Does the Ear of the Eigenface Hear the Eigensound?

Application of PCA in Music Classification

Kate Li[1], Seyedhamidreza Sadatian[1], Benjamin Zivan[1]

[1]Georgia Institute of Technology

Abstract

In this project, we explore the classification of audio files, specifically music pieces ranging from the classical solo piano (Chopin, Mozart, and Liszt), pop music (Adele and Billboard top 30 pop songs) to jazz. Considering the immense feature-space of our data (220,500 features per 5-second sample), exploitation of eigensounds has formed the principal components of our data preprocessing. Specifically, considering low-throughput networks that are still standard in many places worldwide and the amount of data available and needed for training a useful model, we have focused on minimizing the amount of sound data (quality and decomposition) needed for fast and accurate classification.

Background and Motivations

Our problem lies at the intersection of Music Structural Analysis ("MSA"), a subfield of Music Information Retrieval (Müller 2015), a.k.a Music Informatics Research (Nieto 2020) ("MIR") and general machine learning algorithm comparison. In this problem domain, the neural network (*NN*) family of models, particularly convolutional neural networks (CNN), meet with great favor among researchers (Nasrullah 2019; Haunschmid 2020) due to their abilities to identify the key structural elements of the musical piece, such as the chorus of a pop song or motif of a symphonic movement. Preprocessing for these models using raw audio data generally involves converting the audio information via Fourier Transform into mel-spectrograms (Müller 2015; Huang 2018), which display the intensity of a given frequency per time window. A related approach that is sometimes used is a 'chromograph' (Kirkbride 2015), which maps frequency information directly onto the Western 12-semitone chromatic scale, but since this approach obfuscates intervals that span more than an octave (e.g., a ninth), the spectrogram is the standard approach. It has also been shown that using a spectrogram instead of raw audio data can increase modeling accuracy[1].

Where many papers in this domain focus on genre classification using tagged datasets such as artist20[2] or GTZAN[3], we were interested in working solely with the music itself to see the extent to which

[1] https://github.com/derekahuang/Music-Classification

[2] http://labrosa.ee.columbia.edu/projects/artistid/

[3] https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification/

dimensionality reduction via Principal Component Analysis (PCA) in combination with non-neural network (*nonNN*) models (e.g., k-NN, Naive Bayes) can classify with similar accuracy as neural networks but with greater speed. Again, we are considering low-throughput data networks in this project, so any method that reduces computational resource overhead versus neural networks while maintaining acceptable accuracy would be preferable.

As we have seen in this class, PCA works by performing eigendecomposition on the covariance matrix of the data:

$$\text{For every principal component } k \text{ we calculate: } \mathrm{w}^{(k)} = \arg\max \frac{\mathrm{w}^{\mathrm{T}}\widehat{\mathrm{X}}_k^{\mathrm{T}}\widehat{\mathrm{X}}_k\mathrm{w}}{\mathrm{w}^{\mathrm{T}}\mathrm{w}} \quad \text{for} \quad \widehat{\mathrm{X}}_k = \mathrm{X} - \sum_{i=1}^{k-1}\mathrm{X}\mathrm{w}^{(i)}\mathrm{w}^{(i)\mathrm{T}}$$

This assumption underlying PCA is that variance of the data is the most important differentiation factor. This is because the variance of the orthogonal dimensions represented by the eigenvectors is highest for the eigenvector with the largest eigenvalue.

Data Source

We use the public-domain music library[4], The Internet Archive[5], and Piano Society[6] for all audio files as our primary data source. We decided to pick three composers, Chopin, Mozart, and Liszt, from two classical music periods with distinct characteristics. Since Chopin only has piano music, we use only piano music from Mozart and Liszt to minimize the variance arising from musical instruments' characteristics. We picked a jazz ensemble with piano to keep the similarity in musical instrumentation for the 'jazz' samples.

In addition to instrumental music, we add in the mix of vocal music by including one Adele album as an audio file and a compiled audio file of recent top 30 Billboard pop songs. Vocal music productions typically include a wider range of timbres and therefore tend to have more mixing and more layers on the production level. We explore how this impacts the results of the classification models.

[4] https://imslp.org/wiki/Main_Page

[5] https://archive.org/details/audio

[6] https://www.pianosociety.com/pages/mozart_sonatas/

Methodology

Wrangling

A substantial portion of our task is to develop a fast process to transform songs into uniform samples for training and testing purposes.

Data is first downloaded to our local system for improved accessibility. All songs are then standardized from double-channel to single-channel (mono) arrays using the `audio2numpy` and `NumPy` libraries. We then homogenize all arrays into a standard sample rate using the `SoXR` library.

