Delicious Music

John Ebeling, Jake Murphy, Joseph Henderson, Max Brown, and Timothy Brown

March, 25, 2023

1 Abstract

Treating music as time series data, we train a Hidden Markov Model on music midi data for various classical composers using Expectation Maximization. We then use a variety of classification techniques, including KMeans, K-Nearest Neighbors, Spectral Clustering, Random Forest, and ExtraTrees to predict the composer of a piece by its transition matrix. Our results indicate that clustering methods (KMeans and Spectral Clustering) are not effective for classifying composers. A K-Nearest-Neighbors method performed only marginally. However, tree-based classification techniques (RandomForests and ExtraTrees) are very effective at classifying the transition matrices into composers.

2 Problem Statement and Motivation

The term "music" is deliciously abstract. According to the Merriam-Webster dictionary, it can be defined as "the science or art of ordering tones or sounds in succession, in combination, and in temporal relationships to produce a composition having unity and continuity" (Merriam-Webster). Such a definition begs the attention of Mathematicians. If music truly is scientific, then this implies the existence of a set of possible observations made up of the possible orderings of tones or sounds, as well as the possibility of hidden states. Our hypothesis is that the composers of the music can be thought of as the unobserved hidden state for any music. We test our hypothesis using Hidden Markov Models with differing classification methods to determine the composer of a given song. Simply put, we will train our model on a large body of musical data to identify the most likely hidden state transition matrices for each composer. Then we will test our algorithm by feeding it musical data and quantifying the accuracy of its classification by composer. Assuming its success, we will attempt to use the discovered transition matrices for musical generation.

3 Data

Music is commonly stored using the .midi filetype. Unlike audio files (.mp3, .wav, .aac) which store frequencies, the .midi filetype (Midi data) stores the individual notes, rhythms, and dynamics of each instrument the music involves, much like sheet music. We use this sequence of notes and rhythms as our time series for this project.

While our methods would apply to any type of music, our analysis primarily focuses on classical music by composers such as Beethoven, Motzart, and Chopin. These Midi files were retrieved from

