Music Genre Classification via Machine Learning
Category: Audio and Music

Li Guo(liguo94), Zhiwei Gu(zhiweig), Tianchi Liu(kitliu5)

Abstract— **Many music listeners create playlists based on genre, leaving potential applications such as playlist recommendation and management. Despite previous study on music genre classification with machine learning approaches, there is still room to delve into and build sophisticated models for Music Information Retrieval (MIR) problems. In this work, we apply a variety of machine learning techniques on the recently published FMA dataset to classify 16 music genres given input features from music tracks, raising classification accuracy by more than 30% compared to the previously proposed baseline model.**

I. Introduction

Music genre is a key feature of any song that can guide users to their preferred category. Many music enthusiasts create playlists based on specific genres, leading to potential applications such as playlist recommendation and management. Even though there have been previous studies on music genre classification with machine learning approaches, in which various algorithms have been implemented and produced promising results, there is still room for improving performance of genre classifier. In this work, a music genre classification system is established based on various machine learning techniques. The goal of this work is that this genre classifier can be used to correctly classify a new music track given its associated features.

In this report, we will present the efforts we made towards building classification methods allowing us to identify a specific genre from audio features. We will first make an introduction to the dataset we use, and the way we dealt with the data. Then, we tried support vector machine and softmax regression with optimized paramters to train our dataset. These two models are treated as baseline of this study. Additional techniques such as Neural Network and K-nearest neighbors are also attempted afterwards. Based on classification accuracy of each model, error analysis is performed and future work is proposed

The open dataset Free Music Archive (FMA) is used for building the music genre classification system and implementing further calibration on the system such as error analysis. It includes 106,574 tracks of music arranged in a hierarchical taxonomy of 16 genres. Each track contains 518 attributes categorized in nine audio features. These attributes are obtained by data preprocessing of FMA music tracks using Python package LibROSA.

II. Related Work

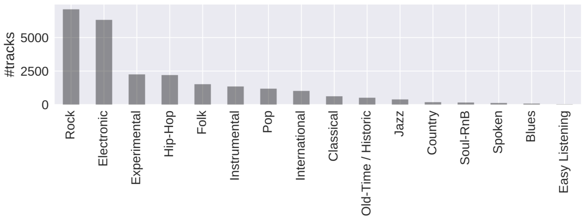
In MIR research community, there are various studies on establishing effective models for music genre classification. For example, Using MFCCs has become a popular way to approach this problem. I. Karpov[1] implemented the delta and acceleration values of the MFCCs, increasing the amount of information that can be collected from the data.

There are other common methods that can be used to classify music, as demonstrated by previous studies in [2], that were not used in this project such as the Octave-Based Spectral Contrast (OSC) or Octave-Based Modulation Spectral Contrast (OMSC).

III. Preliminary Experiments

A. Gathering Data and Data Visualization

The Free Music Archive (FMA)[3] includes 106,574 untrimmed tracks of 30s(.mps files) music, split into 16 genres(Hip-Hop, Pop, Rock, Experimental, Folk, Jazz, Electronic, Spoken, International, Soul-RnB, Blues, Country, Classical, Old-Time / Historic, Instrumental, Easy Listening), and each track contains 518 attributes categorized in nine audio features. Following is the distribution for genres.



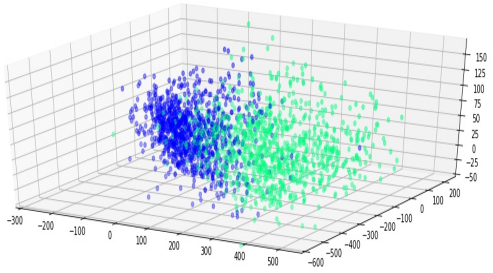
We used LibROSA(a Python package for music and audio analysis) to convert raw data and extract main features from the FMA dataset, and obtain audio features provided by Echonest (now Spotify) for a subset of 13,129 tracks to obtain our coefficients.

Our main features(level 1) include: chroma-cens, chroma-cqt, chroma-stft, mfcc, rmse, spectral-bandwidth, spectral-centroid, spectral-contrast, spectral-rolloff, tonnetz, zcr; we have 518 features in total with 3 levels. Following is the distribution of genres per track.

