Using Machine Learning to Predict Classical Composers from Audio

Nishanth Joshi
Monta Vista High School

Abstract

This research paper explores the application of machine learning techniques to predict classical composers from audio recordings. Classical music, with its rich history and diverse styles, poses a challenge in identifying composers solely based on musical characteristics. The study utilizes a dataset of 2000 Western classical tracks from different eras and employs artificial neural networks for feature engineering. The goal was to develop an accurate predictive model that lists potential composers for a given piece. The results indicate that the LSTM model achieves moderate accuracy, correctly identifying the true composer within the top three predictions. This research is an important contribution to the field as it will further demonstrate the utility of using machine learning to predict the composer of a piece of classical music. It displays the possibilities of using machine learning for other music-related data. It also contributes to the development of future tools that are helpful for both passive music enjoyers as well as professional musicians.

Introduction

Background

Classical music, spanning over four centuries, has captivated audiences with its rich history and diverse compositions. Throughout the years, composers have brought forth their unique artistic voices, creating works that showcase their individual styles and expressions. The ability to differentiate between compositions of different composers has been a skill appreciated by music enthusiasts and scholars alike. However, accurately identifying the composer of a classical piece based solely on musical characteristics remains a challenging task. Here, we aim to address this challenge by leveraging the power of machine learning techniques to predict the likely composer of a classical music piece.

Research Objective

The primary objective of this study is to develop a predictive model that can accurately identify the composer of a classical music piece based on extracted audio features. By harnessing the capabilities of artificial neural networks, we aim to engineer features that effectively capture the distinctive elements and nuances of different composers' styles. These features will serve as the foundation for training and evaluating the predictive model. The comprehensive dataset utilized in this study consists of Western classical tracks spanning various eras and composers, providing a diverse and representative collection of compositions for analysis.

Classifying classical music pieces based on the composer is a challenging task due to the inherent complexity and variations within classical music. Unlike genres such as pop or rock, where artists often exhibit consistent styles and characteristics, classical composers often venture into diverse musical territories, exploring different forms, moods, and techniques. Furthermore, each time a piece is performed, it represents a unique interpretation of the written music, making it even harder to predict. Therefore, accurately predicting the composer of a classical music piece requires a deep understanding of the intricacies of each composer's style and the ability to extract meaningful patterns and features. To achieve this, machine learning techniques provide a powerful toolset. By training a model on a large dataset of classical music pieces, the model can learn the underlying patterns and correlations between the features and the composers. Through this process, the model can develop an understanding of the distinctive elements that define each composer's style, enabling it to make accurate predictions on unseen music samples.

The findings of this study not only shed light on the potential of machine learning in the realm of classical music, but also demonstrate the broader possibilities of using machine learning for music-related tasks. The outcomes of this research can pave the way for future developments in automated music analysis, benefiting both passive music enthusiasts and professional musicians alike. In the following sections of this paper, we delve into the methodology employed to tackle this research objective. We discuss the dataset used, the process of feature extraction, and the model architecture utilized for composer prediction. Subsequently, we present the results obtained from the model and provide a comprehensive discussion of the findings. Finally, we conclude with insights into the implications of this research and its potential impact on the field of classical music analysis and beyond.

Methodology

Dataset

The research utilizes a dataset provided by AudioSets Erlangen (AudioSets Erlangen, n.d.), consisting of 2000 Western classical tracks. The dataset serves as the foundation for training and evaluating the composer prediction model. It is carefully curated to represent a wide range of composers and musical styles, covering various eras from the baroque period (17th century) to the modern era (21st century). Each track in the dataset is associated with the corresponding composer, providing the ground truth labels required for supervised learning. For this current model, only the orchestral tracks were used as training and testing data. Originally, the data set contained chords in the form "A_maj_min7" as well as the duration in seconds of the chord. We modified the data such that only the first 20 seconds were included from each piece. This choice was motivated by the need to maintain consistency across the dataset, as longer compositions often introduce significant variability in both structure and mood. By limiting the analysis to the first 20 seconds, we ensure that the model focuses on the most representative and consistent portion of each piece, which is often where the composer's stylistic signature is most apparent. Then we made it such that the time incremented by 0.1 seconds and split the chord into three parts, letter(A), triad(maj) and sevenths(min7). From there, we One-Hot Encoded all of the data so that we would be able to pass it into our model (Table 1).