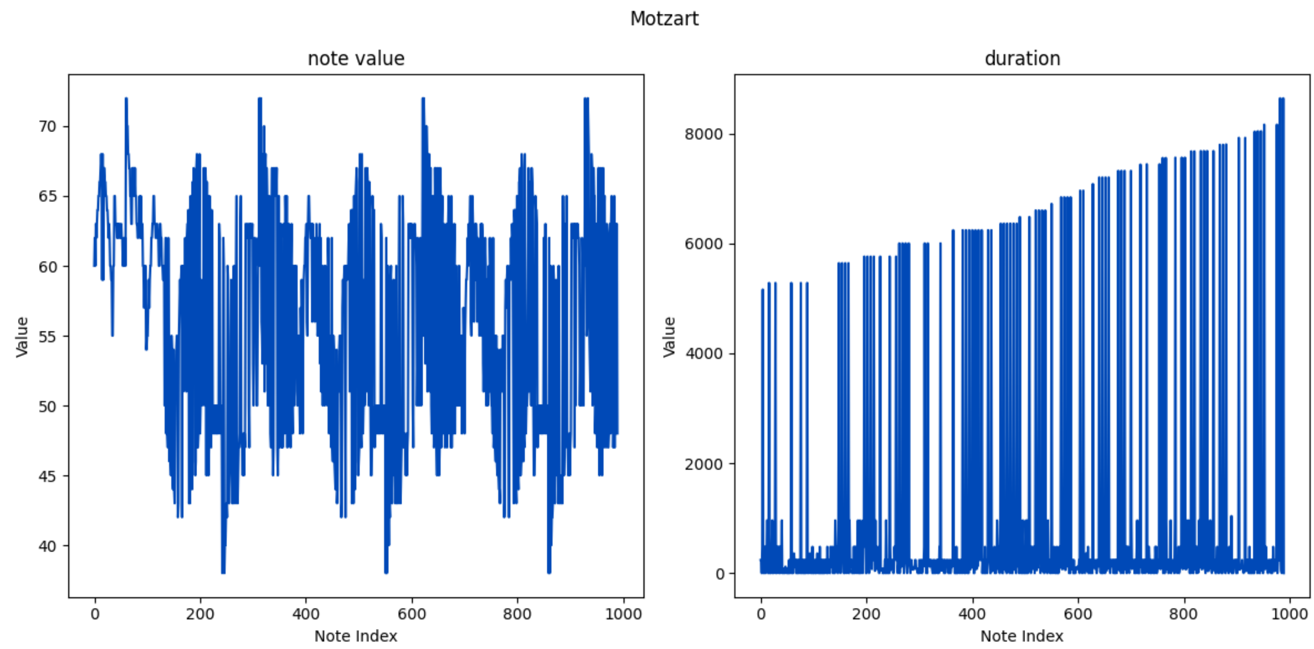

Figure 1: The sequential note value and rhythm values for an individual midi file

www.piano-midi.de. We used Andrew Chenk's midi-to-csv converter(include reference) to convert the midi files to csv files. We denoted 20% of this data as test data and trained on the remaining 80%. This train-test split was performed once and the split data was stored.

Our first dataset was a collection of midi data that had already been converted to csvs. We combined the data from all the songs into one large Multi-Index Pandas Dataframe that we used to analyze the data collectively. We were able to use this Dataframe to further clean our data, removing erroneous values and binning extreme values to decrease the number of possible states in our model. We did have to clean our data as it contained many Nan values for note value and duration. We believe that this data was either gibberish (meaning the midi file essentially doesn't find meaningful information in that point and continues to the next), or the Nan values were errors. Either way, it did not make sense to fill in the values with a forward or backward fill because that could tell the model that a note existed where it should not. To circumvent this problem, we treated Nan values in note duration and value as an additional discrete value for data, meaning that the model could transition to Nan if that state was likely. This approach aknowledges that Nan values can occur in Midi files and may happen on occasion.

Our data alone showed some interesting results, as shown in the figures 2 and 3. During Haendel's day, piano keyboards were actually shorter than they are today. Piano keyboards grew as classical composers felt the need to have more notes available to them. This is reflected in the visualizations we have made: the second plot (Haendel's work) has a much smaller range of notes than the first plot (Chopin's work).

While the first dataset worked well for visualizations, analysis with it was proving very difficult due to the size of the total state space. We found an archive of midi data online and used Andrew Chenk's pipeline to get csv files from them. The csv files produced from his code were cleaner than those from the archive of csvs (containing less Nans) than those we were originally working with and analysis proved much easier. Still, we had to be careful with the total state space size.

We set off on a crusade to get information from experts, talking to both Dr. Luke Howard and

