Music Genre Classification

[Progress Report]

Moritz Gellner
Machine Learning
Northwestern University
moritz.gellner@gmail.com

Josh Jacobson
Machine Learning
Northwestern University
joshjacobson14@gmail.com

Carson Potter
Machine Learning
Northwestern University
cp3192@gmail.com

Sam Toizer
Machine Learning
Northwestern University
samtoizer@gmail.com

ABSTRACT

Genre classification is a commonly performed Machine Learning and Music Information Retrieval practice, producing varying results across a variety of experiments of the past decade. By optimizing on some of today's best practices, and exploring some new classification techniques on the same dataset as used in those prior successes, we hope to glean new information on the nature of genre classification, and further understand potential directions that could improve the art of computational musical classification, deconstruction, and construction as a whole.

1. INTRODUCTION

Genre classification is a somewhat popular topic for research, particularly in the fields of Music Information Retrieval and Machine Learning. There have been many papers published on methods for classifying genres using audio features and other methods. Our goal for this project was to combine common genre classification techniques with new methods and optimization in an attempt to achieve higher classification accuracy. In doing so, we aim to provide comparative analytics between common and new genre classification techniques. By introducing boosting and a naive bayes classification of lyrical content, we demonstrate alternative, and sometimes more effective, means of genre classification.

2. MOTIVATION

The process of classifying genres with machine learning methods can reveal a lot about the fundamental characteristics of different genres, what composes the music within each genre, and the mathematical, predictable tendencies of certain types of music. Given our teams passion for music, we are interested in seeing if we can isolate patterns between certain genres of music, and in doing so unearth behaviors that can be used in applications far beyond genre classification. Many uses exist for effective genre classifiers, such as music cataloguing tools for applications such as iTunes, and more potent recommendation software for similar listening services. Potential applications of successful feature extraction include computational composition of music, and interesting and powerful data mapping of the growth of genre and lyrical growth over the history of music. Additionally, popularity predictors and automated music generators are but a few of a number of interesting theoretical extensions of research such as this.

3. DATASET

In order to conduct our comparative analysis of these two approaches to the genre classification problem, we used the GTZAN Dataset [1], which was developed by George Tzanetakis. This dataset consists of 100 short song clips in each of 10 genres. The clips are each about 30 seconds long, and are drawn from a variety of sources, including live performances and radio broadcasts in addition to typical studio recordings.

The dataset doesn't come with song title and artist information, but we were able to obtain this information for most songs thanks to the research of Bob Sturm. Since the jazz and classical songs included in the dataset are all completely instrumental and therefore don't have lyrics for us to analyze, we decided to work with the other 8 genres only. These genres are blues, country, disco, hip hop, metal, pop, reggae, and rock. To competently attempt classification by lyrical content, we gathered the lyrics of the songs in GTZAN using an API from the website ChartLyrics. We saved the responses for each song in a separate .txt file matching the song title, and parsed it accordingly in our lyrical classification tool.

4. AUDIO FEATURES

In order to extract low-level audio features for use in genre classification, we used MARSYAS[2], an open-source audio processing framework developed by George Tzanetakis.

We provided the following features to our classifier:

[1] http://marsyas.info/download/data_sets/
[2] http://marsyas.info/

Spectral Centroid[3] This provides a measure of the "center of gravity" of the frequency spectrum. Higher values correspond to a "brighter" sound.

Spectral Rolloff This measures the frequency below which 85% of the magnitude distribution is concentrated.

Spectral Flux[4] This measures the amount of spectral variation between adjacent short sections of the audio file.

Time domain zero crossings This measures the noisiness of the signal.

Mel-Frequency Cepstral Coefficients (MFCC's)[5] MFCC's describe the harmonic content of a song through an analysis of the prevailing frequences based on how far apart certain frequences "sound" to our ears in the context of music. in perceptual terms. of perceptually motivated, based on STFT.

Spectral Flatness This is a measure of the spectral variance throughout an audio file.