B. Data Preprocessing

25000 of all 106,574 tracks were used in this project for computational efficiency and information integrity. These 25000 tracks were splitted into training set, validation set and test set with sizes of 19922, 2505, 2573, respectively, and all training data was shuffled randomly. Therefore, training examples were represented as a large matrix of 19922 rows and 519 columns, with 518 features and a label of the genre.

Since most of the algorithms that we used usually treat vectors as inputs, we applied either PCA to our matrix or flattened the matrix to an extremely large vector, and then used this structure as a training example. Below is a visualization of a subset of our dataset with only two genres (Instrumental, Hip-Hop) after applying PCA to reduce the input to three dimensions:



IV. Methods

We built two baseline models: Support vector machine with linear kernel and softmax regression, with the following performance:

Classifier	Train accuracy	Test accuracy
SVM Linear	0.8021	0.5908
Softmax	0.5287	0.5103

With the performance of baseline models, several additional models were established to capture the relationship between features and genres of music: support vector machine, logistic regression, k-nearest neighbors and Neural Network. For SVM approach, second order polynomial kernel and radial basis function kernel were also implemented. Each model was trained by using 19922 training examples and k-fold cross validation with k = 5.

A. Model Selection and Regularization

In order to overcome complexity and overfitting, sequential forward feature selection and regularization were performed for each model. The SVM is L1-regularized with L2-loss. The C parameter found for the linear kernel was much smaller than for the other two kernels, indicating that the linear kernel is less able to perfectly separate the classes, which is one of the results that was found.

Feature selection was also performed using 19922 training examples and k-fold validation with k=5. This was a necessary step, as we have 518 features, which is a large size and many of those features were selected intuitively and needed to be screened for usefulness as not to simply introduce extra features for overfitting. The following figure shows an example of model selection process for the RBF SVM.

Model selection produced desirable results where only a subset of all features were required to minimize classification errors. Because of this, the overall complexity of the problem can be reduced.

B. Support Vector Machine

Support Vector Machines constructs a decision boundary using the input dataset so that the minimal distance from data points to the decision boundary is maximized. In addition to performing classification with a linear decision boundary, we also utilized kernel trick: radial basis function kernel, to perform non-linear decision boundary.

In our project we implemented SVM with linear kernel and radial basis function kernel. We run stochastic gradient descent to minimize the hinge loss function, then output a hypothesis function to make predictions.

For SVM with linear kernel, we implemented both L1-regularization and L2-regularization. Then we get an optimal regularization parameter $C = 0.01$ in L2 regularization, with 300 features, where C is the penalty parameter of error term relative to regularization term. Similarly, we optimize RBF kernel SVM with $C = 1.5$ and 275 features selected from forward model selection.

We can see that the C parameter of linear kernel is much smaller than that of RBF kernels, indicating better classification results for SVM with RBF kernels.

C. *Logistic Regression*

In Logistic Regression, we use sigmoid function as hypothesis function. Then we maximize log-likelihood by gradient descent to fit parameters.

Here we also implemented model selection and regularization with $k = 350$ and $C = 0.09$ to get the optimal prediction for logistic regression.

In this multiclass case, we conducted Logistic regression through the one-vs-rest (OvR) scheme, and uses the cross entropy loss.

D. *Softmax Regression*

Softmax regression is a generalization of logistic regression and is commonly applied to solve multilevel classification problem. The probability of an observation being in a specific class is

$$p(y = i|x; \theta) = \phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)}$$

for a total of k possible classes.

In this project, we implement Softmax as baseline model.

E. *KNN with PCA or Model Selection*

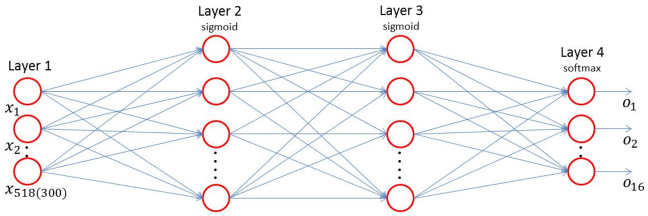
K-nearest neighbors is a simple non-parametric classification technique. The number of neighbors, k, controls model flexibility and adjusts the bias-variance tradeoff.

