Novel Approach to Music Genre Classification using Clustering Augmented Learning Method (CALM)

Soumya Suvra Ghosal and Indranil Sarkar
National Institute of Technology Durgapur
India
{soumyasuvraghosal@gmail.com,indranil.sarkar.nitdgp@gmail.com}

Abstract

This paper proposes an automatic music genre-classification system using a deep learning model. The proposed model leverages Convolutional Neural Nets(CNN) to extract local features and LSTM Sequence to Sequence Autoencoders to learn representations of time series data by taking into account their temporal dynamics. The paper also introduces Clustering Augmented Learning Method (CALM) classifier which is based on the concept of simultaneous heterogeneous clustering and classification to learn deep feature representations of the features obtained from LSTM autoencoder.
Computational Experiments using GTZAN dataset resulted in an overall test accuracy of 95.4% with a precision of 91.87%.

Introduction

With the increasing amount of music available online, there is automatically a growing demand for the symmetrical organization of audio files and that has increased the interest in music classification. To detect a group of the music of a similar genre is the main work of the recommendation system and playlist generators. Thus building a robust music classifier using machine learning techniques is essential to automate tagging unlabeled music and improve user's experience of media players and music libraries. In recent years, convolutional neural networks(CNNs) have brought revolutionary changes to the computer vision community. Meanwhile, CNN's have been widely used for music information retrieval, especially music genre classification. Recently, it became increasingly popular to combine CNNs with recurrent networks(RNNs) to process audio signals, which introduce time-sequential information to the model. In convolutional recurrent networks(C-RNNs), the CNN component is used to extract features while RNN plays the role of summarizing temporal features. The inputs of C-RNNs are soundtrack spectrograms and outputs are probabilities of each genre at each timestep. Inspired by previous literature, we propose to leverage the idea by augmenting LSTM autoencoder with CNN and use a Clustering-based classifier to predict the genre of music.

Previous Works

Music genre classification has been actively studied since the early days. Tzanetakis and Cook [Tzanetakis and Cook2002] used k-nearest neighbor classifier and Gaussian Mixture models with a comprehensive set of features for music classification. Those features could be summarized into three categories: rhythm, pitch, and temporal structure. Zhouyu Fu [Fu et al.2010] proposed a Naive Bayes (NB) classifier framework, namely NB Nearest Neighbor (NBNN) and NB Support Vector Machine (NBSVM) for music genre classification. [Deshpande and Singh2001] compared k-nearest neighbor, Gaussian Mixtures and SVM to classify music into three genres which are rock, piano, and jazz. In recent years, using an audio spectrogram has become mainstream for music genre classification. Spectrograms encode time and frequency information of given music as a whole. Spectrograms can be considered as images and used to train convolutional neural networks (CNNs) ([Wyse2017]). [Li, Chan, and A2010] developed a CNN to predict the music genre using the raw Mel Frequency cepstral coefficients(MFCCs) as input.
In this paper, we aim to combine convolutional nets with LSTM Autoencoders to extract both spatial and temporal features of the audio signal. Instead of baseline classifiers, we propose a clustering-based classification model. In the proposed classification approach we cluster the data based on their inherent characteristics and in the process of learning the best clustering solution we optimize the hyperparameters of the classification model, thereby substantially improving the learning process. We used the mel-spectrogram as the only feature and compared the proposed model with traditional classifiers and previous literature.

Dataset and Representation

Dataset In the paper, we have used the GTZAN dataset. It contains 10 music genres, each genre has 100 audio clips in .au format. The genres are - blues, classical, country, disco, hip-hop, pop, jazz, reggae, rock, metal. Each audio clips has

a length 30 seconds, are 22050Hz Mono 16-bit files. The dataset incorporates samples from a variety of sources like CDs, radios, microphone recordings, etc. The training, testing and validating sets are randomly partitioned following proportion 8:1:1.