Table 1. First 0.5 seconds of Bach sinfonia in g major wq.182 no.1 with the letters One-Hot Encoded

Piece	Time	letter_A	letter_Ab	letter_B	letter_Bb	letter_C	letter_C#	letter_D	letter_E	letter_Eb	letter_F	letter_F#	letter_G	letter_na
Bach	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bach	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Bach	0.1	0	0	0	0	0	0	0	0	0	0	0	1	0
Bach	0.2	0	0	0	0	0	0	0	0	0	0	1	0	0
Bach	0.3	0	0	0	0	0	0	0	0	0	0	1	0	0
Bach	0.4	0	0	0	0	0	0	0	0	0	0	1	0	0
Bach	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0

Example of a piece with the time and letter of the chord one One-Hot Encoded, example does not include triads and sevenths.

Model Architecture

The composer prediction model is constructed using a sequential LSTM (Long Short-Term Memory) architecture. LSTMs are a type of recurrent neural network (RNN) specifically designed to model sequential data. They excel at capturing long-term dependencies and patterns in time series data, making them well-suited for analyzing music.

In the model architecture, the LSTM layer serves as the core component responsible for learning the temporal dependencies in the input data. It processes the extracted audio features in a sequential manner, taking into account the ordering and timing of the musical elements. The LSTM layer's hidden units maintain an internal memory state, allowing the model to retain relevant information over extended time intervals. This memory mechanism enables the model to capture complex patterns present in classical music compositions.

The output of the LSTM layer is passed through a dense layer, also known as the classifier. The dense layer maps the learned representations from the LSTM layer to the predicted composer. The activation function used in the dense layer depends on the specific requirements of the problem. We used softmax activation which transforms the outputs of the previous layer into a probability distribution. Mathematically, the softmax function can be defined as follows:

$$\sigma_{zj} \qquad (z)_i = \frac{e^{z}i}{\sum_{j=1}^{k} e}$$

where (z)i is the input score for class and k is the total number of classes. In this equation, the numerator $e^{z}{}_{i}$ computes the exponential of the input score for class k ensuring that the resulting value is positive. The denominator $\sum e_j^{z}$ calculates j=1 the sum of the exponentiated scores for all classes. This normalization term ensures that the output probabilities sum up to 1. By dividing the exponentiated score of each class by the sum of all exponentiated scores, softmax guarantees that the resulting values represent valid probabilities. The higher the input score for a particular class, the larger the corresponding softmax probability will be. In the context of multi-class classification, the softmax activation function is commonly used in the output layer of a neural network. In this specific application, the softmax function ensures that the model assigns the highest probability to the most likely composer while still accounting for the relative likelihood of other potential composers. This helps improve prediction accuracy by refining how the model differentiates between closely related composers.

The model's architecture and parameters, such as the number of LSTM units, the size of the hidden layers, and the dropout rates, are carefully chosen based on empirical experimentation. The model is compiled with an appropriate loss function, categorical cross-entropy, and an optimizer, Adam, to facilitate gradient-based optimization during the training process. In multi-class classification, the output of the model is a probability distribution across multiple classes. Each class is assigned a probability value, indicating the model's confidence in that class being the correct prediction. Categorical cross-entropy measures the dissimilarity between this predicted probability distribution and the true class labels. The categorical cross-entropy function encourages the model to assign higher probabilities to the true classes and lower probabilities to the other classes. It penalizes large errors in the predicted probabilities and helps the model converge towards more accurate predictions during the training process. By using categorical cross-entropy, the model learns to make more confident and precise distinctions between composers, reducing misclassification errors and improving overall accuracy.

Adam, or Adaptive Moment Estimation, is an optimization algorithm commonly used to update the parameters of a neural network during training. It is an extension of the gradient descent optimization algorithm that incorporates adaptive learning rates for each parameter, allowing for faster and more efficient convergence. The adaptive learning rates provided by Adam allow each parameter to have a different update magnitude based on the magnitude of its gradient and the history of past gradients. This helps the algorithm converge faster by making larger updates for parameters with sparse gradients and smaller updates for parameters with frequent updates. Adam also includes a bias correction mechanism to address the fact that the estimates of the moments are biased towards zero at the beginning of training when the number of updates is small. During the training phase, the model learns to minimize the prediction error by iteratively adjusting its weights using backpropagation and gradient descent. The training dataset is presented to the model in batches, and the model's parameters are updated based on the gradients computed from the batch. The training process continues for multiple epochs until convergence or a predefined stopping criterion is reached. By employing Adam as the optimizer, the model benefits from stable and adaptive learning rates, which enhances its ability to capture nuanced differences between composers, ultimately leading to improved prediction accuracy.

