Multimodal Genre Prediction

Israel Anjorin, Tiantian Li, and Ellie Paek

{israel.anjorin,tiantian.li,ellie.paek}@emory.edu

Abstract. In an age where digital music consumption is prevalent worldwide, the importance of music recommendation has grown exponentially, particularly for users to discover music that aligns with their tastes. However, with elements of globalized genres being used for production, it is difficult to pinpoint what exact genre tracks may be a part of.
Machine learning helps facilitate such identification, utilizing musical data such as the key of the song, the lyrics, and extracted audio features such as danceability ratio, to assist with the task. Genre Classification is a multi-class, supervised machine learning task in the interdisciplinary field of Music Information Retrieval (MIR). Common use cases of Genre Classification include music recommendation systems, automatic audio recognition, and music generation.
In this work, we curated data from multiple sources and merged into a dataset with 7,299 tracks, including the basic information (track name, artist, album name), lyrics, MIDI file, and genre of each track. Using this dataset, we show that simple machine learning models such as multi-layer perceptron and random forest trained with multi-modal data are able to achieve high performance in genre prediction task.

Introduction

Automatic music genre prediction is an important service that has burgeoned alongside growth of music content distribution and has played an important role the attraction of customers[16].In an era dominated by digital music consumption worldwide, the significance of music recommendation has surged, especially for users seeking tunes that resonate with their preferences. Yet, as many contemporary songs draw from globalized genres, pinpointing the exact genre of a track has become a challenge.

To address this challenge, machine learning plays a pivotal role by leveraging musical data, such as the song's key and danceability ratio. Genre Classification emerges as a multi-class, supervised machine learning task within the interdisciplinary field of Music Information Retrieval (MIR). Genre Classification finds applications in diverse areas like music recommendation systems, automatic audio recognition, and music generation.

This paper proposes a multi-modal algorithm for automatic genre prediction, incorporating not only meta-data from music tracks but also the lyrics and MIDI files associated with each song. Since lyrics and MIDI files exist in formats unsuitable for direct input, we convert lyrics into embedded vectors and

employ an audio feature extractor to transform the MIDI file into a suitable representation. The model's performance is compared across random forest and neural networks, assessing whether the combination of two or three modes offers comparable performance to a single mode. Notably, our work demonstrates that incorporating new modes enhances the accuracy of genre predictions, presenting a novel approach, particularly in the evaluation of genres with MIDI files.

The contribution of this paper includes:

1. We published a dataset containing the metadata, lyrics, audio features, and MIDI files for 7,229 songs that can be used for future Genre classification studies.
2. We evaluated and reported the performance of two machine learning models on Genre Classification. We compared the performance of models trained with different data (i.e. data from different modalities).

Background and Related Work

The solution to automating genre classification has been reviewed with machine learning strategies, albeit usually with one type of data. Previous work in genre classification predominantly depended on the audio of songs themselves, typically extracting audio signals to use as the primary features. [2,12,16]. Others have attempted generating spectrograms to output songs into images, and classifying genres in that manner. [12]

Much of single-modal Genre Classification has attempted to use lyrics and natural language processing (NLP) as well. Embedding techniques are used to transform the lyrics into word vectors, which are then utilized as features for various traditional and neural network models. [9,15] While leveraging lyrics has yielded comparable, if not superior, results compared to previous works relying solely on audio-based features, the achieved accuracy, in the mid-to-high 60s, indicates that further refinement and the incorporation of additional modes are imperative for advancing the precision of genre classification models [9].

Recently, however, there have been studies that leveraged multi-source data. An example of this is a model that performed Genre Classification with lyrics as one of its core features. By using chords based off of ukulele or guitar tabs, the model generated joint-embeddings between the two features — lyrics and chords — to predict genre labels [5]. Research has also demonstrated potential of multi-source Genre Classification through audio, visual, and text representations of a given track. [11]

Most previous studies on Genre Classification exploited the dataset GTZAN [16], which was published in 2002 for the studying music signal processing[1,2]. GTZAN contains 1,000 songs from 10 different genres, including blues, disco, hiphop, etc. Despite its wide usage in Genre Classification, GTZAN dataset only contains audio signal data, missing other aspects of a song that contains rich information such as lyrics. Additionally, the GTZAN dataset may have little representative power of genres that emerged or gained popularity in the past 20 years such as indie and RnB. Additionally, considering genre itself is subjective

to the society, the manually annotated labels back in 2002 may no longer be as appropriate and accurate now.