Features A popular representation of sound is the spectrogram which captures both time and frequency information. In this study, we used the Mel spectrogram as the only input to train our neural model. A mel spectrogram is a spectrogram transformed to have frequencies in the mel scale, which is logarithmic, more naturally representing how human senses different sound frequencies. To convert raw audio to Mel spectrogram, one must apply Short Time Fourier Transforms(STFT) across sliding windows of audio, around 20ms wide.

In this case, the music features are extracted using the `LibROSA` library in Python using 128 mel filters, frame length of 2048 samples and a hop size of 1024. We got a spectrogram of size 647×128.

Proposed architecture and methodology

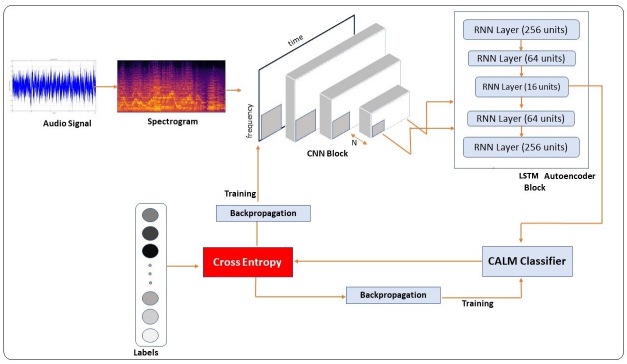

Figure 1: Model Architecture

Architecture

The model consists of a four-layer convolutional neural network (CNN) which is followed by an LSTM Sequence to Sequence Autoencoder(AE) and ultimately consists of the proposed CALM classifier. Not only to make the network unconstrained of any handcrafted features, but the convolutional layers are also used to extract meaningful and useful features from the song. The output of the CNN is a sequence in which every timestep strongly relies on both the immediate predecessors and long term structure of the entire song. To capture both transient and overall characteristics, we use LSTM Sequence to Sequence Autoencoder and for classification, we propose CALM, which is explained in the following sections. The assumption underlying this model is that the temporal pattern can be aggregated better with LSTM Autoencoders than CNNs while relying on CNNs on input side for local feature extraction.

The CNN architecture consists of 4 convolutional layers of 64 feature maps, 3-by-3 convolution kernels and max-pooling layers of dimensions (2×2)-(3×3)-(4×4)-(4×4). In all convolutions, we pad zeros to each side of the input to keep size fixed. Dropout(0.5) is applied to all convolutional layers to increases generalization. The CNN output has a feature map size of N×1×15 (number of feature maps×frequency×time). For extracting temporal pattern we use an LSTM-based architecture. The architecture uses LSTM layers having $\{256,64,16\}$ units as the encoder LSTM_{enc} and LSTM layers having $\{64,256\}$ units as the decoder LSTM_{dec}.

Methodology To start, features are extracted from the spectrogram using convolutional layers. The output of Convolutional Neural Networks is fed to an LSTM Seq to Seq Autoencoder which collects key information about the temporal properties of the input sequence in its hidden state. The final hidden state of the LSTM_{enc} is then passed through some layers, the output of which is used to initialize the hidden state of the LSTM_{dec}. The function of the LSTM_{dec} is to reconstruct the input sequence based on the information contained in its initial hidden state. The network is trained to minimize the root mean squared error between the input sequence and the reconstruction. Once the training is complete, the activation of the fully connected encoded layer is used as representations of the audio sequence and is fed as input to Clustering Augmented Learning Method Classifier. This system showed 98% accuracy at the end of the training.