Training and Evaluation

The model is trained using the extracted features from the training dataset and their corresponding composer labels. The training process involves iteratively presenting batches of training examples to the model and updating its weights based on the computed gradients. For our model, we used a batch size of 12. This process enables the model to learn the underlying patterns and relationships in the data, allowing it to make accurate predictions about the composers based on the input audio features. To assess the model's performance and its ability to generalize to unseen data, it is evaluated using a separate testing dataset. The testing dataset consists of pieces that were not used during the training phase, ensuring an unbiased evaluation. The model provides a list of predicted composers for each piece, along with their respective probabilities. The top three predicted composers and their probabilities are reported for each piece. The model predicts the composers for the test examples based on their extracted features, and the predictions are compared against the ground truth composer labels to calculate various evaluation metrics, such as accuracy and precision. These metrics provide insights into the model's predictive performance, its strengths, and potential limitations.

Results

The trained model was evaluated on a separate testing dataset to assess its predictive performance. The evaluation yielded a test loss of 1.196 and a test accuracy of 50.4%. Despite the modest accuracy, the model demonstrated encouraging indications of its capability to identify the composer of a given piece. Notably, the true composer was part of the top three predicted composers in over half of the cases.

Upon closer examination, two outliers emerged, namely composers 1 and 7, with lower performance in the top three predictions, achieving 40% inclusion and only 10% as the top prediction (Table 2). Further investigation revealed that both composers were often predicted as composer 12, with the latter being included in the top three predictions 90% and 100% of the time, respectively, and selected as the top composer in 60% and 40% of the instances (Table 3). This intriguing observation suggests a potential similarity in composing styles between composers 1, 7, and 12, although it is more likely attributed to the limited data available for composers 1 and 7, each with only 10 pieces imputed to the model. Excluding composers 1 and 7, the model demonstrated favorable accuracy in predicting the correct composer, with a top three prediction rate of 82% and a top composer prediction rate of 46%.

Table 2. Composer by Composer breakdown of the prediction results displaying how often the correct composer was predicted in the top three, the top and the average prediction %

	Top 3 Composer	Top Composer		Average Prediction
Composer Name	Predicted	Predicted	Total Pieces	%

1	40%	10%	10	31.71%
2	90%	48%	40	31.00%
3	78%	72%	60	78.33%
4	83%	38%	40	26.35%
5	85%	10%	40	22.96%
6	90%	70%	10	33.63%
7	40%	10%	10	23.30%
8	78%	52%	60	36.89%
9	78%	54%	50	54.95%
10	88%	35%	40	36.14%
	Top 3 Composer	Top Composer	Average Prediction	Composer Name
Predicted	Predicted	Total Pieces	%	
11	90%	70%	10	72.92%
12	80%	35%	20	45.80%
13	50%	20%	10	28.58%

TABLE 3. Composer by Composer breakdown of the prediction results displaying how often the most predicted composer was predicted in the top three, the top and the average prediction %

Composer Names	Most predicted Composer	Top 3 Composer Predicted	Top Composer Predicted	Average Prediction %
1	12	90%	60%	35.76%
2	2	90%	48%	31.00%

3	3	78%	72%	78.33%
4	4	83%	38%	26.35%
5	5	85%	10%	22.96%
6	6	90%	70%	33.63%
7	12	100%	40%	28.50%
8	8	78%	52%	36.89%
9	9	78%	54%	54.95%
10	10	88%	35%	36.14%
11	11	90%	70%	72.92%
12	12	80%	35%	45.80%
13	13	50%	20%	28.58%

Discussion

The evaluation of our machine learning-based model for predicting classical composers from audio recordings has provided valuable insights into its performance and limitations. In this section, we discuss the implications of the evaluation results, address the model's strengths and weaknesses, and propose potential avenues for further improvement and research.

