ML1

Michael Yitayew
Student Researcher
email

Abhinav Khanna
Student Researcher
akhanna@princeton.edu

Abstract

1 Introduction

Music Information Retrieval is a small but growing field, and music genre classification is an important subsection of this field. With the rise of internet services like Pandora, Spotify, and other recommendation systems, the ability for a computer to tag and associate songs has become ever more pressing. Music genre classification is a challenging problem as many songs can be classified under multiple genres. Furthermore, a song is a complex audio wave that must be translated into a digital vector before being classified. This transformation process takes a continuous wave and condenses it down to a vector set, creating some information loss inherent in the problem. In this work, we focus on testing various classification methods, and various features for classifying 30 second music clips into their appropriate genres. This work builds on the work of Tzanetakis and Cook, who published a paper titled ”Automatic Musical Genre Classification of Audio Signals,” that focuses on the same gtzan dataset that is used in this project. We do a deeper analysis of different classifiers using a similar set of features, analyze the effects of including different features in the classification task, and dissect which genres lead to the highest classification confusion. Through our experiments, we discovered that Blues and Rock are the hardest for our classifiers to isolate from other genres, and thus we report results that both include Blues and Rock as well as exclude those two labels and data subsets. With Blues and Rock, the highest classification accuracy we were able to achieve was 70%. Without the Blues and Rock data subset, we were able to achieve a 80% accuracy.

2 Related Work

3 Methods

3.1 Description of data and data processing

The songs in our study are 30 second clips, and each song was provided as a set of frame-level features for each 20 millisecond frame in the song. “Frame level features do not encode temporal variation, which is important for genre classification”[3] thus Fisher vectors and exemplars were used to aggregate temporal features into a per song feature vector. We performed manual feature selection by observing the discriminant strength of subsets of features on different classifiers; we found MFCC(Mel-Frequency Cepstrum Coefficients) to be the most discriminant feature for genre classification. Other features such as Chroma and Key Strength also slightly improved overall accuracy.

3.2 Classification Methods

We used the following 8 different classifiers; eight are from the SciKitLearn Python Libraries and the Neural Net is from Matlab.

1. *K-nearest-neighbors(KNN)*
2. *Logistic Regression with l_2 penalty(LR)* using one-vs-rest scheme
3. *PCA followed by Logistic Regression (PCALR)*
4. *Random Forests Classifier*
5. *PCA followed by Random Forests classifier*
6. *Support Vector Machine (SVM)* with an exponential kernel function
7. *Naive Bayes classifier(NB)* using the Multinomial Naive Bayes algorithm
8. *AdaBoost* with Random Forests as Weak Learners
9. *NNet Neural Network* from Matlab - A feed forward network with a single hidden sigmoid layer containing 50 neurons and an output softmax layer.

Decision Trees were not included as Random Forest builds on the Decision Tree data structure, making checking both seem slightly redundant.

3.3 Evalulation

We performed stratified 10-fold cross-validation for each classifier. In this procedure, input is divided into 10 random equal size folds. The classifier is then trained on 9 of the folds and tested on a single fold. The accuracy of a classifier is average of the accuracy values achieved when each fold is used as a testing fold.

In addition, we looked at confusion matrices to understand how the error is distributed across labels for each genre category. A confusion matrix is a true labels by predicted labels two dimensional table that counts the number of times a predicted label occurs for a song with a given true label. It provides detailed analysis of where the classifications are missing, and what genres are creating the most misclassification errors.

