David Diston
September 20th 2020

BrainStation Capstone Project

The AI Pianist

An LSTM RNN Classification

Developed by David Diston

David Diston
September 20th 2020

Table of Contents

David Diston
September 20th 2020

Abstract

The purpose of this project was to classify music by composer. I also sought both to classify music based on the century during which it was composed, and to compare the classification accuracy of binary models each predicting between two different composers.

Midi data was collected from the e-Piano Competition of the University of Minnesota School of Music and preprocessed to bin note values and convert the midi data to arrays. LSTM RNN architecture was used to produce each of four models. Test accuracy for my ten-class composer classification model was 61%, for my century classification model 80%, and my two binary composer classification models were 67% and 91%. Future research based on these results includes further testing with binary composer classification models to better understand the variation in classification success, and how the models are making predictions. There are also several applications for models that can classify and play midi data in the style of a particular composer.

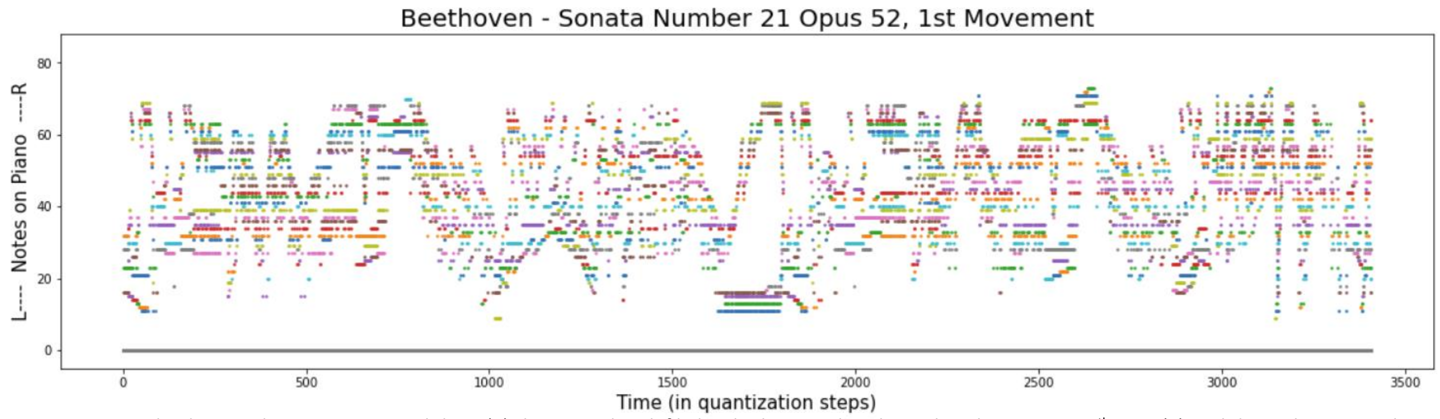


Figure 1: sample plot visualizing preprocessed data. (1) the original midi file has had note values binned to the nearest 16th note (2) each binned time step has been appended to a list of time steps (3) the list of time steps has been converted to an array. This plot shows the 88 keys of the piano along the y-axis, and each timestep of the piece along the x-axis. Each point on the plot represents that note being played at that time. Plotting data in this way shows the general distribution and density of notes over the course of a piece.

Introduction

The goal of this Capstone Project was to investigate the predictive power of Machine Learning models to classify music by different metrics, primarily by composer. This goal is summed up in the following problem statement:

Can a Machine Learning model predict the composer of a piece of music?

To narrow the scope of this problem, I decided to focus only on classical music, as I have substantial expertise in the subject matter. Most past research I had found used music wave data converted to spectrograms and CNN architecture to classify music by genre. I decided to investigate the less-traveled path and focus on music information data with RNN architecture, where the primary structural format is midi. I also chose to work with piano music as this includes the complication of polyphony (playing multiple notes simultaneously like an orchestra), while still being coded as a single instrument track in midi format.

Data Collection

Data for this project was collected from the e-Piano Competition of the University of Minnesota School of Music:

http://www.piano-e-competition.com/

The competition maintains a public archive of all midi recordings from past competition years. Due to the structure of the website, web-scraping the data was not a possibility. Data collection involved individually downloading, and naming (with the performer name, nationality, gender, and composer name, and piece name) each of the more than 2500 performance files. While this process encompassed a substantial amount of manual upfront work, this paid dividends when categorizing and iterating over files during EDA, preprocessing, and modeling.

Exploratory Data Analysis

Several interesting observations were made regarding the distribution of the dataset. I observed a substantial gender imbalance among the 343 unique performers in the dataset. Similarly, a substantial imbalance in performer nationality exists. However, the most pertinent observations for this project concern the distribution of pieces by composer in the dataset. Figure 2 outlines this distribution for the top 10 composers by piece prevalence in the data. Since the top 10 composers account for 74% of all pieces in the dataset, I concluded that this was both a representative sample of the data, and a reasonably sized class problem to solve.

Composer	Pieces
Chopin	396
Beethoven	318
Liszt	283
Schubert	197
Bach	181
Prokofiev	114
Ravel	109
Rachmaninoff	109
Haydn	105
Schumann	98

Figure 2: the top 10 composers ranked by number of pieces in the dataset, accounting for 74% of the data

Data Preprocessing

Midi files are composed of a series of on/off note messages (Figure 4), rather than actual audio wave data. Each midi message contains the note value (from 21-108), the note velocity (from 0-127), and the time delta in microseconds since the last message. To reduce data density, I chose to bin all notes to the nearest fraction of the beat as shown in Figure 3. Arrays for each piece were created where all notes occurring within a binned beat were appended as a single row in the array, where all arrays have 88 columns (one per piano note) and the value of each observation is the velocity of the note being played at that time.