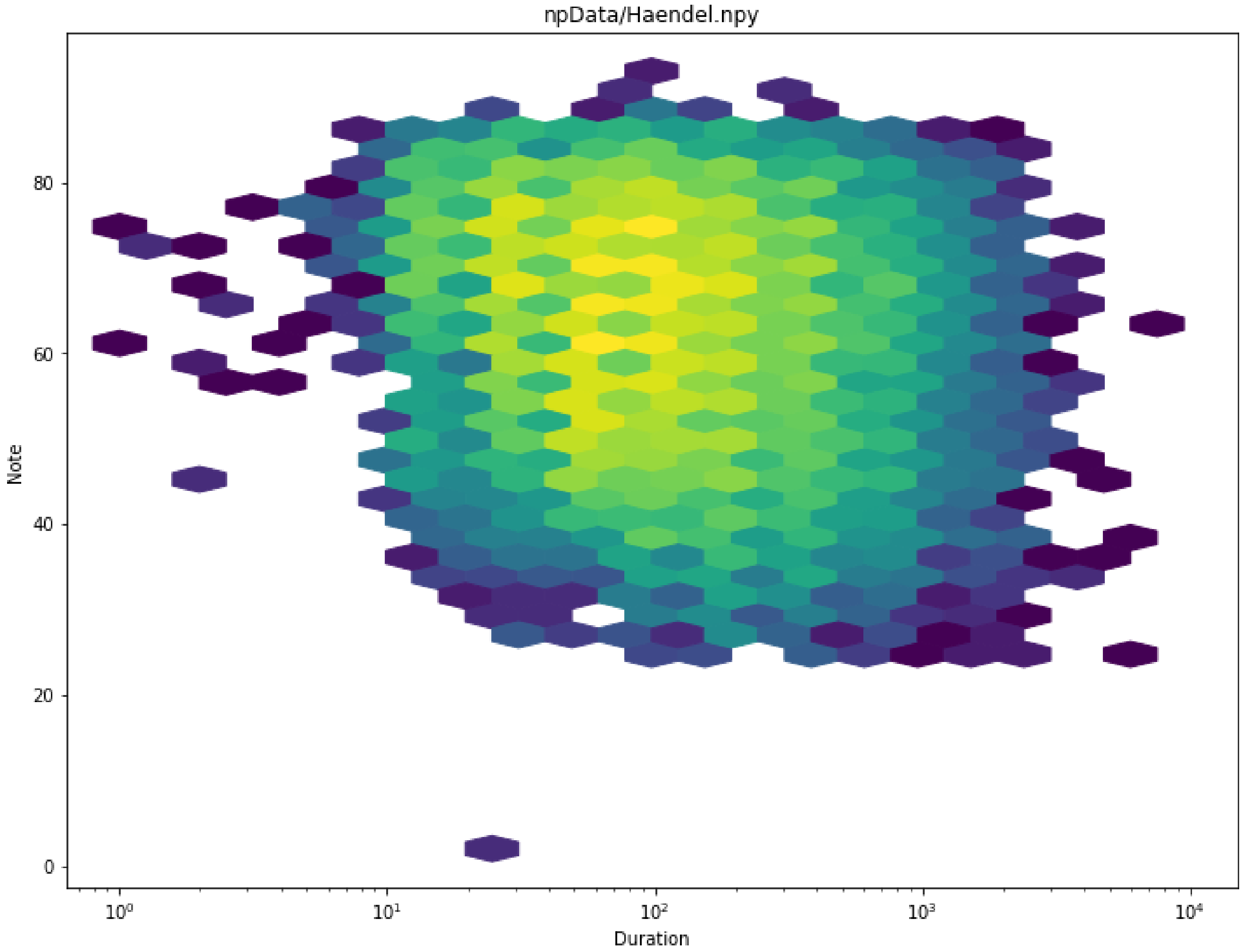

Figure 2: Haendelnotes

Dr. Brent Yorgason. From their advice we decided to train the HMM on only the notes and their durations, effectively eliminating 4 features from the dataframe. Additionally, they recommended that we try to classify single instrument music from composers like Schubert, Haydn, Handel, J.C Bach, etc, since those musicians tended to have more predictable forms and would probably be easier to learn than more virtuosic works.

In the end. Our analysis dataset consisted of the truncated csvs of piano works from many composers they recommended as well as several others.

4 Methods

This section explores the specifics of listing our assumptions, constructing our model, and possible variations we explored.