Approach

Data

We will use the dataset from Ens and Pasquier, a collection of MIDI files and its respective meta data [3]. In total, there are 436,631 MIDI files, with each file accompanied by the following two features: the artist and title, and the genres associated with the MIDI file. In addition, the Spotify ID associated with each MIDI file along with its correlation score is included.

Data Accumulation We use the Spotify IDs associated with the MIDI files to scrape data using the Spotify API. For each Spotify ID and MIDI file associated with it, we accumulate the track name, artist, lyrics, popularity (ranged from 0 to 100), the details of the albums it is from, danceability, energy, estimated overall key, overall loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration, and the language of the lyrics. In total, we scrape 136,543 tracks with the associated features.

Data Filtering Despite the accumulation, we noticed that many of the associated tracks were duplicates of one song, many of which, despite being versions of the same song, did not have lyrics provided with the track. To make sure that that data accumulated was consistent throughout all of the features, we filtered out tracks that were missing lyrics and were duplicates. We also noticed that some of the lyrics were not English, and to ensure consistency of the quality of data, chose to remove tracks with non-English lyrics as well. Figure 1 demonstrates the pipeline followed to produce our final dataset. The songs that were associated with the eight most common genres were selected (shown in Table 1), resulting in a dataset of 7,229 tracks, each with their associated metadata, English lyrics, and related MIDI file.

Lyric Transformation

Various approaches have been proposed to represent text data in machine learning models such as pooling word embeddings [14], using the CLS token from BERT [8], or more recently Sentence-BERT, a BERT-based Siamese network architecture [13]. While our model is compatible with any embeddings, in our experiments we used the `all-MiniLM-L6-v2` model, which is a light-weight pretrained Sentence-BERT model. While `all-MiniLM-L6-v2` accepts both lists of sentences and paragraphs, each lyrics was inputted into the transformer as a string acting as a paragraph due to the inconsistency of the number of verses between all tracks.

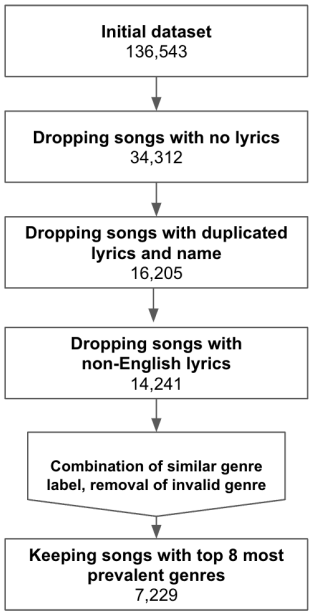

Fig. 1. Inclusion criteria for the final dataset containing the meta data of each track, the lyrics (if found), the genre (target label), and MIDI file. Table 1 reports the top 8 most prevalent genres and the corresponding number of tracks

Genre	Number of tracks
Pop	2,744
Rock	2,260
RnB-soul	516
Country	474
Dance-Electric	414
Alternative-Indie	352
Hip-Hop-Rap	253
Punk	216

Table 1. Top 8 most prevalent genres and the number of tracks that falls under each genre in the final dataset

In addition to utilizing `all-MiniLM-L6-v2`, we experimented with whether further preprocessing assisted with producing better results. In an iteration of the data, all of the tracks' lyrics were extracted. Each track's lyrics had stopwords removed and was lemmatized before being inputted into the sentence transformer to generate embeddings. Table 6 demonstrate the difference in results when the lyrics were preprocessed and not preprocessed.

MIDI Transformation

Most existing studies that uses MIDI files as input documents aim to build generative models for music pieces. Main methodologies for representing MIDI files are generating spectrograms [6], event representations for notes and chords[6,7], and graph representations [10]. For the purpose of this study, we adopted the encoder model in [6] which slices MIDI file into 5-second frames, transforms each frame into note events and encode with indices identified in [4]. On average each track in our dataset contains 40 frames, we average pool over the middle 10 frames. The representation generated are sparse, 2048-dimension vectors, where, in most cases, the last 1600 dimensions are trailing zeros. Given the size of our dataset, we take the first 256 dimension as the final representation.