Clustering Augmented Learning Method (CALM)

Proposed Approach

Input augmentation As in [Ghosal et al.2019], we consider a matrix of input data D and a set of cluster centers C. Since in this case study, there are 10 music genres, we keep C as 10. In this paper, we use clustering to augment input data $x \in D$ for better learning. To augment the input data, we add a new set of features representing either an input example belongs to a cluster or not. To distinguish input examples, we introduce an additional index $h \in \{1, \ldots, |D|\}$ representing the number of an input example (x_1 is the first input example of D). We define also a vector c_h composed of c_{hl}, $l \in C$ for each example $x_h \in D$. It is a one-hot representation containing zeros except for the index of the cluster it belongs to (e.g. $c_1 = [0,0,0,1,0,0,0,0,0,0]$ means that the first input example x_1 belongs to the 4th cluster out of 10 clusters). Finally, we augment input examples by concatenating the vector x_h with the vector c_h for each $h \in \{1, \ldots, |D|\}$.

Cluster centers To determine the cluster centers, CALM consists of a clustering model and a Feed-Forward Neural Net(FNN) having a softmax output to classify the music genres. For the clustering model, we propose to use a Random Forest classifier to determine cluster centers. After the FNN is trained using a state-of-the-art solver for data belonging to a single cluster $\in \{1, \ldots, |C|\}$, a Random Forest Classifier is used to find the best cluster center. Hence we repeat $|C|$ instances of training the FNN to find the $|C|$ centers. For any instance l of the model, we use the one-hot encoded vector of l as labels for all the input sample in that cluster. In simple words, while predicting center of 4th cluster (for example) we use $[0,0,0,1,0,0,0,0,0,0]$ as label for

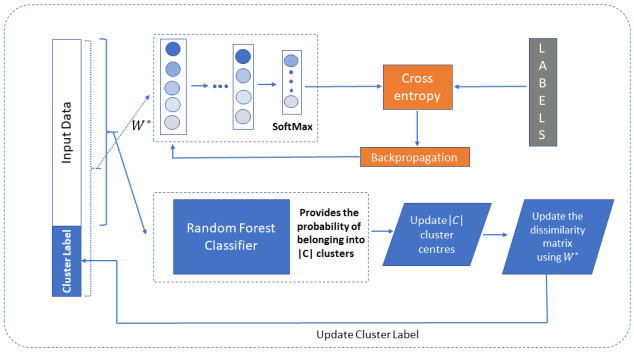

Figure 2: Architecture of Clustering Augmented Learning Method(CALM) Classifier

all input samples, since $|C|$ is 10.

We propose that the input sample which has the lowest error in predicting its cluster label is considered as the center of that cluster in the subsequent iteration of the proposed approach. In such a manner, the center would be the input sample which is the most fitting representative of that cluster. As a result, the clustering process would aggregate the data having similar characteristics resulting in better learning by the FNN classification model.

Clustering Problem

We have a distance/dissimilarity measure d_{il} between input examples $i \in D$ and cluster centers $l \in C$. The clustering problem aims to assign each input example to a cluster such that the total distance between the elements of a cluster and its center is minimized.

In this paper we also propose a novel dissimilarity measure based on the weights of the trained FNN classifier. It uses the average of weights linked to each neuron of the input layer. Assuming that the original input (without the new clustering feature) has d dimensions ($x_h = [x_h^1, \ldots, x_h^d], h \in \{1, \ldots, |D|\}$) and the weight linking node n of the input layer to node $j \in \{1, \ldots, n_1\}$ of the following layer is w_j^n, the two distances measures are formulated as follows:

$$d_{il} = \sum_{n \in \{1 \ldots d\}} \underset{j \in \{1, \ldots, n_1\}}{\text{avg}} w_j^n |x_i^k - x_l^k|$$

Thus the distance measure computes the distance between two examples based on how important is the contribution of each input feature to the resulting prediction. Therefore, the resulting clusters contain examples with similar potential to improve the classification results.

Proposed Algorithm

We propose an approach (Algorithm 1) where we iteratively train the FNN classifier, use its weights for input data clustering thus changing the input vector, train again the FNN classifier using the new input data, and so on until a stopping criterion is attained. The stopping criterion is triggered if the cluster assignment remains the same for consecutive 10 iterations, i.e., the clustering problem converges.