We implemented both principal component transformation and model selection for KNN model, and model selection outerperforms PCA in this situation.

F. *Neural Network*

In our Neural Network model, the input layer read in k features selected through forward model selection method. We tuned the number of hidden layers to be 2 to achieve best performance, both of which use sigmoid activation functions, with 320 and 32 hidden units respectively. The activation function for output layer is the softmax function, which gives a probability distribution for multiple genre labels. We set mini batch size to be 200, and trained the model iteratively.

The structure of our Neural Network model is as following:



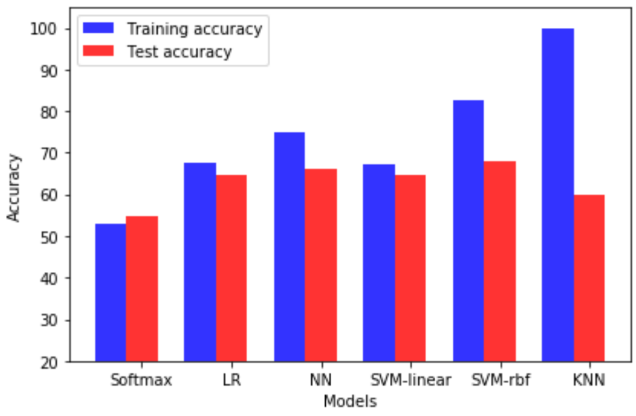
V. Results and Discussion

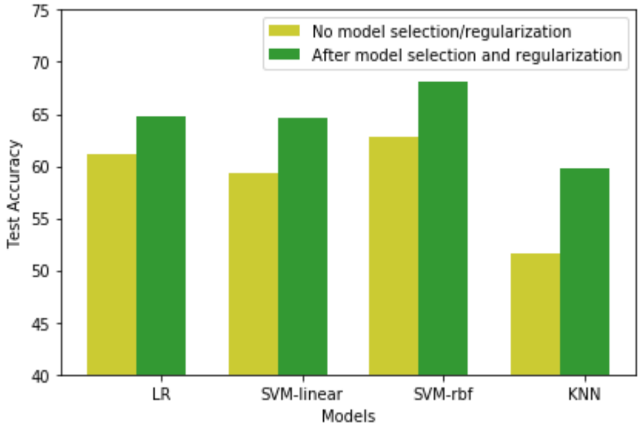
Each of the models was evaluated using the same training set of 19922 examples, dev set of 2505 examples and test set of 2573 examples.

Feature selection and regularization are conducted for each model. The training accuracy and test accuracy are shown in following figure and table.

Genre Classification Accuracy

Models	Train(%)	Dev(%)	Test(%)
Softmax	52.87	51.03	54.61
Logistic Reg.	67.45	62.61	64.75
Neural Network	74.93	63.19	66.03
SVM Linear	67.38	61.46	64.55
SVM RBF	82.61	64.32	68.07
KNN	99.98	57.87	59.71





Overall, the classification accuracy shown in above table are encouraging. SVM model with RBF kernel has the highest test accuracy. However, considering its higher training accuracy, there is still certain level of overfitting, even though feature selection and regularization are implemented.

The models mentioned above were first fitted using all 518 features without any regularization or feature selection schemes, and test accuracies obtained were lower than their corresponding values reported in above Table. For example, direct implementation of SVM with RBF kernel resulted in test accuracy of only 62.88%, which is lower than model after model selection and regularization for about 6 percent.

It is also likely that using all 518 features gives multicollinearity issue, or features could be correlated just by chance. It also leads to overfitting to training data in each model, which results in high variance.

Overfitting and multicollinearity issues were suppressed by using either L1/L2 regularization (shrinkage) and forward model selection, and these two methods both improved test accuracy of proposed models. As shown in above table, regularized SVM-RBF with 275 preselected features give test accuracies of 68.07%.

