Comparison of Lyrical and Audio Features for Music Classification

1 Introduction

1.1 Abstract

In this project, Comparison of Lyrical and Audio Features for Music Classification, we experimented with several machine-learning algorithms to conduct genre and decade classifications of songs based on their lyrics and audio properties.

1.2 Motivation

We explored the classification and analysis of songs based on lyrics and audio properties using different machine-learning algorithms and metrics. For people, it is often easy to determine a song's genre, or decade of release just by listening to the song, because there are certain key traits of a song genre, or songs from a certain decade that human can easily identify.

Using machine-learning techniques and algorithms, we want to determine which features in a song's lyrics and/or audio input help determine its genre and decade.

Our project can be used in the field of music classification, music recommendation and music analysis. For example, many music website and applications, e.g. Spotify, Pandora, and Google Play, are starting the feature of recommending users music based on their past songs/genres preference and selection. Our project will explore this field of music classification, finding the fittest features and eliminating irrelevant features.

1.3 Basic Approach and Summarized Result

We mainly used Naive Bayes and Support Vector Machine to train the model. First, we trained the model purely based on lyrical data, and then trained the model purely based on audio data. Eventually, we did the training with the combination of the two inputs.

We found that the model that trained by the combination of lyrical and audio data performed the best in the genre classification. We also found that some genres can be identified by their lyrics, but not by audio features and vice versa. Also in genre classification, pure lyrical data is a better indication of a music genre than pure audio data. However, neither lyrical features nor audio features is a good indicator for decade of release. We also found that linear SVM performed consistently better than than Naive Bayes on all our datasets.

2 Problem Definition and Methods

2.1 Task Definition

In this project, we mainly addressed three problems.

2.1.1 Lyrics versus Audio input

Between lyrical and audio input, we want to find out which proves a better indication for the genre/decade that the song belongs to.

When a human listen to a song, he or she both understands the lyrics and detects certain trends in the music, both of which provides some insight of the genre and decade of release of this song. However, for a machine to classify a song into certain genre or decade, we are curious which set of features (lyrics vs audio) are more indicative.

2.1.2 Features of lyrics or audio data

Among both lyrical and audio features, we want to determine which features are an especially strongly signifier of a song's genre or decade of release.

We are interested in identifying which lyrics or audio features are strong signals of a song's genre or decade. What are the signifying words for each genre or decade? Which words or audio features indicate a song is not part of a particular genre or decade? Can any audio features, like rhythm or energy, be a good indicator of the genre/year classification?

2.1.3 Comparison of SVM and Naive Bayes

We wanted to experiment how different algorithms(SVM and Naive Bayes) behave and perform differently for music genre and decade classifications with different input.

After learning SVM and Naive Bayes in class, we are curious about their differences in practice. Do the extra independence assumptions that Naive Bayes makes really affect its performance so dramatically? Or are they are reasonable assumptions to make in this case?

2.2 Algorithm and Methods

We used supervised learning algorithms, specifically the SVM-Multiclass[1] and Naïve Bayes learning algorithms to learn and predict a song's genre and publishing decade.

SVM is a discriminative supervised learning model used for classification and regression analysis. SVM-multiclass uses one-vs-all with normal binary SVM for classification into multiple categories. In our project, categories are different genres or decades, and each song is classified as one of those genres. C is the parameter of tradeoff between margin size and training error, or the measure of how "hard" the SVM is. The higher the C value, the narrower the margin is and the "harder" the SVM is. We used cross-validation to choose the best C value.

Naive Bayes is a generative supervised learning algorithm based on Bayes' theorem. It assumes that each feature is independent of others.

3 Experimental and Evaluation

3.1 Methodology

3.1.1 Evaluation of Performance

We use test set accuracy to evaluate the performance of the models. By contrasting the test set accuracy between classification based on lyrics and classification based on audio properties, we can determine whether audio properties or lyrics provide a better indication of a song's genre. Furthermore, we can also use test set accuracy to determine which learning algorithm performs better on a given dataset.

3.1.2 Data Source

We obtained the lyrical data and decade of release labels of all songs from the "Million Song Dataset" from Columbia University.[2] It is a freely available collection of audio features and metadata for a million contemporary popular music tracks. The songs come from a broad range of categories and genres. The genre labels of each song are queried from the Musixmatch API.[3] We obtained the audio features from the Echo Nest API. [4] We used the "audio_summary" field of all the songs from the API, which includes features such as danceability, key, loudness, and speechiness.