The configuration of the proposed model is given as:

A) Convolutional Neural Network(CNN): It is used to extract local features from the input . $\texttt{CNN1} \xrightarrow[\text{filter size} = 2\times2]{\text{Maxpool}} \texttt{CNN2} \xrightarrow[\text{filter size} = 3\times3]{\text{Maxpool}} \texttt{CNN3} \xrightarrow[\text{filter size} = 4\times4]{\text{Maxpool}} \texttt{CNN4} \xrightarrow[\text{filter size} = 4\times4]{\text{Maxpool}} \texttt{LSTM AE}$. Dimension of all convolution kernel is 3×3.

B) LSTM Autoencoder : It is used to aggregate temporal features `LSTM` Layer1 $\longrightarrow$ `LSTM` Layer2 $\longrightarrow$ `LSTM` Layer3 (Embedded layer) $\longrightarrow$ `LSTM` Layer4 $\longrightarrow$ `LSTM` Layer5. Dimension of the LSTM Layers are {256, 64, 16, 64, 256} respectively.

C) Classification Model: `FC1` $\longrightarrow$ Leaky ReLU $\longrightarrow$ `FC2` $\longrightarrow$ Leaky ReLU $\longrightarrow$`FC3` $\longrightarrow$ Softmax . Dimension of `FC1`: 128. Dimension of `FC2`: 32. Dimension of `FC3`: 10.

D) Optimizer: ADAM Learning Rate 0.001, momentum rate 0.9, weight decay(L2 regularization):1e-4.

Algorithm 1: Clustering-augmented learning method

Step 0: Data obtained after extracting the local features using CNN and temporal information using LSTM sequence to sequence Autoencoder acts as input to CALM.

Step 1: Initialization of the cluster centers $u_1, u_2, ..., u_{|C|}$ randomly. Clustering of the output data obtained from LSTM autoencoder and augmenting each data sample with its one-hot encoded cluster label.

Step 2: Training the FNN & clustering model

foreach $l \in \{1 \ldots |C|\}$ **do**
- Train the FNN model on data belonging to cluster l to learn classification.
- For supervised training of the random forest classifier we use one hot encoded representation of clusters as labels. Running the clustering model gives the cluster center u_l.

Step 3: Clustering

Update dissimilarity matrix using W^*

if *stopping criterion is attained* **then** Stop.
else go to **Step 2**.

Results and Discussions

The proposed model is trained by ADAM [Kingma and Ba2014] for 150 epochs or early stop [Prechelt1998] if no improvement in 25 epochs. The performance of all networks are evaluated using Precision, Recall, and Accuracy which are defined as :

$$\textbf{Precision} = \frac{N_C}{N_C + N_F}$$
$$\textbf{Recall} = \frac{N_C}{N_C + N_M}$$
$$\textbf{Accuracy} = \frac{total_c}{total_m}$$

where N_C is the number of accurately predicted music tracks, N_F is the number of falsely predicted music tracks, N_M is the number of missed music tracks, $total_c$ is the

number of all accurately predicted music tracks and $total_m$ is the number of all music tracks.
To further interpret the results, we plotted the confusion matrix(Table 3) of the proposed model. Looking more closely at our confusion matrix, we see that our proposed model managed to correctly classify 80% of rock audio as rock, labeling the others as mainly country or blues. Additionally, it incorrectly classified some country, as well as a small fraction of blues and reggae, as rock music.