Preprocessing

We normalize the data by max-min scaling to the range of [0, 1] and standardize features by removing the mean and scaling to unit variance using the `sklearn.preprocessing` module. At this stage, we perform PCA for each composer's matrix to produce the composer' Eigensounds.'

Eigensounds

We export the eigensounds and evaluate what the eigensounds capture. Here are a few observations:

1. The eigensounds capture the most prominent features of the audio files.
    a. We could hear choruses of the vocal songs and the climax of instrumental music
    b. We speculate that the features captured are the segments of a music piece that has the largest volume
    c. This implies that the larger volume features/data-points dominant lower volume features/data-points
2. Eigensounds capture complete musical passages for different pieces of the music and overlaps them.
3. The jazz piece and the top billboard songs are harder to recognize any melody than the other files. This could be due to:
    a. These more modern musical genres consist of more musical dissonances than the other musical genres, so the melody of the original files are hard to recognize
    b. These original files consist of complicated instrumentation. While capturing the eigensounds through PCA, an instrument is very likely to be mixed up with another instrument with the same sound wavelength or frequency

Modeling

In this project, we apply various algorithms in an analogous fashion to that of image classification to our sample chunks of music. Our model then classifies the randomly-chosen test samples according to the entire music sample from which the test samples were extracted.

We selected six different models to investigate various approaches to classification: from linear and density-based models to ensemble and neural networks. We performed a repeated 5-Fold cross-validation for the following models: Gaussian Naïve Bayes (NB), Support Vector Machine (SVM), k-Nearest

Neighbor (*k*-NN), Random Forest (RF), Linear Discriminant Analysis (LDA), and Multi-Layer Perceptron (MLP).

We then chose the best models in terms of accuracy and time and evaluated them against a more optimized MLP *NN*. This step highlights how *nonNN* models can potentially perform this approach to music classification with comparable accuracy to *NN* at a fraction of processing cost.

In addition to model selection and comparison to a neural network model, we also want to explore how the results change when we change the number of dimensions in PCA. Here we can see how a reduction in the number of PCs increases the overall cross-validated accuracy of our models by reducing the overfitting and 'noise' in our data:

Averages of 5 Repetitions

5-folds Cross-Validated

	NB w/ 30 PCA	**NB w/ 5 PCA**
Accuracy	90.57% (±1.38%)	95.24% (±0.68%)
Time	13.93ms	11.79ms
	NB w/ 25 PCA	**NB w/ 3 PCA**
Accuracy	91.43% (±1.22%)	96.14% (±0.72%)
Time	13.78ms	11.86ms
	NB w/ 15 PCA	**NB w/ 2 PCA**
Accuracy	94.24% (±1.07%)	96.41% (±0.63%)
Time	13.17ms	10.98ms
	NB w/ 7 PCA	**NB w/ 1 PCA**
Accuracy	94.86% (±0.87%)	96.75% (±0.65%)
Time	13.11ms	11.44ms

Evaluation

First, we identify the most efficient models in terms of processing time-accuracy trade-offs. Then, we run all models for both before and after PCA.

	Without PCA		With Scaling and PCA	
	Averages of 2 Repetitions 5-folds Cross-Validated		Averages of 10 Repetitions 5-folds Cross-Validated	
NB				
Accuracy:	83.56%	(±1.21%)	96.67%	(±0.69%)
Time:	3.09ms		11.00ms	
SVM				
Accuracy:	87.08%	(±0.97%)	87.98%	(±0.87%)
Time:	162.55ms		194.31ms	
***k*NN**				
Accuracy:	65.02%	(±1.62%)	96.51%	(±0.75%)
Time:	1.37ms		10.81ms	
RF				
Accuracy:	83.59%	(±0.42%)	76.65%	(±2.23%)
Time:	597.83ms		154.93ms	
LDA				
Accuracy:	65.08%	(±1.68%)	70.05%	(±1.66%)
Time:	62.12ms		11.48ms	
MLP				
Accuracy:	72.12%	(±19.75%)	95.18%	(±4.80%)
Time:	2760.54ms		1861.52ms	

We can see that the *k*-NN model consistently has the fastest processing times. NB has the highest accuracy after PCA and moderately high accuracy without PCA. Although SVM has a high accuracy before PCA, the accuracy drops by 13% with PCA. Considering the volatility, we will pick k-NN and NB models to compare with the MLP neural network model.