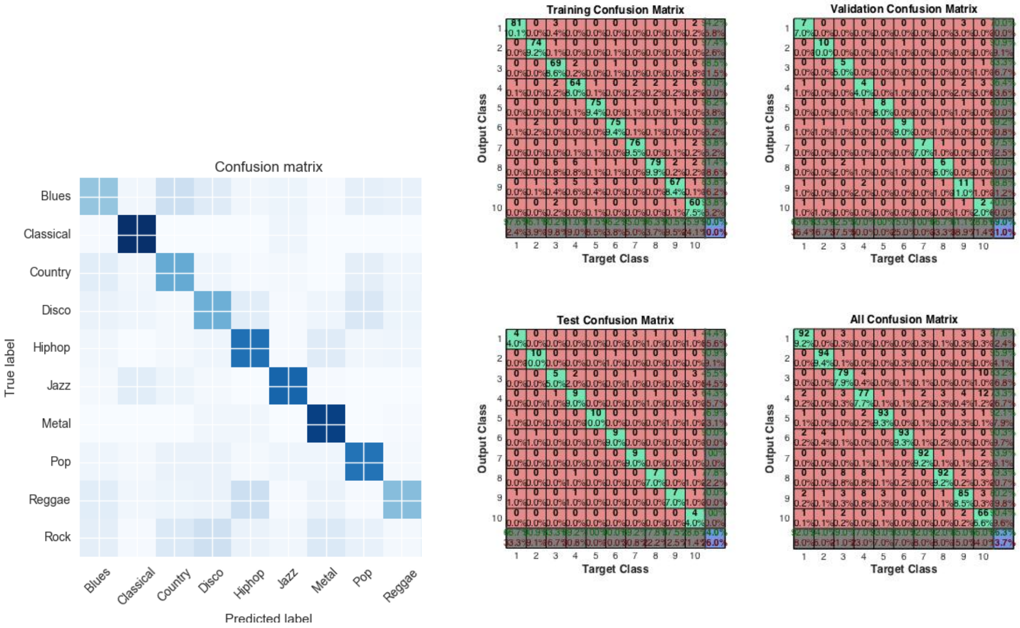
4 Results

We tested the classifiers against a data set containing all ten class labels as well as a data set that did not include Blues and Rock. The following error-rates were achieved at the classifiers' optimal parameter settings against the complete data set. Optimal parameter settings were found by plotting the error-rate vs parameter graph and finding the minimum. Performance was variable across the classifiers with Neural-Nets, Random Forest and AdaBoost with Random Forests performing the best.

Classification accuracy	
Classifier	**Error-rate %**
KNN	42
LR	41
PCALR	39
RF	**35**
PCARF	39
SVM	46
NB	45
Ada	**35**
NNet	**26**

In order to better understand what the distribution of misclassifications looked like, we took the top classifiers, and created confusion matrices of their classifications. As you can see in the figure below, the confusion matrix for Random Forest Classifier suggests that Blues and Rock both are often mislabeled as another class, while the rest of the genres appear to be far more accurately classified. This trend reappeared for AdaBoost, and NN based classifiers. In order to verify this, we reran the top classifiers with Blues and Rock removed. The table below showcases the accuracy rates on the data subset that does not contain Blues and Rock data. Without Blues and Rock labels,

Figure 1: Confusion Matrix for Random Forest and Neural Network - With Blues and Rock



the error rates decrease, and in the best case scenario the feed forward neural network decreases by almost 10%. This seems to agree with our observation that Blues and Rock are heavily misclassified, and that without them in the data set, the other classification labels are relatively discriminate.

Classification accuracy	
Classifier	**Error-rate %**
KNN	32
LR	0
PCALR	25
RF	**25**
PCARF	29
SVM	0
NB	0
Ada	**24**
NNet	**16**

4.1 Parameter Optimization

For the purposes of brevity, we will only talk about the parameter optimization for the top 3 classifiers, the feed forward neural network, the random forest classifier, and the AdaBoost classifier. We focused on optimizing the random forest for two of its parameters: max depth and max learners. We manually fiddled with the other parameters once the optimums for these values were selected. To determine the optimum depth and learner values, we graphed the change in error as the value of one (while the other was held constant) was changed from a small value to a relatively large value (or until the curve plateaued). Figure 2 showcases the change in error vs the number of learners and the change in error vs the depth, but in the end, values of 110 and 31 were used for max learners and max depth respectively. The AdaBoost classifier was optimized for the number of estimators and the learning rate. The base estimator used was identified via manual choice, and the one with the lowest error was chosen. Similar to the Random Forest classifier, the optimal values for the number of estimators and the learning rate were chosen by plotting the range of values and identifying the minimum as can be seen in Figure 3. The final values for the AdaBoost classifier's hyperparameters were 10 for the number of estimators and 0.01 for the learning rate. The other parameters for AdaBoost were not ranged, but categorical, and thus were fiddled with and the option that minimized error selected after the optimal hyper parameters were chosen. The feed forward neural network was tried with the following hidden layer sizes: 10, 30, 50, 75, and 100. After trying these values, it was discovered that post 50, the error rate does not decrease significantly but the time needed for the net to converge grows dramatically.