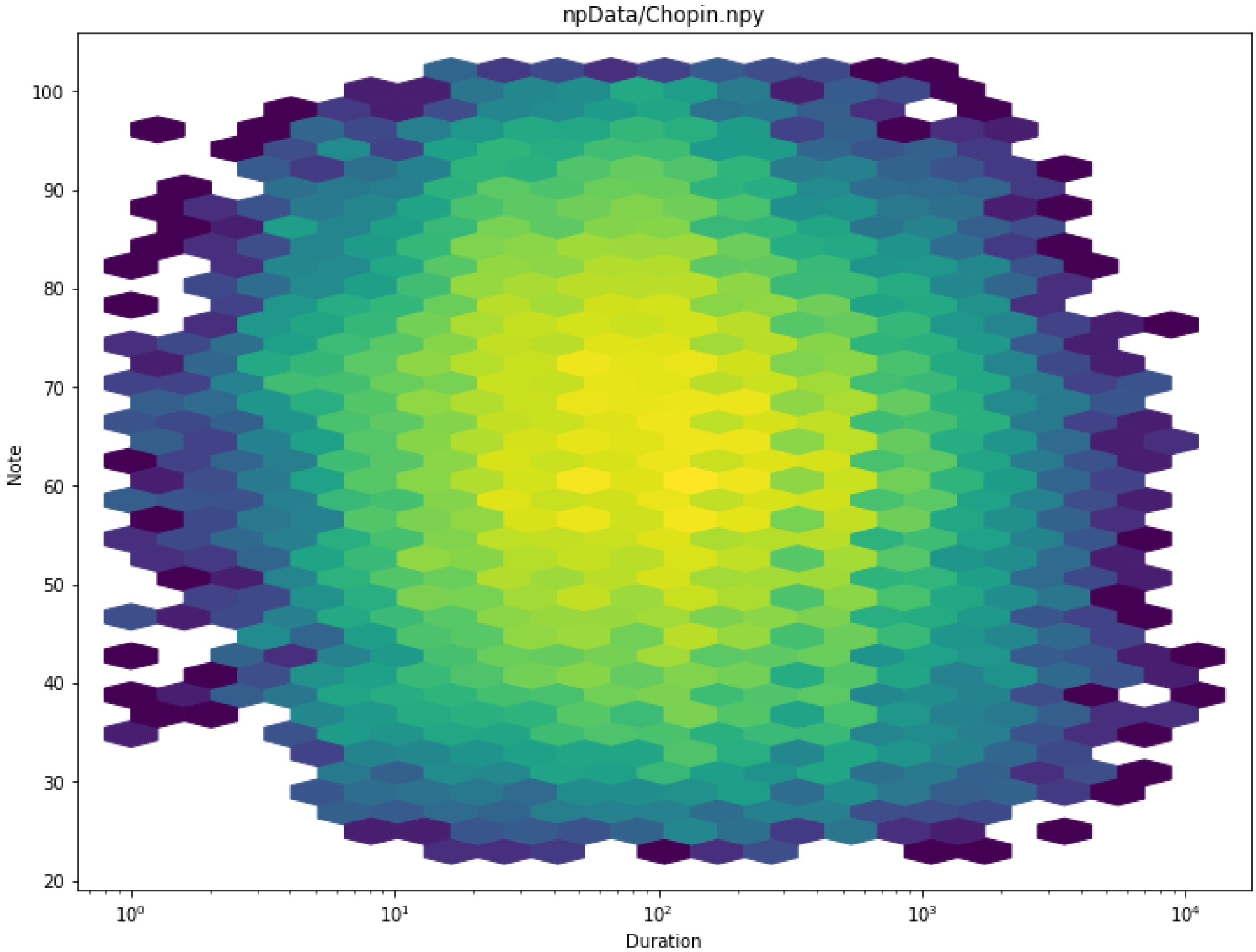

Figure 3: Chopin Notes

4.1 Model Construction

We decided to attempt classifying the pieces based solely on the HMM transition matrices from individual songs. The HMM used observed states of note duration and note value, we also defined our initial models to have 3 hidden states. We considered these two attributes the most important of the possible features to use, but computational complexity also limited our choices as more and more features was taking too long to train. Excluded attributes included the key, control, and velocity features. Originally, we used a Guassian HMM because we believed that the note durations where continuous. However, after considering our data more closely, we decided that here are only finitely many durations and note values, and we used a categorical HMM to model the pieces. Dynamax was used to estimate the transition matrices for each of the pieces. We than constructed various classification architectures to attempt to classify a song based solely on the transition matrices. We specifically used K-means clustering, K-Nearest Neighbors, and tree based methods such as Random Forest and ExtraTrees.

5 Results

Initially, we were using the more convoluted data, a Gaussian HMM, and clustering methods for classification and we did not get good results, achieving only achieved 33 % accuracy with nearest neighbors, and even worst with K-Means clustering. We also attempted to fit a linear regression to the data to cover all of our bases. As expected, linear regression also did really poorly.