Model Selection

For our models, we decided to use a multi-layer perceptron (MLP), as well as random forest to help classify the genres. Multi-layer perception has fairly robust computational capabilities and an ability to handle large numbers of input features. Considering the large number of dimensions that our data has with all three modes, MLP was the best choice to capture such large dimensions.

Random forest, like MLP, has an advantage of being able to handle a large number of input features, which fits well with the large dimensions of our dataset. In addition, it is relatively robust to errors and outliers compared to other models. As the data we have accumulated may be diverse with a number of outliers, we hypothesized that using random forest would be advantageous to producing robust results.

Hyperparameter Tuning As the space of hyperparameter tuning was fairly large for both Random Forest and MLP, we used a grid search cross validation method to find the best hyperparameters. A 3-fold cross validation was used, and the accuracy was used as the validation metric. As running the search on both MLP and Random Forest was relatively expensive with a large search space, we narrowed our search space to reduce the cost of training them. The hyperparameters we tuned for each model are listed below in Figure 2.

```
Random_Forest_params =
   {
   'n_estimators' : [100,200], 'max_depth' : [3,5,7],
   'min_samples_leaf' : [50, 100]
   }

MLP_params =
   {
   'hidden_layer_sizes' : [(100,80,60), (150,120,100), (150,100)],
   'alpha' : [1e-3, 1e-5]
   }
```

Fig. 2. Parameter search space for Random Forest Classifier and MLP Classifier

Evaluation Strategy

To keep consistency between previous works and ours, as well as evaluate if a multimodal model performs better than single modal models, we trained seven models, each with an input of different number of modes and different mode types. Table 2 highlights which model used which data, as well as the final number of dimensions of the given input data.

For metrics, we measured the accuracy of the correctness of labels, as well as the precision, recall, and f_1 score of the labels. Specifically, the macro precision, recall, and f_1 score were used to give equal weight to each of the features used to predict the genre. Each track had a single label of genre associated with it, and thus the scores were calculated with whether the prediction via the model was correct compared to the given label.

Results

Table 2 reports the performance, measured in precision, recall, f_1, and accuracy of MLP and Random Forest on different datasets. The highest value (best performance) per column is bolded. MLP model trained on data from all three modalities shows best or second to best, performance in three of the metrics, while MLP trained on metadata and MIDI has the highest accuracy and MLP on MIDI alone has the highest recall. Our experiment shows that comparing to baseline models that were trained with only features from one modality, increasing the number of modalities help boost performance in most cases. Additionally, while numerous studies focused on using metadata, MIDI and lyrics showed superior performance in all metrics. The Random Forest model, trained using metadata and MIDI, excelled in the top two positions across all four metrics. Despite the superiority of models trained with data from two modalities over those trained with just one, the model incorporating all three modalities did not

achieve the highest performance. Comparing between the two, MLP models also achieved significantly higher performance than Random Forest.

Figure 3 shows the confusion matrix of the MLP model trained with all features. The model has the best performance in predicting `alternative-indie` and `rnb-soul`, while it made more mistakes in distinguishing between `rock` and `pop`.

Model	Metadata	Lyrics	MIDI	dimension	precision	recall	f1	accuracy
MLP	✓			11	0.651	0.615	0.628	0.672
		✓		384	0.849	0.815	0.827	0.838
			✓	256	0.920	**0.914**	0.916	0.920
	✓	✓		395	0.878	0.846	0.859	0.867
		✓	✓	640	0.909	0.894	0.900	0.903
	✓		✓	267	0.924	0.913	0.918	**0.925**
	✓	✓	✓	651	**0.938**	0.909	**0.923**	0.924
Random Forest	✓			11	0.237	0.153	0.129	0.456
		✓		384	0.554	0.176	0.166	0.484
			✓	256	0.686	0.300	0.348	0.583
	✓	✓		395	0.557	0.169	0.154	0.478
		✓	✓	640	**0.793**	0.275	0.319	0.557
	✓		✓	267	0.685	**0.319**	**0.370**	**0.591**
	✓	✓	✓	651	0.685	0.277	0.317	0.567

Table 2. Metric results on test set for MLP and Random Forest trained with different datasets. Highest value (best performance) each column is bolded.

position of frame	precision	recall	f1	accuracy
First 10 frames	0.918	0.912	0.914	0.913
Middle 10 frames	0.920	0.914	0.916	0.920
Last 10 frames	0.898	0.909	0.903	0.907

Table 3. A study of the impact of the position of audio frames taken in a track on MLP trained with only audio features.