3.1.3 Data Processing

Lyric data:

The lyrical data in the Million Songs Dataset are preprocessed for the users (and only available preprocessed due to copyright laws). The data are in the form of word counts of the top 5000 word roots after stemming by the Porter2 Algorithm. [5]

In addition to the raw word count, we also performed the TFIDF (term frequency-inverse document frequency) transformation on the word counts using the text feature extraction functions from the Scikit-learn Python library.[6]

Audio Data:

We normalized the values of all audio the data using Z-score Normalization formula:

(original value - training set mean)/ (training set standard deviation)

Combining TFIDF lyric data with audio data:

We combined the TFIDF lyric data with the audio data by filtering out songs that only had audio or lyric data, and then appended the 10 normalized audio features to the end of the 5000 TFIDF lyric features.

3.1.4 Validation

We split our samples into a training set (49% of the whole dataset), validation set (21% of training set) and test set (30% of the whole dataset). Using the training set, we trained the SVM-multiclass model with different C values (C = 0.01,0.1, 1, 10, 100, 1000, 10000), and tested each model on the validation set. We compared the performance of the difference models and found the best model. Finally, we ran the model on the test set to get the test set accuracy.

3.1.5 Detailed Methodology and Use of Algorithm

Before the training process, we first counted the number of songs in each genre. We produced data sets with only the top 3-6 genres by filtering out genres with less than 300 instances. Since our dataset contains many more instances of a certain label than others, in order to train our learning algorithms more intelligently, we also produced data sets with the same number of songs in each category. We did this by first determining the number of songs in the category with fewest songs, and then sampling that number of songs from each of the genres. For example, of the data set with 3 genre classes, we further randomly selected 1400 songs from each genre into a new set of examples where all the genres have equal number of songs. Then, we further split each processed data set into a training, validation and test set. We then trained linear SVMs with different values of C on the training set, and determined the best of those models based on validation set accuracy. Lastly, we ran the model on the test set to get the final test set accuracy. Furthermore, we also analyzed the weights of different features by the best SVM model, as well as how the SVM model performed on instances of different genres, by counting the number of times an instance of a certain genre was classified into each of the genres.

We also used the Naive Bayes algorithm for classification based on word counts (stemmed by Porter2). We compared those results to a linear SVM trained on the same dataset, to evaluate whether SVM or Naive Bayes performed better. We compared SVMs trained on datasets consisting of just audio and just lyrical data to determine whether lyrics or audio input is more effective. Lastly, we trained an SVM on a combined dataset to find the features that most significantly indicate a song's genre.

3.1.6 Analysis based on lyrics

We approached classification of genre and decade of release based on a lyrics dataset in four different ways.

1. We trained linear multi-class SVMs on our training set based on stemmed word counts. We set our C values to 0.001, 0.01, 0.1, 1, 10, 100, 1000, and 10000 and compared the validation set accuracies. Then we use the model that produced the best overall accuracy to classify our test set and recorded the results and test set accuracy.

2. We repeated the above process on the TFIDF transformed data and recorded the results and test accuracy for comparison.

3. We trained a Naive Bayes classifier on our training set of raw word counts, and recorded the associated results.

4. We trained a linear multi-class SVM on lyric data that has an equal number of data from each label (using the method described in 3.1.5). Furthermore, after comparing the prediction result of the 6-genre SVM with the real labels, we selected the 3 most distinct genres (ie. the genres that were least likely to be misclassified as another genre) for further analysis. We performed another linear multi-class SVM classification over the dataset with purely the three most distinctive genres, and recorded results.

3.1.7 Analysis based on audio properties

Similarly to lyrics, we trained multi-class linear SVMs (with C = 0.01, 0.1, 1, 10, 100, 1000 and 10000) on our full dataset with normalized audio features, and compared their validation accuracies. We determined the model with best validation accuracy, and used that to classify our test set and recorded its results. Like for lyrics, we also trained multi-class linear SVMs on an dataset of audio features with an equal number of data from each label, which provided more informative results.

3.1.8 Analysis based on the combined audio and lyrical data

Again, we trained multi-class SVMs (with C = 0.01, 0.1, 1, 10, 100, 1000 and 10000) on our full dataset of combined features, determined the best model using the validation set, and used that model to classify the test set. We also trained the SVMs on a dataset with an equal number of instances per label.

3.2 Results

3.2.1 Decade of Release classification

We first trained a linear multi-class SVM and Naive Bayes classifier on a dataset of roughly 16,000 songs from the 1930s to the 2000s containing their decade as a label. Then, we adjusted our dataset so that all the decades had roughly the same number of instances, and again trained a linear multi-class SVM and Naive Bayes classifier on this new dataset. Our results are as follows:

	Unfiltered Data Set	Filtered Data Set**
Naive Bayes	28.62%	24.67%
SVM	48.14%*	27.3%

* SVM classifies most of the test examples as the 2000s decade

**After filtering, there are same number of instances from each decade

Upon closer inspection of the results of the SVM on the unfiltered dataset, we realized that the SVM was mainly relying on the unbalanced nature of our dataset, because more than 47% of our dataset was from the 2000's decade, and so the SVM managed to achieve such a high accuracy based on essentially guessing that most given songs are from the 2000's.