MARSYAS provides automated command-line tools for collecting all of these features from a collection of audio files and compiling the results into a single data file in a format (.arff) that can be readily imported into Weka. Weka[6] is a machine learning toolbox with implemented in Java with a graphical interface that allows easy implementation of a variety of machine learning methods. Since MARSYAS was built in order to be able to integrate with Weka, the interface between the two is seamless and we can simply import the .arff file generated by MARSYAS into Weka for analysis. Using Weka, we tried out a variety of learning methods and analyzed their effectiveness using 10-fold cross-validation with our 800-song dataset. In addition to the normal range of learning methods, we were also able to implement boosting with the AdaBoost algorithm. We ultimately generated the results seen below with a bayes net classifier and 10-fold validation. We AdaBoosted with 10 iterations.

5. LYRICAL CONTENT

As mentioned above, we collected lyrical data using a python script in conjunction with the ChartLyrics API[7], searching based on the GTZAN dataset information provided by Bob Sturm. Even though we were unable to find the lyrics for some songs in the database, and some songs in the dataset turned out to be instrumentals, we were able to successfully scrape lyrics for 68.375% of the songs that we tried. With a total of 547 songs about evenly distributed among all 8 genres, we had lyrics for about 68 songs per genre.
We used three different methods to attempt to classify lyrics, which are referred to as the Frequency-Independent Method, the Frequency-Dependent Method, and the Word Count Method. The Frequency-Independent Method consisted of a Naive Bayes Classifier that used a dictionary made up of words from the songs. In this method, we only added to a word's probability if it existed at all in a song and didn't increase its probability based on the number of times it was used in a song. If our training set consisted of genre G with songs with the lyrics:

Song 1: I love you, you love me
Song 2: Love is all you need

then the probability of love being in G is 1. The Frequency-Dependent Method was also a Naive Bayes classifier, but the dictionary was calculated differently. For this method, the probability was number of times a word appeared in a genre over number of words in that genre. For the above example, the probability of love being in G is 3/11, or 0.272. The Word Count Method simply found the average number of words in each genre and classified a song S based on the closest average to the number of words in S.

		Classified as							
		Bl	Co	Di	HH	Me	Pop	Reg	Ro
Correct genre	Blues	60	9	11	0	0	7	4	9
	Country	4	67	10	0	1	0	4	14
	Disco	1	7	59	6	2	4	11	10
	Hip Hop	1	0	10	54	4	13	16	2
	Metal	0	0	1	0	81	0	1	17
	Pop	4	7	6	8	1	63	4	6
	Reggae	2	16	14	5	3	8	58	4
	Rock	5	18	7	0	14	1	8	47

Table 1: Confusion Matrix for Audio Feature analysis using Bayes Net learner

6. RESULTS

6.1 Audio Features

After experimenting with a variety of different learning methods in Weka, we decided to use a Bayes Net classifier. We performed 10-fold cross validation on our dataset, which has a total of 800 examples evenly divided into 8 genres (blues, country, disco, hip hop, metal, pop, reggae, and rock). So in each fold, Weka trains on 90 examples from each class (720 total) and tests on 10 from each class (80 total). We will discuss the overall results for a few different approaches.
First of all, in using all of the audio features presented above with a Bayes Net learner, we were able to achieve 61.125% accuracy in overall classification, which is similar to the 61% accuracy reported by George Tzanetakis in his paper[8] on genre classification. The confusion matrix for this classifier is shown in `Table 1`.

The number of correctly classified audio files (out of 100) for each genre is along the diagonal. The results are fairly close to what we would expect based on an intuitive understanding of genres. Metal was the most successfully classified genre, and almost all of the misclassified metal songs are interpreted as rock, which is the most similar genre. There are many similar trends to the misclassifications, such as the trend of many pop songs being misclassified as hip hop or rock. Overall, the accuracy is quite good and the results seem to make sense from a human listener's perspective. Using the AdaBoost algorithm[9] on top of our Bayes Net

[6] http://www.cs.waikato.ac.nz/ml/weka/
[7] http://www.chartlyrics.com/api.aspx
[8] http://dspace.library.uvic.ca:8080/bitstream/handle/1828/1344/tsap02gtzan.pdf?sequence=1
[9] http://en.wikipedia.org/wiki/AdaBoost

		Classified as							
		Bl	Co	Di	HH	Me	Pop	Reg	Ro
Correct genre	Blues	63	12	8	0	0	4	4	9
	Country	6	73	8	0	1	0	2	10
	Disco	0	9	60	7	3	4	7	10
	Hip Hop	1	0	6	67	6	7	12	1
	Metal	0	0	0	0	85	0	1	14
	Pop	1	8	4	9	2	65	4	7
	Reggae	2	7	12	7	3	6	57	6
	Rock	5	17	7	1	13	2	7	48

Table 2: Confusion Matrix for Audio Feature analysis using Bayes Net learner and AdaBoost

classifier, we were able to improve the accuracy to 64.75%. The confusion matrix for this method is shown in `Table 2`.