Comparison with Baseline Classifiers We trained four traditional classification models on the dataset as baseline classifiers, including k-nearest neighbors, logistic regression, random forest, multilayer perceptrons, and linear support vector machine, using Mel Frequency Cepstral Coefficients(MFCCs) by flattening them into a 1-D array. Apart from baseline classifiers we also experimented by stacking a Logistic Regression classifier with the features obtained from Convolutional Net and LSTM Autoencoder to test the performance of the CALM classifier. As evident from Table 1, CALM outperforms the Logistic Regression classifier when augmented with Convolutional nets and LSTM autoencoder. Moreover, Fig.4 shows how the intra-cluster variance decreases after approximately 75 iterations and then stabilizes. To measure intra-cluster variance, we used Euclidean distance in this case study. Similarly, it is evident from Fig. 3 that testing loss starts decreasing after 80 epochs and gradually as the clustering solution converges, the accuracy begins to improve. This observation bolsters our initial assumption that clustering data based on inherent characteristics would improve the learning process of FNN.
For a fair comparison, all models are trained and tested on the same dataset as the proposed model. The hyperparameters are tuned by a grid search to ensure that the best model configuration is adapted. In table 2 we have also compared our model with relevant literature and it is evident that proposed architecture performs strongly.

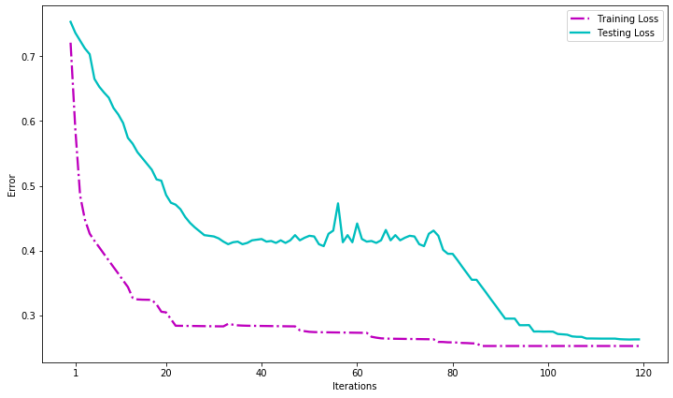

Figure 3: Training and Testing Loss

Conclusion

In this paper, we present a specially designed network for accurately recognizing the music genre. The proposed model aims to take full advantage of low-level information of Mel-spectrogram for making the classification decision. We have shown how our model is effective by comparing the state-of-art methods, including both hands crafted feature approaches and deep learning models. In this work, we use the GTZAN dataset which is a common benchmark dataset. Our proposed model has achieved an impressive accuracy of 95.4% while testing, which outperforms all other models. In the future, we will try to improve the model by improvising some new distance metric methods to compute the similarity between genres.

Table 1: Performance of models

Model	Train Accuracy	Test Accuracy
CNN + LSTM AE + CALM (Proposed Model)	0.98	**0.954**
CNN + LSTM AE + Logistic Regression	0.914	0.873
Logistic Regression	1.0	0.77
k- Nearest Neighbours	1.0	0.36
Multilayer Perceptron	0.9725	0.83
Support Vector Machine	1.0	0.28
Random Forest	1.0	0.76

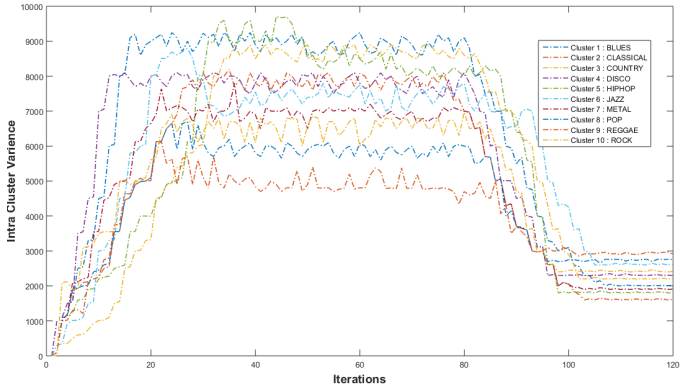

Figure 4: Plot of Intra-cluster Variance vs Iterations

Table 2: Comparison with Literature

Models	Accuracy
Proposed Model	**0.954**
Liu *et al.* [Liu et al.2019]	0.939
Multi-DNN [Dai et al.2015]	0.934
CVAF [Nanni et al.2017]	0.909
Hybrid Model [Karunakaran and Arya2018]	0.883
NNet2 [Zhang et al.2016]	0.874
Bergstra *et al.* [Matityaho and Furst2006]	0.825