Features selected with forward model selection revealed that the original 518 features seem to contain redundant information and are not all necessary for the genre classification. After feature selection, all models achieved a better performance in prediction, as shown in following figure.

We can see that after model selection, test accuracy increases for all models shown above, which is because model with too many features will result in overfitting and instability because of fitting too much on training data and the collinearity among different feature.

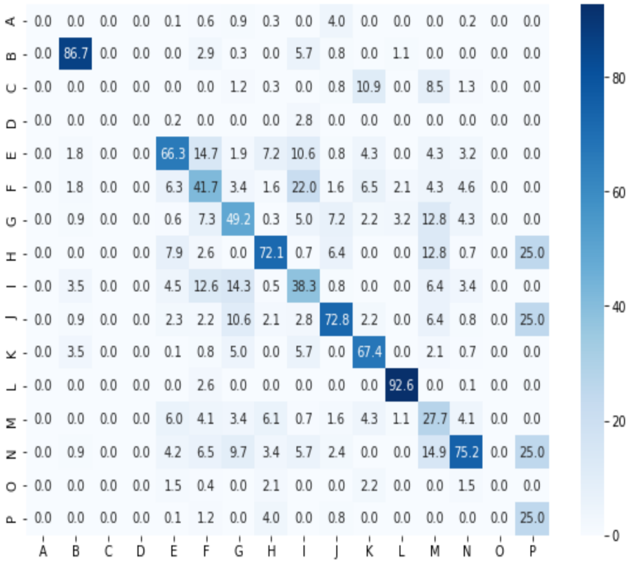
We further improve models' performance through regularization, including L1 and L2 regularization. For example, as shown below is the process of tuning penalty parameter to control the regularization for model SVM with rbf,

Softmax, linear kernel SVM, logistic regression, and KNN appear to have difficulties in capturing the non-linearities of the data, thus they achieve less accuracy than Neural network and rbf kernel SVM. Examinations of the data show that there is mixing of the classifications near the decision boundary that these models have trouble capturing.

We can also see that KNN classification technique gives a low test accuracy, compared to other models. This is because KNN classification suffers from the curse of dimensionality. That is, with increasing dimensionality, the volume of feature space rapidly increases and training examples become sparse. As a result, training and test accuracies significantly decreases. To suppress this effect, both principal components analysis and model selection ware applied in our project to reduce dimensionality of feature space. For KNN with PCA, we found that first 3 principal component can only explain 23.71% variance, which is too low for PCA to have a good performance, and such that KNN-PCA only obtains 54.13% test accuracy. As for model selection, we found that KNN model, with highest test accuracy, 59.71%, has 169 preselected features from best subset selection. Further shrinkage of feature space reduces test accuracy since additional relevant features are deleted, which increase bias of the model. Overall, KNN is outperformed by other classification methods such as SVM.

For the original neural network, the result of optimized model showed a relative high variance. This might because the amount of data needed to determine the weights of the network will increase as the number of weights in the networks increases, and consequently caused the issue of over-fitting. PCA was first applied for dimension reduction of input features, where it didn't show a better result, which might because the discriminative information that distinguishes one class from another might be in the low variance components on applying PCA. NN with model selection to 300 input features turned out to have a best result, increased our testing accuracy from 62.23% to 66.03% with less overfitting.

