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter24.03-Fast-Fourier-Transform.html>