Audio classification

Coauthor
Affiliation
email

Coauthor
Affiliation
email

Abstract

1 Introduction

In this report, we consider a common problem in musical machine learning: classification by genre. Specifically, we attempt to classify the well-known GTZAN dataset of a thousand 30-second snippets of songs divided evenly across the genres of blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock [1].

The remainder of this paper describes some of the existing research genre classification, our methods of extracting and transforming features from the data, and the trials we ran to explore the relative effectiveness of various combinations of features and classifiers. We conclude by presenting and discussing the results of these trials.

2 Related Work

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3 Methods

As we approached this classification problem, we decided to begin by visualizing the data in order to gain some understanding of how the song samples differed across genre. Afterwards, we began running classification trials and comparing results. In these trials, we varied each of the following properties: features considered, classifiers used, and methods of combining the results of multiple classifiers.

3.1 Data Visualization

First we did preliminary analysis of the song data, namely looking at the time-average mean of MFCC coefficients for each genre to verify that there is a measurable difference between genres. We also did a visualization of the dataset by projecting the MFCC feature and HCDF feature using t-distributed stochastic neighborhood embedding (t-sne) to verify that clustering based algorithms are viable for classifying this dataset.

3.2 Trial Design

We did 10-fold CV with zero-one loss to calculate the cross validation error of each feature set and classifier. The dataset is first shuffled and the training dataset is sampled proportional to the label (i.e each training set of 900 will contain all 10 labels in equal proportion).

3.3 Features

Initially, we selected five features based on their promising performance in the literature: MFCC, Chroma, Energy, Spectral Flux, and HCDF. We then used the provided scripts to load the song samples in .mat format and extract the five features from each sample. Because each feature considered is a frame-level feature and therefore high dimensional, we used the provided scripts to apply Fisher Vectors to each feature to generate descriptors before converting the results into .csv format.

Once we had completed the extraction and quantization process, we experimented with varying the features considered in three ways.

3.3.1 Experiment 1: Combinations of Features using Fisher Vectors across classifiers

In this experiment, we ran trials for every combination of the five features encoded with Fisher Vectors. For each of eight classifiers considered, we collected the results of predicting based on each possible combination. We used the following classifiers:

Clustering classifiers:	K-Nearest Neighbors (KNN3, KNN5)
Generative classifiers:	Gaussian Naive Bayes (GNB)
Discriminative classifiers:	Linear-, Quadratic Discriminant Analysis (LDA, QDA), SVM (linear, and rbf kernels), Random forest (RF)

3.3.2 Experiment 2: Using MFCC and Chroma Raw Data

The goal of this experiment was to compare the performance of Fisher Vector-generated descriptors with the performance of MFCC and Chroma raw data. We extracted the first 1000 frames of each song and concatenated the features to form a 32000×1000 matrix for MFCC and 12000×1000 matrix for chroma, then we used PCA and t-sne to project the data down to a 900-dimensional feature to maintain a feature-sample ratio of less than 1. We then compared the performance of these matrices with the corresponding Fisher Vector matrices.

3.4 Classifier Combinations

Once we had collected the results achieved by single classifiers, we decided to try pooling the predictions of several classifiers into a single set of predictions. Therefore, we experimented with hard voting and soft voting using our best-performing classifiers.

3.4.1 Experiment 1: Hard Voting

In this experiment, we ran trials in which three classifiers - 5-Nearest Neighbors, Linear SVM, and Gaussian Naive Bayes - engaged in a hard vote to determine the final classification of a sample. This means that each classifier cast one vote per sample considered. Performance across each combination of FV features was considered.

3.4.2 Experiment 2: Soft Voting

In this experiment, we conducted a soft vote of four classifiers: 5-Nearest Neighbors, Linear SVM, Gaussian Naive Bayes, and Random Forest with a forest of 50 trees. The best performing classifier, Linear SVM, was weighted double. Again, performance across each combination of FV features was considered.

4 Results

4.1 Preliminary Analysis

We see that the distribution of the means of MFC coefficients do indeed differ across genres, suggesting that MFCC is an informative feature for classification.

We also see even in an extremely compressed 3-dimensional projection of the MFC data, t-sne is able to discern clear clusters of songs in each genre. This gives us a hint as to which genres are easily separated and which are easily confused.