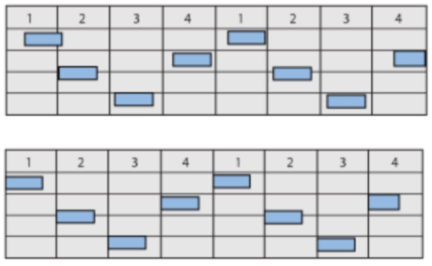
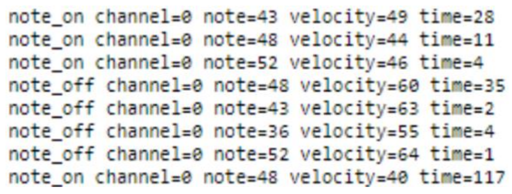


Figure 3: beat binning (quantization) moves notes played in absolute time (top) to the nearest beat or fraction of beat (bottom), reducing the number of time steps required to represent the data in an array
© Iman Malik, Neural Translation of Music Style

Arrays ranged in length from several hundred, to a few thousand rows after binning note values. To reduce this length for modeling, and to correct for the class imbalance between composers, 5000 sample arrays of 100 rows were taken at random for each composer. This was done after splitting the validation/test sets to prevent data leakage.



```
note_on channel=0 note=43 velocity=49 time=28
note_on channel=0 note=48 velocity=44 time=11
note_on channel=0 note=52 velocity=46 time=4
note_off channel=0 note=48 velocity=60 time=35
note_off channel=0 note=43 velocity=63 time=2
note_off channel=0 note=36 velocity=55 time=4
note_off channel=0 note=52 velocity=64 time=1
note_on channel=0 note=48 velocity=40 time=117
```

Figure 4: sample of midi message structure, showing the note on/off command, pitch, velocity, and time step

Modeling

LSTM RNN architecture was used for this project due to its success with sequential data (logistic regression and CNN testing models were also created with poor and average results respectively). To create a baseline model for future comparison, I chose to classify between human performed midi files (with high note and velocity variation) and computer performed midi files (with rigid timing and minimal velocity variation). Model tuning was performed to find the optimal model structure, with overfitting reduced by regular dropout. Several combinations of LSTM and Dense hidden layers were tested with varrying numbers of nodes (factors and multiples of 88 [number of array columns] were used for tuning). The optimal model structure for this binary-class human vs. computer problem was found to be:

88 Input LSTM nodes -> 88 hidden LSTM nodes -> 1408 hidden Dense nodes -> 2 dense output nodes

resulting in 252,738 trainable parameters. Tuning found no significant decrease in loss or increase in validation accuracy after 20 epochs. Test accuracy for this model was 99%.

I performed similar tuning for my 10-class composer classification model, finding an optimal structure of:

44 Input LSTM nodes -> 44 hidden LSTM nodes -> 1408 hidden Dense nodes -> 10 dense output nodes

resulting in 116,522 trainable parameters, and no significant increase in validation accuracy after 15 epochs. Test accuracy for this model was 61%, with class f1-scores ranging from 40%-87%.

Further investigation of ML classification accuracy was performed by grouping composers by century. Model tuning found the same optimal model structure as above (with 3 output nodes), and no significant improvement in accuracy after 20 epochs. Test accuracy for this model was 80%, with class f1-scores ranging from 74%-85%.

Finally, binary-class models were created to assess the classification accuracy of composers with similar and varying demographics. My Debussy-Mozart classification model achieved a test accuracy of 91%, while my Prokofiev-Rachmaninoff model achieved a test accuracy of 67%.

Conclusions

Composers with a consistent compositional output style (Ravel/Haydn) saw high classification accuracy in the 10-class composer model. Despite a varied output, Bach saw a high classification accuracy as well. However, it was discovered that almost all Bach pieces in the dataset were preludes and fugues, helping to explain this high accuracy observation. Liszt, with a varied compositional output, and several compositional quotes from past composers written in his own style, saw a comparatively low classification accuracy.

Classification by century resulted in a predictably high classification accuracy, as each class is a gross approximation of a compositional epoch. My hypothesis that the model was classifying based on perceived compositional style was bolstered by observations from the confusion matrix showing most classification errors were between adjacent centuries representing more similar compositional styles than distant centuries.

Finally, my binary-class composer models again produced predictable results when compared to those of previous models. Debussy and Mozart (1800's French and 1700's Austrian respectively) saw much greater classification success as composers of disparate style, when compared to Prokofiev and Rachmaninoff (both 1900's Russian) who shared a much more similar compositional style.

Further Research and Applications

I would like to create additional binary-composer classification models to understand the variation of classification success more thoroughly, dependent on the composers being compared. This may also help to provide insight and interpretability for how these models are making predictions.

Applications in music education are possible for LSTM models that can classify and play midi music in the style of a particular composer. This would be especially useful for auditory earners in music history classes for example. Additionally, in the post-COVID-19 music performance industry, there is a substantial deficit of electronic tracks usable for performance collaboration by solo musicians. Such a model could fill a large product gap as classical performance shifts to an ever-more prevalent online format.