Our next attempts were with the data generated from the midi-to-csv pipeline and we added the random forest method with 100 trees for classification. Random forests appear to be particularly well-suited for this classification problem. Using them we improved our accuracy enough to consider changing the dimension of our hidden state. The results of our grid search for all three methods were as follows:

Hidden State Dimension	K-Means	Nearest Neighbors	Random Forests
3	.30	.38	.61
4	.30	.38	.61
5	.46	.46	.46
6	.23	.61	.76
7	.38	.46	.69
8	.23	.38	.76
9	.46	.46	.846
10	.23	.77	.846
11	.38	.69	.61
12	.38	.30	.53

Finally, we used an ExtraTrees Classifier to cluster the F-matrices. An ExtraTrees Classifier is similar to random forest, but the maximum depth of each tree is typically smaller and the split of each selected feature is split at random. This method of classification also produced great results, yielding accuracy scores between 0.84 and 0.92.

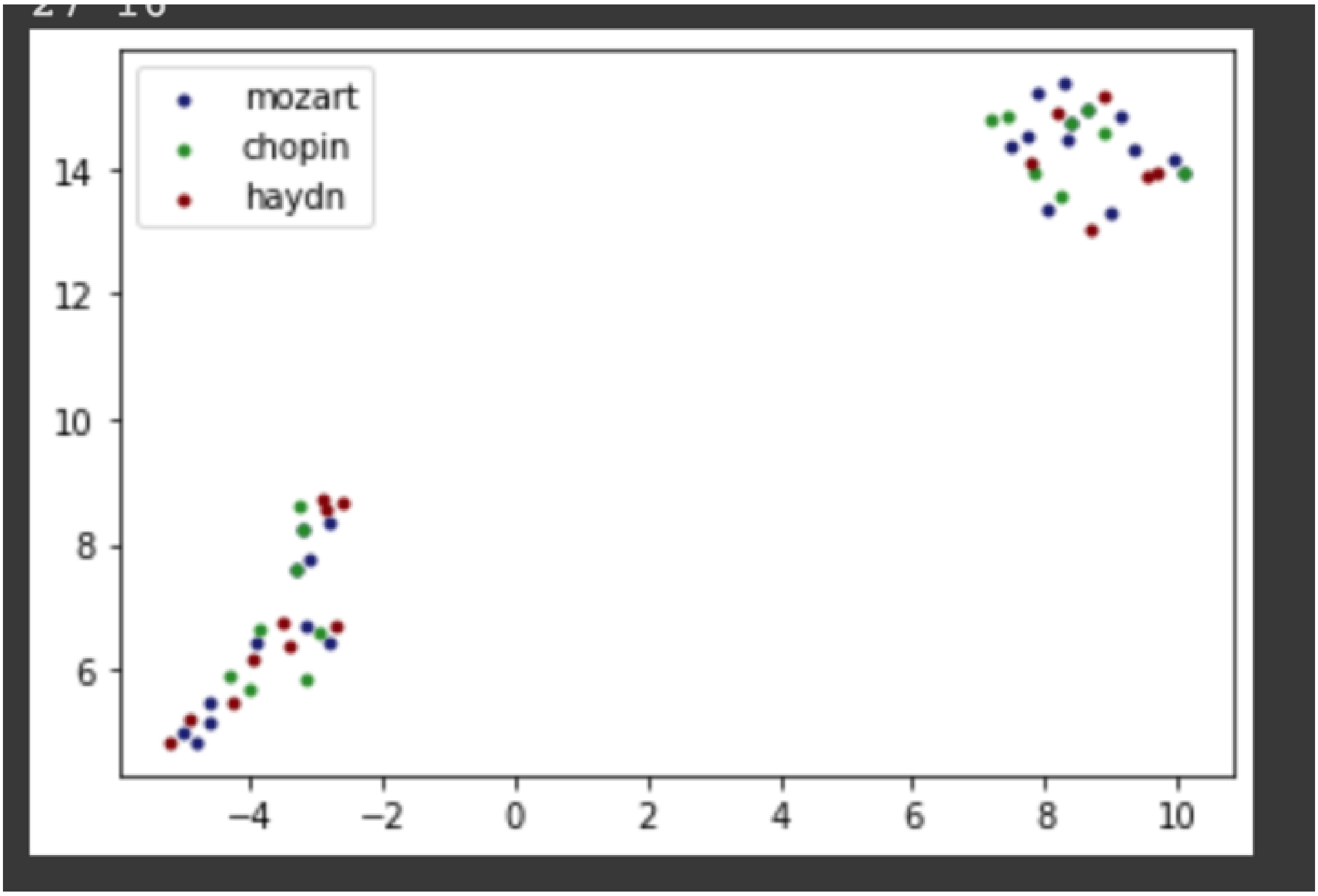

Figure 4: Umap of Data

6 Analysis

It seems that tree-based methods are most effective at classifying the transition matrices. Clustering based methods of classification were much less effective. This probably tells us that there are a few specific elements of the transitions matrices that are very informative in classifying a transition matrix. Further analysis on these methods may reveal the most important feature for the trees on average. Poor clustering results is expected from our discrete HMM which would have discrete transitions matrices, and for this same reason it would appear that nearest neighbors would outperform traditional clustering.

Another reason that our clustering methods probably did not work well is demonstrated in figure 4 showing a umap projection of the F matrices. We see in the particular case of Mozart, Chopin, and Hayden that the data does not cluster well by class. Although the projection of the data is a simplification, it suggests that our classification is probably best described non-spatially but through some other relationship between the features, as we found out with the trees.

7 Ethical Considerations

The results of our methods do not present any immediate ethical considerations. Algorithmic classification is a very well established problem and has two common ethical concerns (references). We enumerate each ethical concern below along with our consideration of its implications.