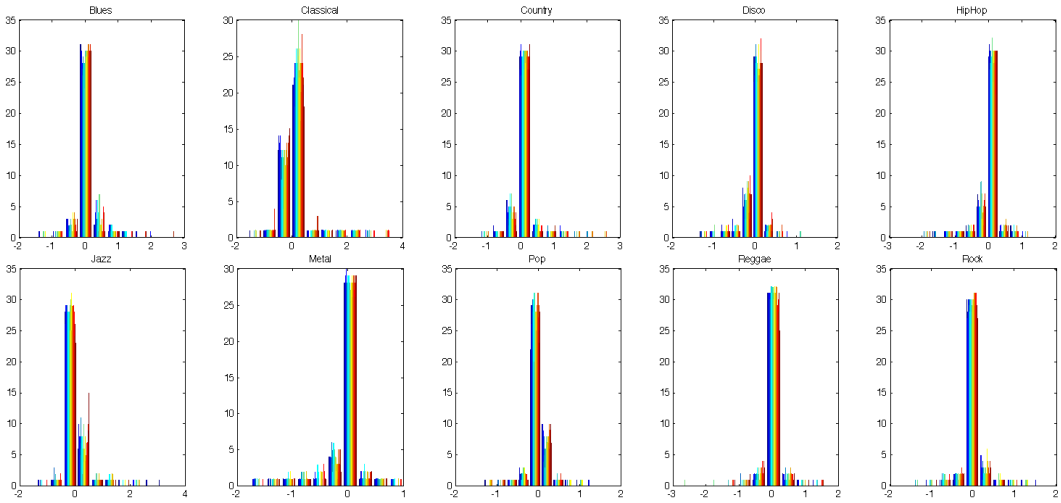
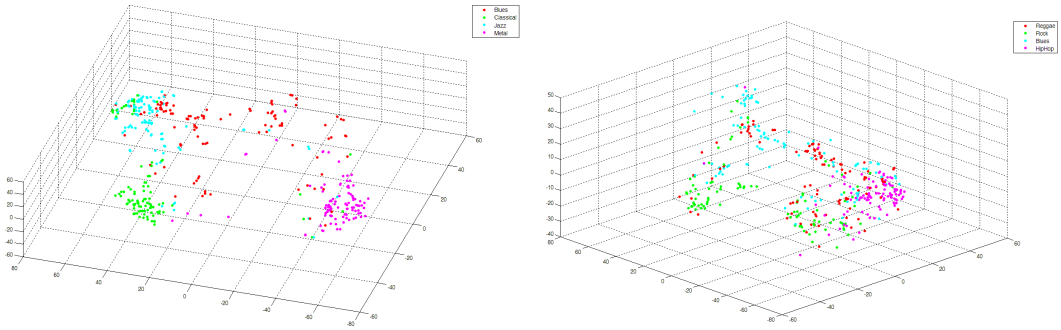


Figure 1: Plots for time-averaged means of MFC coefficients for the 10 genres, every colored line represents one song



(a) t-sne subplots for Blues, Classical, Jazz, Metal showing clean separability

(b) same t-sne subplots for Reggae, Rock, Blues, Hip-Hop showing confusion

Figure 2: t-sne plots in 3-D of a projection of 960-dimensional MFCC FV

4.2 Combinations of features with different classifiers

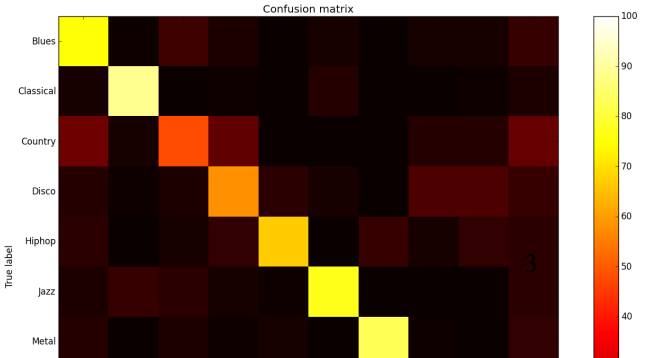
We see in figure 3 that the feature set of MFCC, chroma and brightness performed with 0.245 error rate with a soft voting algorithm of KNN5, SVM(Linear kernel) Gaussian NB, and Random Forest.

4.3 Fisher vectors vs PCA of raw timeframe data

In figure 4 we see that the generated Fisher Vectors performs better than just a simple collection of raw data vectors

5 Discussion and Conclusion

5.1 Confusion matrix



The confusion matrix in figure 5 is built from MFCC+HCDF Fisher Vector feature set with KNN5 classifier. We see that Classical, Metal , Blues and Jazz are the best identified genres, with accuracy exceeding 70%, this corroborates with our visualization in figure 2a which showed clear

	KNN3	KNN5	SVM-Lin	SVM-RBF	GNB	RF	Softvote 2,3,5,6	Hardvote 2,3,5
Min error	0.343	0.333	0.255	0.384	0.37	0.288	0.245	0.261
Feature set	MFCC HCDF			MFCC	MFCC energy	chroma HCDF	MFCC chroma brightness	HCDF brightness

Figure 3: 10-fold CV classification error rate for different classifiers and the features that performed the best, for full data table see section 6 Appendix

	KNN3	SVM-Lin
MFCC FV	0.343	0.255
MFCC Raw data	0.461	0.396
Chroma FV	0.605	0.558
Chroma Raw data	0.671	0.679

Figure 4: The error rates of two Feature FV against their raw data counterparts with two classifiers

clustering for those genres. Genre pairs with the most confusion are {Reggae, Hip-Hop}, {Rock, Disco}, {Rock, Country}, which also corroborates with our visualizations in figure 2b.

5.2 Classifiers and features

It is no surprise that the best single classifier before ensemble learning is linear SVM, because our underlying features are represented in Fisher Vectors, which lends itself to efficient linear classification. The Fisher Kernel inherits advantages from both generative and discriminative models by building a kernel from a generative model (in this case GMM), it characterizes a sample by its deviation from the model measured by computing the gradient of the sample log-likelihood with respect to the model parameters.[2].

Acknowledgments

References

[1] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[2] SANCHEZ, J., PERRONNIN, F., MENSINK, T., AND VERBEEK, J. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision 3* (2013), 222–245.

6 Appendix