In this confusion matrix, we see improvement in classification accuracy across the board, with the most notable improvement in the classification of hip hop songs: 67/100 correctly classified with boosting, vs. 54/100 without. However, this boosting carries with it a high risk of overfitting to the dataset, so the utility of this approach very much depends on the application. We also tested the same Bayes Net learning approach using more limited subsets of the features. When excluding the spectral flatness feature from the analysis, the learner achieves only 54.125% accuracy (or 54.625% with boosting). Full confusion matrices for these cases are not included here, but our results through experimentation with limited feature sets confirmed the utility of the features under consideration in our full set. A t-test comparing our results against the prior classification probability of 12.5% further reinforces the strength of our results. With our maximum accuracy of 64.75% and its standard deviation of 26.26%, we see a t-value of 56.2777 across our 800 size dataset. This corresponds to a two-tailed p-value of less than 0.0001, indicating an extremely statistically signficant difference.

6.2 Lyrical Content

Neither of the Naive Bayes methods used for lyric-based genre classification worked well in practice. In order to test the statistical significance of our results, we did 10-fold cross-validation. We used a single-sample t-test to compare the effectiveness of our method to prior probability. We used this test because we were attempting to determine if the correct classification rates of our methods were statistically different the known correct classification rate of prior probability.

6.2.1 Frequency-Independent Method

This method worked just about as well as prior probability. The mean correct classification rate was 0.1275 with a standard deviation of 0.0249. When compared to a prior probability of 0.125, there was no statistical difference (p = 0.7577). There was, however, an interesting trend in the data. This can be seen in the plot of the correct classifications (`Fig. 1`) and the confusion matrix (`Table 3`).

As can be seen, almost all of the songs are classified as hip-

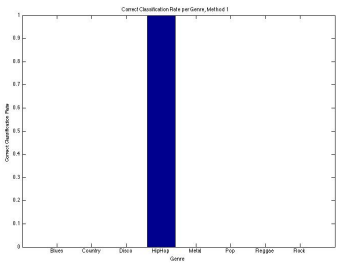

Figure 1: Classification Results for Frequency-Independent Method

		Classified as							
		Bl	Co	Di	HH	Me	Pop	Reg	Ro
Correct genre	Blues		1		4				
	Country			1	4				
	Disco				5				
	Hip Hop				5				
	Metal				5				
	Pop				5				
	Reggae				5				
	Rock				4		1		

Table 3: Confusion Matrix for Lyric Classification using Frequency-Independent Method

hop. This may be because of the sheer volume of words in many hip-hop songs. Since there are so many words in hip-hop, the chance of a word existing in a hip-hop song is much greater than the chance of a word existing in any other genre.

As an empirical example, in our dataset the average number of words in a hip-hop song is about 415 and the average number of words in a blues song is about 165. This means that for the word car, there is about a 2.5 times higher chance it exists in a hip-hop song than in a blues song. This difference makes it much more likely that our classifier will say a song is hip-hop than a different genre. The full table of average word count per genre is shown in `Table 4`.

Since hip-hop seemed like an outlier in our dataset, we wanted to investigate what the results would look like if the genre wasn't included in the dataset at all. We found a mean correct classification rate of 0.2829 with a standard deviation of 0.0579. This was statistically different from the prior probability of 0.143 (p = 3.1505e-5). Though these results are promising, we were simply following an intellectual curiosity. Hip-hop is an important genre, and for our classifier to be unable to have it in the dataset is detrimental to its success.