Ablation Studies We performed experiments to determine the effectiveness of several approaches mentioned previously. For MIDI transformation, we studied the effect of different number of frames taken, different output dimensions, and different positions of frames taken. For lyrics transformation, we studied the effect of pre-processing. Table 3 reports the performance of MLP model trained with only MIDI feature, with frames taken from different position in a track.

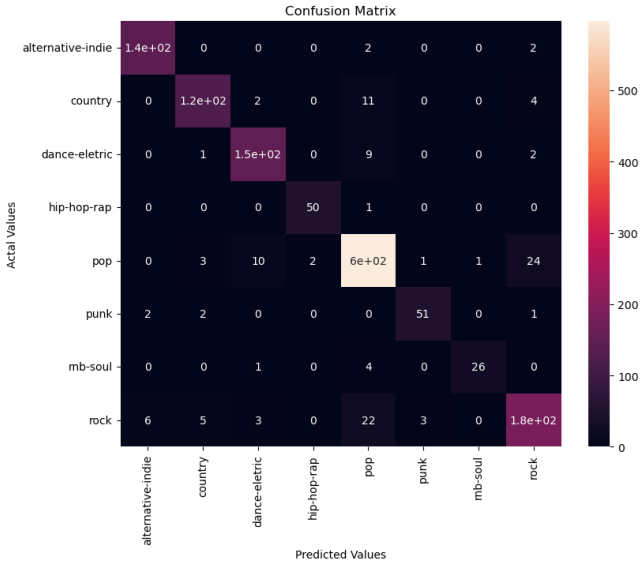

Fig. 3. Confusion matrix of MLP model trained with all three modalities

vector length	precision	recall	f1	accuracy
256	0.920	0.914	0.916	0.920
384	0.915	0.916	0.915	0.924
512	0.931	0.907	0.918	0.919

Table 4. A study of the impact of the output vector length for MIDI processor on MLP trained with only audio features.

Number of frames	precision	recall	f1	accuracy
10	0.920	0.914	0.916	0.920
15	0.931	0.907	0.918	0.919

Table 5. A study of the impact of the number of frames taken from a track for average pooling on MLP trained with only audio features

Preprocessing on text	precision	recall	f1	accuracy
yes	0.849	0.815	0.827	0.838
no	0.831	0.819	0.823	0.834

Table 6. A study of the impact of preprocessing techniques on lyrics on the performance of MLP trained with only lyrics features

Taking frames from the beginning and middle of a track produced similar results, while taking frames from the end resulted in slight degradation in results. Table 4 reports the effect of different vector length. Comparing across different vector lengths, we notice that there is no significant drop in performance in truncating more dimensions. Given the training dataset and resource available for training, 256 dimension is the most suitable for the purpose of our experiment.

Table 5reports the effect of different number of frames used for average pooling. While increasing the number of frames used has slight boost in precision and f_1 score, accuracy and recall experienced some kind of degradation. Table 6 compares the performance between MLP model trained with original and preprocessed lyrics. From the table, observe that preprocessing leads to superior results in precision, f_1, and accuracy.

Discussion

In general, our results show that MLP had better performance compared to random forest, with all of the metrics in MLP surpassing that of random forest in all of the modes. This suggests that Genre Classification operates as a nonlinear problem, one in which MLP is more prone to solving accurately. Moreover, although MLP outperformed random forest, its variance as a consequence may be higher, leading to possible overfitting issues if new data is introduced.

In addition, there was a general trend in which adding different modes produced better results. However, it is interesting to note that some of the best performance metrics seem to come from the one or two-mode models rather than the three-mode model. Notably, such models utilize MIDI vectors as their input, indicating that employing the MIDI files to predict genres may enhance accuracy in genre classification.