Comparing the *k*-NN model and NB model with a more optimized MLP model (with comparable accuracy), we get the classification results as follows:

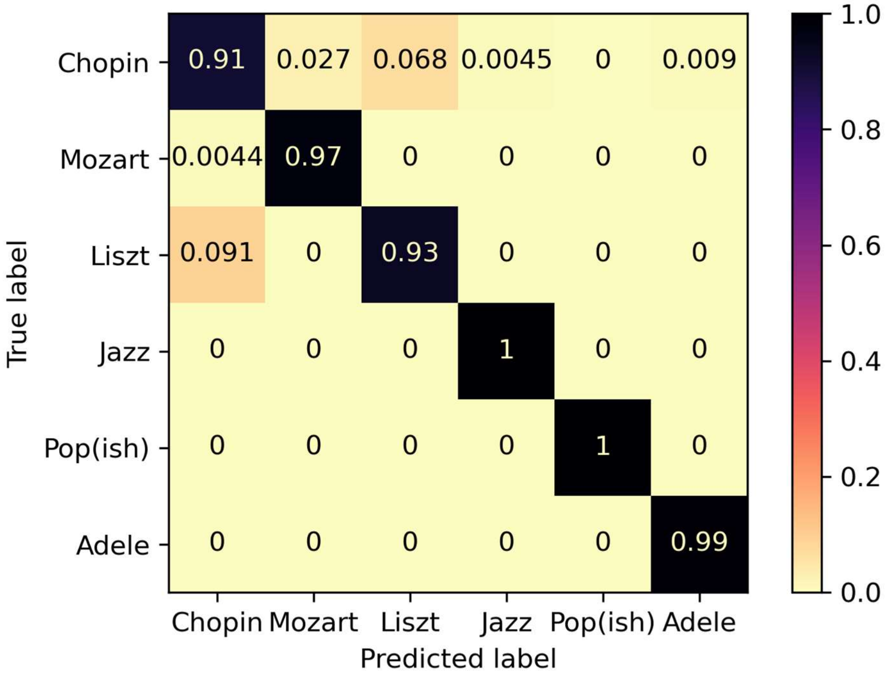

Figure 2: *k*-NN Model Confusion Matrix

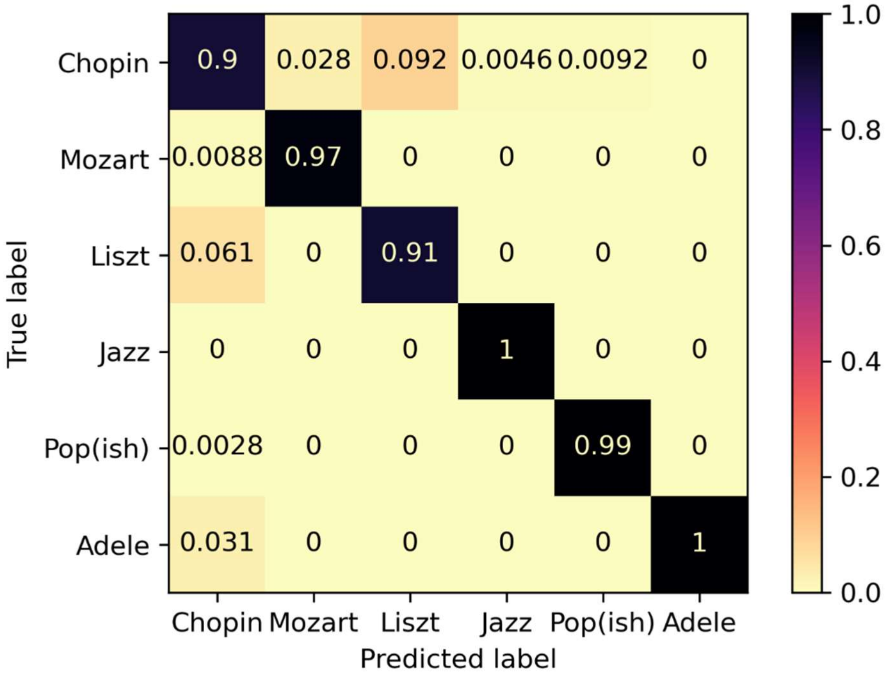

Figure 3: NB Model Confusion Matrix

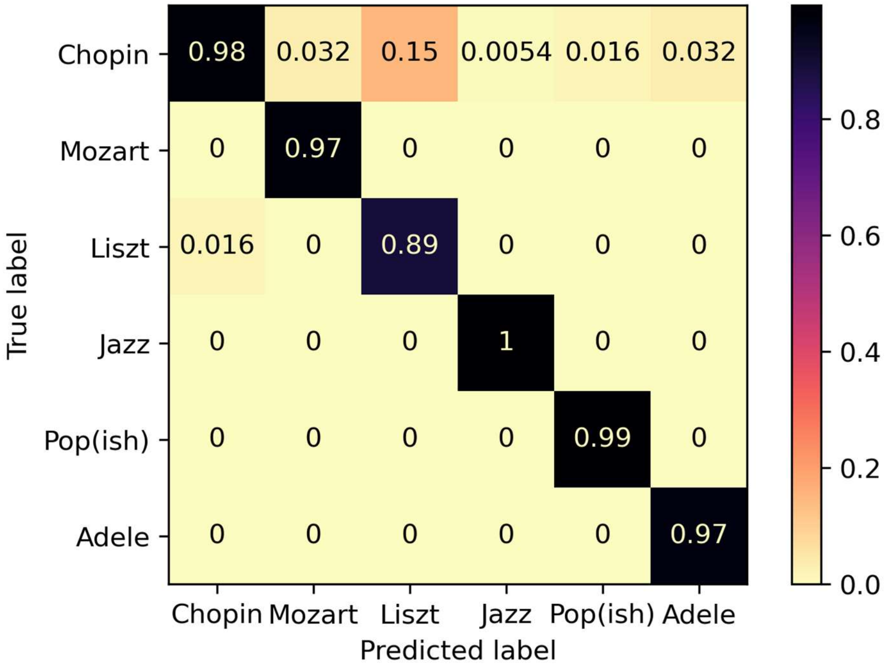

Figure 4: MLP Model Confusion Matrix

From the confusion matrix, we could see that the prediction accuracies for these three models are relatively the same. We then time the three models. NB took the shortest time of 1.36ms per run for 7,000 runs. In comparison, our partially optimized MLP model took 3.44s per run to complete 7 runs (~2530 times slower than NB). With such a high time-accuracy trade-off, we do not believe using an MLP model is justified in this scenario.

As the next step for exploration, we tested different PCA dimensions and ran the same results. We observed that as we reduce the PCA dimensions, the models become more accurate. This could be because a lower PCA dimension reduces the noise level of the dataset and reduces over-fitting in our models. As we discovered while listening to the eigensounds, only the loudest melodies or most prominent features are captured in eigensound.

Conclusions and Next Steps

In a nutshell, we could produce eigensound for musical files and use it to successfully classify test samples with near-perfect accuracy using a resolution sample rate of 50 Hz (a 99.9% reduction in file content) for 5 seconds. Listening to these eigensounds, our impression is that the eigendecomposition of the raw music arrays generally results in components that capture the loudest melodies of each sample. These results held whether with solo instrumental music, ensemble music, or even vocal ensembles with both traditional and electronic instruments.

Based on our results, we recommend using NB model for eigensound classifications over *NN* and other commonly used classification models. As a next step, we could implement our findings in a broader context:

1. Delve deeper into the effect of PCA dimensions on audio file classification