6.2.2 Frequency Dependent Method

This method was nowhere near as successful as the previous method, which wasn't any good itself. We found a mean

Bl	Co	Ro	Me	Reg	Di	Pop	HH
165.21	183.36	207.35	216.09	216.76	247.56	268.80	414.65

Table 4: Avg. Word Count by Genre

		Classified as							
		Bl	Co	Di	HH	Me	Pop	Reg	Ro
Correct genre	Blues	2					2	1	
	Country	5							
	Disco	2		1		1		1	
	Hip Hop	5							
	Metal	3				1		1	
	Pop	4					1		
	Reggae	3	1					1	
	Rock	1		1		1		2	

Table 5: Confusion Matrix for Lyric classification using Frequency-Dependent Method

correct classification rate of 0.900 and a standard deviation of 0.0316. This was significantly different from the prior probability of 0.125, with p =0.0067. However, this method was significantly worse than prior probability. The correct classification rates per genre are shown in `Fig. 2` and the corresponding confusion matrix is shown in `Table 5`.

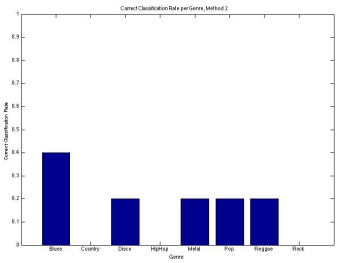

Figure 2: Classification Results for Frequency-Dependent Method

A different interesting trend shows up here. Now, a lot of songs are being misclassified as blues. This makes some sense; blues songs contain the least number of words, so when looking up probabilities for a word in a test song, the weight that blues gives is much more than the weight hip-hop gives. Similar to the previous method, we attempted to remove the outlier genre from the dataset to see if we got better results. Doing so actually worsened the quality of the classifier! We found a mean correct classification rate of 0.0714 with a standard deviation of 0.0309. This method was also significantly worse than prior probability, with p = 4.4716e-4.

6.2.3 Word Count Method

After both of these methods failed, we attempted a third method, namely using the total word count in a song to classify it to a genre. This method is very naive and we didn't expect it to be very successful at all. However, we saw a mean correct classification rate of 0.2325 with a standard deviation of 0.0624. This proved statistically better than prior probability, with p = 4.0758e-4. The plot of correct classification rate is shown in `Fig. 3` and the corresponding confusion matrix is shown in `Table 6`.

We can see from the plot that the classifier always failed

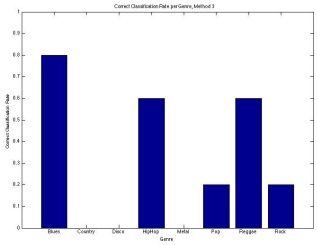

Figure 3: Classification Results for Word Count Method

		Classified as							
		Bl	Co	Di	HH	Me	Pop	Reg	Ro
Correct genre	Blues	4					1		
	Country	3		1					1
	Disco	1			1	1	1		1
	Hip Hop	1			3				1
	Metal	2	1	2					
	Pop	1		1	2		1		
	Reggae	1		1				3	
	Rock	1	1	1				1	1

Table 6: Confusion Matrix for Lyric classification using Word Count Method

on Country, Disco, and Metal. There's a very good reason for this. If we look back to the average number of lyrics per genre ordered from least to most (`Table 4`), it's easy to notice that all three of these genres are sandwiched pretty closely between other genres. The most evident example of this is metal. Using this classifier, for a song to be classified as metal it must have between 211.72 and 216.425 words. There's not much room for variation here! It's because of this strange binning that we don't believe this is a good way to classify songs into genres. Perhaps if all of the genres had very different word counts this method would be successful, but we believe that this method's success on our dataset shows some consistency in the word count of our dataset, not in the quality of the classifier. In other words, we believe that our results using only word count shows overfitting, not quality.

7. CONCLUSION

The results found here support the usefulness of audio features in classifying genre. While it is highly likely that Adaboosting on our dataset caused a localization of our classifier on our data, such potential overfitting did marginally improve our performance, and as such is worth consideration in future genre classification experimentation. Lyrical-based classification, however, seems highly unreliable at the depth used in this experiment. Perhaps on a significantly larger dataset, including heavily weighted and more refined heuristics–such as the word count one used here–such classification could begin to see fruitful and relevant results. Ultimately, genre classification is a somewhat reliable practice in a controlled environment. As data processing enables more sophisticated training across a larger variety of genres, it will be interesting to see how such classification technology fares against a real-world context wrought with thousands of only subtly distinguishable genres.