Figure 2: Error vs Learners and Error vs Max Depth

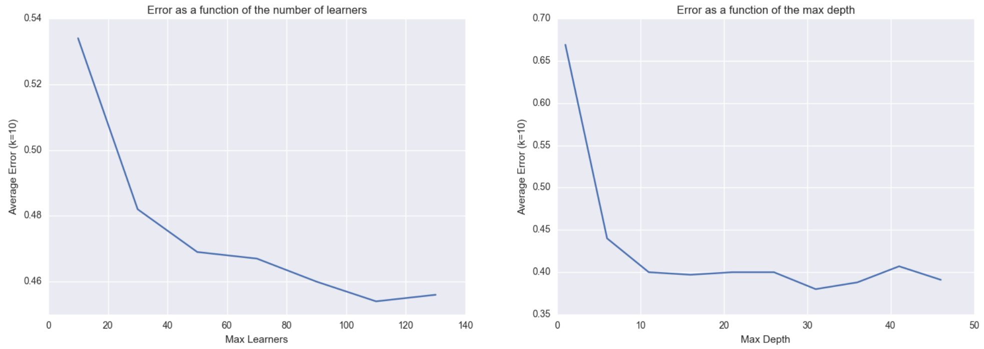
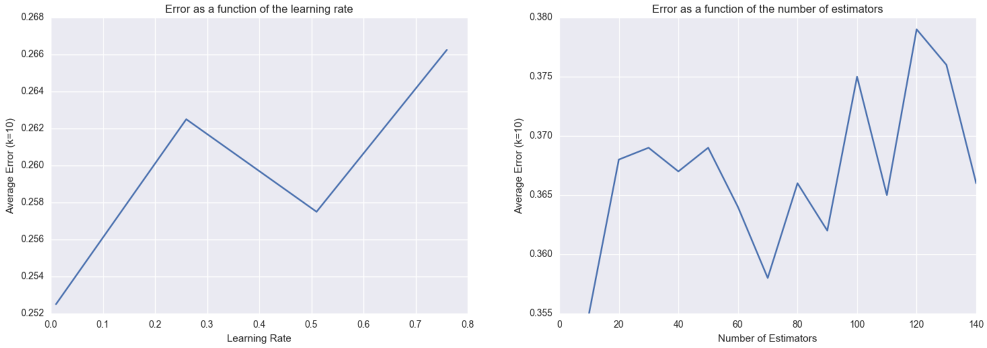


Figure 3: Error vs Learning Rate and Error vs Max Depth



5 Discussion and Conclusion

In this work, we compared 10 classification methods to predict the music genre for a particular 30 second clip after converting the audio signal to a digital vector representation of the features we listed above. After performing manual feature selection and picking the most discriminatory features, we also compared the classification methods with and without Rock and Blues data subset entries from the GTZan data set. Considering the confusion matrices and the classification error rates of the different classifiers with and without the Rock and Blues data subsets, the feed forward neural network performs the best on the data including Blues and Rock subsets, and the data excluding Blues and Rock. Including more Frame-level music features in the Fisher Vector creation process helped yield higher accuracy rates for the neural networks. From a human analysis perspective, it makes sense that the most helpful features for the neural net are features that pertain specifically to musical elements in the song. Its conceivable to imagine that humans also recognize genres based on features of these type.

There are a number of directions we could go to improve this work. Seeing as the Neural Network performed the best on this data set, and that the performance of neural networks increases with the increase in training data, gathering more training data could immensely improve accuracy. Furthermore, different types of neural networks may do a better job at capturing the differences between these classes than the standard feed forward neural network. Recurrent Neural Networks and other custom neural networks may provide increased accuracy and are worth trying. Focusing on better feature extraction, it has been mentioned in literature that instrument details about a given song can be greatly beneficial to genre classification. Adding in instrument recognition to our Fisher Vectors may allow further distinction between these classes. However, because Blues and Rock originate from similar roots and form the basis for many of the other genres, instrument details may not help us resolve the issue of mislabeling Blues and Rock.

Acknowledgments

We would like to acknowledge Princeton University for offering the wonderful COS 424 class that provided us with the opportunity to take on this research. We would also like to thank our Professor, Professor Englehart, and her TA team for providing us with guidance and the tools necessary to make this project happen.

-Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12: 28252830