TJHSST Senior Research Project
Computer Music Analysis
Computer Systems Lab, 2007-2008

Josiah Boning

April 4, 2008

Abstract

Although music is one of the most universal aspects of human culture, it is very difficult to define. Most definitions of music have been dependent on attributes such as rhythm, melody, and harmony, which are extremely subjective, so the ability to identify music has been limited to humans. This project aims to better define "music" by applying machine learning techniques to music analysis and recognition, allowing computers to autonomously identify whether a given audio sample is musical in nature.

Keywords: music analysis, machine learning

1 Introduction

This project aims to unify machine learning and signal processing techniques in a program that will learn to distinguish between musical and non-musical audio recordings. The end product will be a program that will learn to identify audio samples of musical nature. The program will be "trained" on a number of audio samples typically agreed upon as "music" as well as a number of non-musical samples. After sufficient training, the program will be able to distinguish between musical and non-musical audio recordings.

2 Background

Computers have already been used to perform analysis of music. In 1999, Bigerelle and Iost determined that different genres of music could be distinguished by fractal dimension, and in 2004, Basili et al. showed that machine learning techniques could successfully indentify musical genres[?][?]. Other research has attempted to deconstruct music in terms of rhythmic and melodic patterns, and even looked at writing software to generate music conforming to such patterns[?]. However, as Bigerelle and Iost point out, each instrument has a different sound quality, and composers write music with these timbral differences in mind. Simply analyzing the notes on sheet music precludes the use of these differences in the analysis. Audio recordings, in contrast, allow analysis of exactly what the composer intended his audience to hear.

3 Program Design

3.1 Spectral Decomposition

The program performs a Fourier transform on the audio data to obtain a frequency spectrum of the audio file. This data will then be processed in the neural network. The program uses the open source FFTW library to perform the transform.

3.2 Fractal Dimension

The program calculates the fractal dimension of the input audio data using two methods, the Variation method and the ANAM method.

The Variation method:

$$\Delta = \lim_{\tau \to 0} \left(2 - \frac{\log \left(\frac{1}{b-a} \int_a^b \left| max(f(t))_{|x-t|<\tau} - min(f(t))_{|x-t|<\tau} \right| dx \right)}{\log \tau} \right) \quad (1)$$

The ANAM method:

$$\Delta = \lim_{\tau \to 0} \left(2 - \frac{\log \left(\frac{1}{b-a} \int_{x=a}^{x=b} \left[\frac{1}{\tau^2} \int_{t_1=0}^{\tau} \int_{t_2=0}^{\tau} |f(x+t_1) - f(x-t_2)|^{\alpha} \right]^{1/\alpha} dx \right)}{\log \tau} \right) \quad (2)$$

3.2.1 Accuracy

Because the Variation and ANAM methods are being applied to discrete audio data, the accuracy of the numerical integration is limited by the sample rate. To work around this difficulty, cubic splines will be used to interpolate between audio data sample points and increase the accuracy of the calculations.

3.3 Machine Learning

The program uses a feedforward perceptron topology with a single hidden layer. The perceptron will be trained on audio samples that clearly are or are not music. The network will probably use a Back Propagation learning algorithm.

3.4 Program Architecture

The program is written in C. The source is organized into files by subject: the input functions, fourier transformation bindings, fractal dimension calculations, and neural network functions each have their own file. The main file is simply a driver that calls the functions in the other files.

4 Conclusions

The program can currently read audio data from WAV files. It can perform spectral analysis to determine the frequency makeup of the audio. It can also perfom fractal dimension calculations, although the numeric integration methods are currently somewhat inaccurate. The integration methods will be improved; instead of using Riemann sums, the program will model small sections of the data with cubic splines and perform the integrations mathematically. The program also currently creates rudimentary neural networks, but does not have a training algorithm in place.

References

[1] Basili, Roberto, Alfredo Serafini, and Armando Stellato. 2004. "Classification of Musical Genre: A Machine Learning Approach." Presented at the 5th International Conference on Music Information Retrieval.

[2] Bigerelle, M., and A. Iost. 2000. "Fractal Dimension and Classification of Music." *Chaos, Solitons & Fractals.* 11(14):2179-92.

[3] Leach, Jeremy, and John Fitch. 1995. "Nature, Music, and Algorithmic Composition." *Computer Music Journal.* 19(2):22-23.