1. Privacy and Security: The models described above have been trained on classical music, already in the public domain, eliminating immediate privacy and security concerns. While others could conceivably leverage our work with data that violates the privacy and security of others, such ethical concerns are more reflected in by the unethical use of data rather than an unethical use of our algorithm.

2. Effects on Real People: There are no immediate uses of our model which directly affect individuals in unethical ways; however, we do not underestimate the potential of individuals to abuse our results.

Simply stated, our projects presents no immediate unethical concerns. We, furthermore, cannot conceive obvious ways others can abuse our methods, but acknowledge that any advancement can be abused.

8 Conclusion

Our HMM model was able to construct transition matrices that contained enough information to correctly classify by composer. The tree-based classification methods were much more effective than classification using clustering or nearest neighbors. Further research would include attempting to forecast the songs using the transition matrices that we obtained from the HMM models. We are optimistic that these learned transition matrices will be meaningful in other settings because we were able to do classification with them.

Additional improvements to our classification models stem from the conversations we had with the consultant music experts (Doctors Howard and Yorgason). Two modifications that we can try based on their expertise would be learning by phrasing and learning by modulation. Composers of the era in question used similar styles but differed in the way that they transitioned between the Tonic and Dominant keys of the music. It is possible that by training on these modulations individuals and creating transition matrices for each one we might be able to improve the accuracy of some of our classifiers, though the analytic recognition of such musical nuance is rebarbatively evasive. An easier problem may be to divide the pieces based on the musical phrasing. The problem is slightly simpler but still difficult to incorporate into the programming.

In short, HMM's coupled with Transition Matrices show promise at being an effective analyzer of the composer behind the music.

9 References

https://github.com/andrewchenk/midi-csv

Merriam-Webster. (n.d.). Music. In Merriam-Webster.com dictionary. Retrieved March 25, 2023, from https://www.merriam-webster.com/dictionary/music

https://github.com/tabrownies/Vol3Project

www.piano-midi.de

https://towardsdatascience.com/machine-learning-an-ethics-review-5f4c35112be5

https://www.journals.uchicago.edu/doi/abs/10.1086/449140