2. Apply the model further into song identification for songs in similar genres
3. Discover any applicable use cases in the realm of audio recognition

As we seek to dive deeper at capturing the unique style of a given musical artist (e.g. could we train a model using only Mozart's operas but still recognize his piano works in the test set), one area, in particular, we would like to explore is the efficacy of tensor decomposition methods versus the eigensound approach. An algorithm making use of Nonnegative Tucker Decomposition ("NTD") explored in Marmoret, et al. (2021) has shown promise in increased accuracy of segmentation analysis when used in combination with the β-distance and log-mel spectrograms. This might prove useful in extending our basic eigensound approach to more complex problems in the MSA problem space.

Contributions

We collaborated consistently throughout the final six weeks of this course, ensuring equal contribution during this project with frequent group calls at each stage of the progress.

References

Haunschmid, Verena, et al. “AudioLIME: Listenable Explanations Using Source Separation.” ArXiv:2008.00582 [Cs, Eess], Sept. 2020. arXiv.org, http://arxiv.org/abs/2008.00582.

Kirkbride, Ryan, and Kia Ng. Infinite Remix Machine: Automatic Analysis and Arrangement of Musical Recordings. 2015. DOI.org (Crossref), https://doi.org/10.14236/ewic/eva2015.61.

Marmoret, Axel, et al. "Nonnegative Tucker Decomposition with Beta-Divergence for Music Structure Analysis of Audio Signals." ArXiv:2110.14434 [Cs, Eess, Math], Nov. 2021. arXiv.org, http://arxiv.org/abs/2110.14434.

Müller, Meinard. Fundamentals of Music Processing. 1st ed., Springer Berlin Heidelberg, 2015.

Nasrullah, Zain, and Yue Zhao. "Music Artist Classification with Convolutional Recurrent Neural Networks." 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8. DOI.org (Crossref), https://doi.org/10.1109/IJCNN.2019.8851988.

Nieto, Oriol, et al. "Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications." Transactions of the International Society for Music Information Retrieval, vol. 3, no. 1, Dec. 2020, pp. 246–63. DOI.org (Crossref), https://doi.org/10.5334/tismir.54.