Table 3: Confusion Matrix

Actual \ Predicted	Blues	Classical	Country	Disco	HipHop	Jazz	Metal	Pop	Reggae	Rock	Recall
Blues	95	0	0	0	0	1	0	0	1	3	95%
Classical	0	95	0	0	0	1	0	0	1	1	97.9%
Country	1	1	82	1	0	0	0	0	0	6	90.1%
Disco	0	0	3	75	1	0	1	0	1	3	89.3%
Hiphop	0	0	0	0	72	0	0	3	6	4	84.7%
Jazz	1	2	0	0	0	77	0	0	1	1	93.9%
Metal	0	0	0	3	0	0	64	0	0	3	91.4%
Pop	0	0	0	0	1	0	1	75	2	1	94.9%
Reggae	1	0	0	0	1	0	0	3	76	3	90.4%
Rock	2	0	7	3	0	1	1	0	1	85	85%
Precision	95%	96.9%	89.1%	91.5%	96%	96.3%	95.5%	92.6%	86.4%	79.4%	91.24% / 91.87%

References

[Dai et al.2015] Dai, J.; Liu, W.; Dong, L.; and Yang, H. 2015. Multilingual deep neural network for music genre classification. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[Deshpande and Singh2001] Deshpande, H., and Singh, R. 2001. Classification of music signals in the visual domain.

[Fu et al.2010] Fu, Z.; Lu, G.; Ting, K. M.; and Zhang, D. 2010. Learning naive bayes classifiers for music classification and retrieval. In *2010 International Conference on Pattern Recognition*.

[Ghosal et al.2019] Ghosal, S. S.; Bani, A.; Amrouss, A.; and El Hallaoui, I. 2019. A deep learning approach to predict parking occupancy using cluster augmented learning method. In *2019 International Conference on Data Mining Workshops (ICDMW)*, 581–586.

[Karunakaran and Arya2018] Karunakaran, N., and Arya, A. 2018. A scalable hybrid classifier for music genre classification using machine learning concepts and spark. In *International Conference on Intelligent Autonomous Systems (ICoIAS)*, 128–135. IEEE.

[Kingma and Ba2014] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Li, Chan, and A2010] Li, T. L.; Chan, A. B.; and A, C. 2010. Automatic musical pattern feature extraction using convolutional neural network. In *2015 Data Mining and Applications*. IEEE.

[Liu et al.2019] Liu, C.; Feng, L.; Wang, H.; and Liu, S. 2019. Bottom-up broadcast neural network for music genre classification. *arXiv preprint arXiv:1901.08928v1*.

[Matityaho and Furst2006] Matityaho, B., and Furst, M. 2006. Aggregate features and adaboost for music classification. *Machine Learning* 473–484.

[Nanni et al.2017] Nanni, L.; Costa, Y. M.; Lucio, D. R.; Silla Jr, C. N.; and Brahnam, S. 2017. Combining visual and acoustic features for audio classification tasks. *Pattern Recognition Letters* 49–56.

[Prechelt1998] Prechelt, L. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer. 55–69.

[Tzanetakis and Cook2002] Tzanetakis, G., and Cook, P. 2002. Musical genre classification of audio signal. *IEEE Transactions on Speech, and Audio Processing* 10(3):293–302.

[Wyse2017] Wyse, L. 2017. Audio spectrogram representations for processing with convolutional neural networks. *arXiv preprint arXiv: 1706.09559*.

[Zhang et al.2016] Zhang, W.; Lei, W.; Xu, X.; and Xing, X. 2016. Improved music genre classifi- cation with convolutional neural networks. *INTERSPEECH* 3304–3308.