Composer Classification Using Symbolic Music Data: Progress Report

James Diotte
CUNY Graduate Center
(Dated: February 28, 2017)

Abstract: The purpose of this project is to extract features found in symbolic music data and train a classification model to discern between different classical composers, in this case Beethoven and Mozart. We used the built in feature extraction module found in the Python package music21 to extract our feature vectors. We chose an initial 177 features based on the work of several other researchers. The next step in our project will be the process the data and train a classification model.

I. INTRODUCTION

Using machine learning to distinguish between the works of two composers is related to the broader task of automated music genre recognition. The task is to look for inherent characteristics in the music which allow us to classify. Music is typically stored either directly as an audio file or indirectly in a symbolic format such as MIDI file which contains instructions for a machine to reproduce the song (essentially like digital sheet music), The latter of which we will use in this project. Beethoven and Mozart were chosen for this project as both were very prominent composers during the Classical period of Western Music and as such there music is on some level very similar. The challenge to distinguish between the two is made further difficult by the fact that they both wrote pieces for a large variety of different instruments and settings, e.g. piano, viola, violin music etc as well as symphonies, operas, concertos etc. Also their music spans a large period of time in their individual lives over which their style could change. We will use the music21 Python package [1] in this project to process and extract 177 features based on pitch and rhythm characteristics from 642 and 680 works by Beethoven and Mozart respectively.

II. LITERATURE REVIEW

The features we chose are based on the work of Cory McKay in his doctoral dissertation on automatic music classification [2], furthermore McKay had previously implemented a Java package jSymbolic to process MIDI files and extract 111 of the features he describes in his dissertation [3]. The music21 developers have recreated 57 of the jSymbolic features as well as designed a few of their own. We reflected the work done by Cuthbert, Ariza and Friedland [4] in choosing our features which ended up totalling 177.

III. THE DATA

A. Extraction

The raw data used was 642 and 680 MIDI files of music from Beethoven and Mozart (resp.), the music selection spans a wide range of the works of both composers. We use the music21 DataSet object to process the data into feature vectors. The DataSet object has two containers one for parsed MIDI files and another for a set of feature extractor methods, once the MIDI files and the feature extractors are loaded into a DataSet object there is a process method which applies all the feature extractors to all the parsed MIDI files and then gives the option to save the results as a csv file for easier loading and processing. See [4] for more details.

B. The Features

The features extracted from the MIDI files are all high level statistics coming from the pitch and rhythm information. Some of the features are themselves are vectors. The largest vector feature is the Pitch Histogram which is an element of $\mathbb{R}^{128}$, each component of which is the relative frequency of the particular note taking into account its octave. The 128-dimensional Pitch Histogram Feature is related to the Fifths Pitch Histogram which is a transformation of the full histogram of notes onto a circle of fifths, the result of which is a vector in $\mathbb{R}^{12}$. These two features alone account for 140 of the 177 total features, another 12-dimensional feature is the Pitch Class Distribution. The remaining 25 1-dimensional features will be listed here:

- Initial Time Signature 0
- Initial Time Signature 1
- Compound or Simple Meter
- Triple Meter
- Quintuple Meter
- Changes of Meter
- Most Common Pitch Prevalence

- Most Common Pitch Class Prevalence
- Relative Strength of Top Pitches
- Relative Strength of Top Pitch Classes
- Interval Between Strongest Pitches
- Interval Between Strongest Pitch Classes
- Number of Common Pitches
- Pitch Variety
- Pitch Class Variety
- Range
- Most Common Pitch
- Primary Register
- Importance of Bass Register
- Importance of Middle Register
- Importance of High Register
- Most Common Pitch Class
- Unique Note Quarter Lengths
- Most Common Note Quarter Length
- Most Common Note Quarter Length Prevalence

All of the features except the last three related to note quarter lengths are music21 implemented jSymbolic features based on McKay [2], the last three are native features to music21.

IV. GOING FORWARD

This is the extent of the project thus far, extracting the data from the MIDI files is very slow and took over 20 hours of computational time on an Intel i5- 4690 @ 3.9 GHz. Now that the data has been extracted and saved into a csv file it can be rapidly loaded for processing. In the next stage of the project we plan to process the data in order to train a model with linear-classification boundary. The data is 177-dimensional and there are only roughly 1200 data points and as such there may be issues arising from the curse of dimensionality so we plan to compare our models on various transformed versions of the data. Some avenues we plan to explore include: PCA, testing for significant differences in the distributions of features between the two classes, testing for colinearity of any features, data normalization, using only the circle of fifths histogram as opposed to full histogram, and other transforms we may encounter in the literature. The models we plan to use for testing are logistic regression, linear svm, single layer perceptron and naive Bayes. Furthermore we may compare the results of linear classifiers to non-linear to see if there is any performance to be gained.

[1] M. Cuthbert and C. Ariza: "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 63742, 2010.

[2] C. McKay:"Automatic Music Classification with jMIR," Ph.D. Dissertation, McGill University, 2010.

[3] C. McKay and I. Fujinaga:"jSymbolic: A feature extractor for MIDI files" *Proceedings of the International Computer Music Conference*, pp. 3025, 2006.

[4] M. Cuthbert, C. Ariza and L. Friedland: "Feature Extraction and Machine Learning on Symbolic Music Using the music21 Toolkit," *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011