Implications of Evaluation Results

The evaluation results shed light on the model's ability to accurately predict classical composers based on audio features. The model demonstrated promising generalization capabilities across different compositions and composers, indicating its potential to capture the distinctive elements of each composer's style effectively. This suggests that the feature extraction process, employing denoising auto-encoders, has been successful in enhancing the representation of relevant musical elements.

The model architecture, utilizing a sequential LSTM (Long Short-Term Memory) model, is a suitable choice for capturing temporal dependencies within the input data. LSTMs are well-suited for analyzing sequences and have been successful in various natural language processing and time series analysis tasks. However, the performance of the model may benefit from further experimentation with different architectures, such as incorporating additional layers or exploring alternative recurrent neural network (RNN) variants. However, when another LSTM layer was added, the accuracy of the model dropped to 10.3%. This is most likely due to overfitting by the model.

Model Limitations and Challenges

The accuracy limitations observed in the model can be primarily attributed to the intrinsic complexity and variation present in classical compositions. Classical music is known for its rich diversity of styles, spanning from the Baroque era to the Modern era. Each composer's unique artistic voice, characterized by their experimentation with musical forms, structures, harmonies, and melodic patterns, poses a significant challenge for capturing their distinct style solely based on audio features. Another potential factor influencing model accuracy is dataset imbalance. Since certain composers are underrepresented in the dataset, the model may struggle to learn their distinctive musical characteristics effectively, leading to biased predictions favoring more frequently occurring composers.. The overfitting observed when adding an additional LSTM layer highlights the importance of carefully balancing model complexity to prevent such issues. The performance drop suggests that deeper architectures may lead to overfitting, emphasizing the need for cautious consideration of model design choices. To mitigate overfitting, techniques such as dropout regularization can be employed, where a fraction of LSTM units is randomly ignored during training to prevent reliance on specific neurons. Hyperparameter tuning, such as adjusting the number of LSTM units or optimizing the learning rate, can also help find an optimal balance between model complexity and performance.

Dataset Size and Diversity

The dataset used for training and evaluation, consisting of 2000 Western classical tracks from various eras and genres, provided a comprehensive representation of classical music. However, the size and diversity of the dataset are crucial factors influencing the model's performance. While the dataset covered a broad range of composers and compositions, incorporating data from specific composers or musical periods could potentially improve the model's ability to capture the unique characteristics of individual composers. As noted in the results section additional pieces for composers with a smaller data set could also help increase the overall accuracy of the model.

Conclusion

This research demonstrates the utility of machine learning techniques, particularly the use of artificial neural networks, in predicting classical composers from audio recordings. The subjective nature of music appreciation and the complexity of artistic expression make establishing definitive rules for composer identification challenging. The limitations and challenges encountered in accurately capturing the distinct style of each composer in classical music highlight the complexity of the task. Further improvements in feature engineering techniques, model architecture, and dataset size and diversity are essential for enhancing the model's performance. However, despite these challenges, the results of this research indicate promising potential for composer prediction. The ability to identify the true composer within the top three predictions showcases the effectiveness of the model. Machine learning models serve as tools to assist and augment human understanding rather than replace it entirely.

This research contributes to the broader understanding of machine learning applications in music analysis and offers insights for future developments in automated music analysis and composer identification. By leveraging advanced techniques and comprehensive datasets, machine learning can play a crucial role in the appreciation and analysis of classical music. Both casual listeners and professional musicians can benefit from the application of machine learning models in composer prediction. For instance, this model could enhance music recommendation systems by helping streaming platforms suggest pieces similar in style to a listener's preferences based on composer classification. Additionally, in historical musicology, it could aid in the authentication of disputed compositions by analyzing stylistic patterns. Educational tools could also integrate this model to help music students recognize and study different composers' stylistic traits, providing interactive learning experiences that deepen their understanding of classical music composition. By combining machine learning techniques with human expertise, we can enhance our understanding and appreciation of classical music, making it accessible to a wider audience and facilitating in-depth analysis for researchers and musicians alike.

References

International Audio Laboratories Erlangen. (n.d.). Cross-Era Dataset [Dataset]. https://www.audiolabs-erlangen.de/resources/MIR/cross-era