The confusion matrix of the SVM-RBF displayed below shows test prediction accuracy for each genre.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	0.0	0.0	0.0	0.0	0.1	0.6	0.9	0.3	0.0	4.0	0.0	0.0	0.0	0.2	0.0	0.0
B	0.0	86.7	0.0	0.0	0.0	2.9	0.3	0.0	5.7	0.8	0.0	1.1	0.0	0.0	0.0	0.0
C	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.3	0.0	0.8	10.9	0.0	8.5	1.3	0.0	0.0
D	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	2.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
E	0.0	1.8	0.0	0.0	66.3	14.7	1.9	7.2	10.6	0.8	4.3	0.0	4.3	3.2	0.0	0.0
F	0.0	1.8	0.0	0.0	6.3	41.7	3.4	1.6	22.0	1.6	6.5	2.1	4.3	4.6	0.0	0.0
G	0.0	0.9	0.0	0.0	0.6	7.3	49.2	0.3	5.0	7.2	2.2	3.2	12.8	4.3	0.0	0.0
H	0.0	0.0	0.0	0.0	7.9	2.6	0.0	72.1	0.7	6.4	0.0	0.0	12.8	0.7	0.0	25.0
I	0.0	3.5	0.0	0.0	4.5	12.6	14.3	0.5	38.3	0.8	0.0	0.0	6.4	3.4	0.0	0.0
J	0.0	0.9	0.0	0.0	2.3	2.2	10.6	2.1	2.8	72.8	2.2	0.0	6.4	0.8	0.0	25.0
K	0.0	3.5	0.0	0.0	0.1	0.8	5.0	0.0	5.7	0.0	67.4	0.0	2.1	0.7	0.0	0.0
L	0.0	0.0	0.0	0.0	0.0	2.6	0.0	0.0	0.0	0.0	0.0	92.6	0.0	0.1	0.0	0.0
M	0.0	0.0	0.0	0.0	6.0	4.1	3.4	6.1	0.7	1.6	4.3	1.1	27.7	4.1	0.0	0.0
N	0.0	0.9	0.0	0.0	4.2	6.5	9.7	3.4	5.7	2.4	0.0	0.0	14.9	75.2	0.0	25.0
O	0.0	0.0	0.0	0.0	1.5	0.4	0.0	2.1	0.0	0.0	2.2	0.0	0.0	1.5	0.0	0.0
P	0.0	0.0	0.0	0.0	0.1	1.2	0.0	4.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	25.0

For our purposes, we tend to reach as best performance as possible for each music genre. However, the distribution for each genre are highly skewed. As a result, examples were missed for several genres, as we can see in the confusion matrix. We can see higher test accuracy for genres B and L in confusion matrix, which corresponds to ”Rock” and ”Electronic”, which is because of larger percentage of dataset for these two genres.

Finally, the training error for KNN is actually leave-one-out cross validation. This is used because the distance weighting used is squared inverse and thus would always select the same training example and would result in a trivial training error of 0.

VI. Conclusion and Future Work

We implemented a variety of machine learning techniques to classify music genres using FMA dataset. The highest test accuracy is 68.07%, achieved by RBF kernel-based SVM with fine-tuned parameters. L1-regularization and feature selection were used to reduce model complexity and overfitting, and improved test accuracies were observed. Considering the fact that the dataset consists of 16 genre labels and that test accuracy of simple softmax model is 51.03%, we have built successful models for classifying music genres. In real application, a new music track can turn into features the same way as we mentioned, and applied our machine learning models to predict its genre. To further improve the accuracy, we definitely need more music data to train our model. To make this application more friendly to use, the genre labels could be reduced to a more general level, and consequently the prediction accuracy is sure to increase.

Looking forward, there are a number of extensions to this work that could be done.

1) Dig deeper into svm-rbf to find way to solve its overfitting problem;
2) Gathering more data for genres with less data currently to balance data distribution;
3) Model ensembling: combining classifiers by voting or averaging to improve performance
4) Feature refining: add other musically relevant features for better classification results
5) Real application: input new music tracks and transform them into features the same way as we mentioned, and apply our machine learning models to predict its genre.

Contributions

As a group working on this collaborated project, we contributed equally overall. Li Guo has additional contribution on literature search, writing milestone report and establishment of SVM models. Zhiwei Gu has additional contribution on establishment of neural network system and writing the poster. Tianchi Liu has additional contributions on establishing KNN models and writing the poster.

References

[1] Karpov, Igor, and Devika Subramanian. ”Hidden Markov classification for musical genres.” Course Project (2002).

[2] Lee, Chang-Hsing, et al. ”Automatic music genre classification using modulation spectral contrast feature.” Multimedia and Expo, 2007 IEEE International Conference on. IEEE, 2007.

[3] Laurier, Cyril, et al. ”Audio music mood classification using support vector machine.” MIREX task on Audio Mood Classification (2007): 2-4.

[4] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.