Examining the results of the metadata in particular, one observation that stands out is that models using just the metadata resulted in the lowest scores and least accurate results. Given this, it may be possible that the numerical values of the metadata are highly correlated or less relevant to the genre of the track. As a result, utilizing it as an additional mode may have potentially reduced the results of the multiple modes, particularly when comparing the metrics between the 3-mode random forest model with its comparable models.

Considering the limited dataset size, potential future directions may involve exploring more efficient methods of merging embeddings from different sources, such as mapping representations to a shared dimension. In addition, examining the metadata features for any correlation and possible dimensionality reduction may also help improve results.

Overall, however, the trend in improvement with the increase of modes suggests that utilizing tracks' associated MIDI files, along with their metadata and lyrics as a multimodal model demonstrates state-of-the-art results in Genre Classification. As improvements are made to the dataset and additional modes are explored, the models may continue to showcase enhanced capability across a broader range of genres and tracks.

Contribution

This project was operated by an overall well-distributed workload, ensuring a cohesive effort among all of the team members. Each member actively participated in running the models and compiling the overall results, as well as contributing to the overall data collection process. Ellie handled the textual aspect of the data processing, focusing on converting the lyrics into vectors with sentence transformer models. Tiantian and Israel undertook the task of transforming MIDI data files into viable vectors, exploring different routes to accurately represent the audio before choosing to use an encoder model.

Code and Dataset

The code for this project can be found on Github.

References

1. Elbir, A., Aydin, N.: Music genre classification and music recommendation by using deep learning. Electronics Letters **56**(12), 627–629 (2020). https://doi.org/https://doi.org/10.1049/el.2019.4202, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2019.4202
2. Elbir, A., Bilal Çam, H., Emre Iyican, M., Öztürk, B., Aydin, N.: Music genre classification and recommendation by using machine learning techniques. In: 2018 Innovations in Intelligent Systems and Applications Conference (ASYU). pp. 1–5 (2018). https://doi.org/10.1109/ASYU.2018.8554016
3. Ens, J., Pasquier, P.: Building the MetaMIDI Dataset: Linking Symbolic and Audio Musical Data. In: Proceedings of the 22nd International Society for Music Information Retrieval Conference. pp. 182–188. ISMIR, Online (Nov 2021). https://doi.org/10.5281/zenodo.5624567, https://doi.org/10.5281/zenodo.5624567
4. Gardner, J.P., Simon, I., Manilow, E., Hawthorne, C., Engel, J.: MT3: Multi-task multitrack music transcription. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=iMSjopcOn0p
5. Greer, T., Narayanan, S.: Using Shared Vector Representations of Words and Chords in Music for Genre Classification. In: Proc. Workshop on Speech, Music and Mind (SMM 2019). pp. 46–50 (2019). https://doi.org/10.21437/SMM.2019-10
6. Hawthorne, C., Simon, I., Roberts, A., Zeghidour, N., Gardner, J., Manilow, E., Engel, J.: Multi-instrument music synthesis with spectrogram diffusion (2022)
7. Huang, Y.S., Yang, Y.H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. p. 1180–1188. MM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394171.3413671, https://doi.org/10.1145/3394171.3413671
8. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
9. Leszczynski, M., Boonyanit, A.: Music genre classification using song lyrics (2021), https://api.semanticscholar.org/CorpusID:235344286

10. Lisena, P., Laaksonen, J., Troncy, R.: Understanding Videos with Face Recognition: A Complete Pipeline and Applications. Multimedia Systems **28**(6), 2147–2159 (Dec 2022). https://doi.org/10.1007/s00530-022-00959-x
11. Oramas, S., Barbieri, F., Nieto, O., Serra, X.: Multimodal deep learning for music genre classification. Transactions of the International Society for Music Information Retrieval (Sep 2018). https://doi.org/10.5334/tismir.10
12. Pelchat, N., Gelowitz, C.M.: Neural network music genre classification. Canadian Journal of Electrical and Computer Engineering **43**(3), 170–173 (2020). https://doi.org/10.1109/CJECE.2020.2970144
13. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992 (2019)
14. Shen, D., Wang, G., Wang, W., Min, M.R., Su, Q., Zhang, Y., Li, C., Henao, R., Carin, L.: Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 440–450 (2018)
15. Tsaptsinos, A.: Lyrics-based music genre classification using a hierarchical attention network (2017)
16. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on speech and audio processing **10**(5), 293–302 (2002)