Thus, based on our results for classification of years based on genre, it seems that the lyrics of a song are likely not a good indicator of its decade of release.

3.2.2 Genre Classification

3.2.2.1 Analysis on Lyrical Data

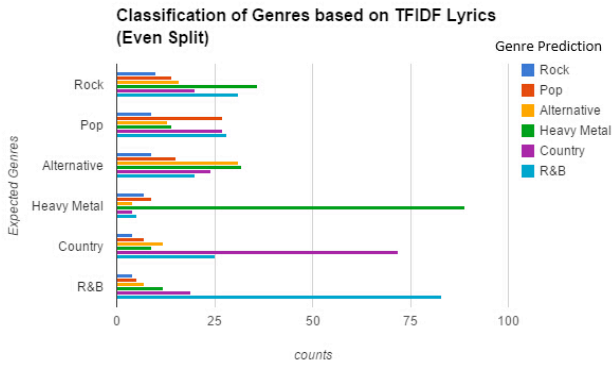
A. Linear Multiclass SVM

We first ran a linear multi-class SVM on the lyrical data of top three genres (Rock, Pop, Alternative). The best validation accuracy with the raw word counts was 47.35% where c = 10. And the accuracy for the test set is 45.99% using this model. In comparison, after applying TFIDF, the best validation accuracy on the validation set occurs when c = 1000 its corresponding test accuracy was 46.35%.

We also run a linear multi-class SVM on the top four, five and six genres similarly. The accuracies on raw word counts and TFIDf transformed data both decrease gradually with increasing number of genres. With six genres the best SVM model on TFIDF data was the one with c=10000 and test accuracy 40.16% The results of raw word counts consistently perform worse than the TFIDF data. With six genres, using the raw word counts only net a 36.07% test accuracy.

In addition, we performed SVM algorithm on the evenly selected data as described in the methodology. The best C value for the three genres case was 100000 with 49.25% accuracy. Similarly, the best C for the four genres data set is 1000 with 44.19% accuracy. In the 5 genres data set, the accuracy decreases to 41.61% with C as 1000. And finally the C value we chose for the 6 genres model was 10000 with 41.43% accuracy.

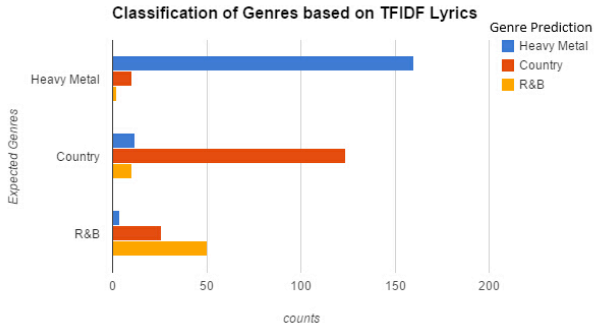
We also plotted the prediction results for an even split 6 genre data as follows.



From the graph above, we can see that it is hard for the SVM machine to distinguish between Rock, Pop, and Alternative songs. On the contrary, the Heavy Metal, Country and R&B songs are much more easily differentiable.

In order to test this hypothesis, we extracted songs from only these three genres (Heavy Metal, Country, R&B) and classified them again. The SVM provided a much better prediction with the best model on the validation set achieving 70.88% accuracy. The corresponding test accuracy was 72.97%.

We again plotted the prediction results for the three promising genres as follows, where we can see that the majority of the songs of each of the genres were classified correctly.



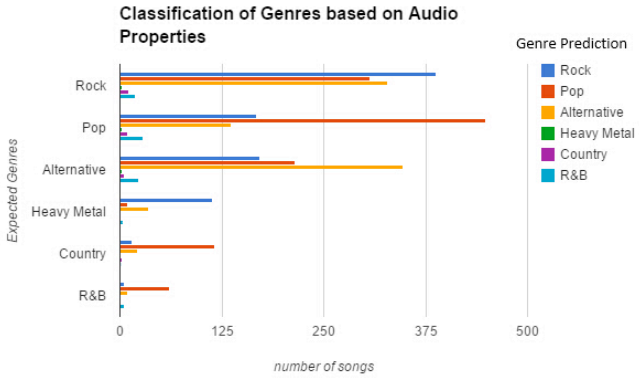
A. Multi-Class Multinomial Naïve Bayes

We ran a Naive Bayes algorithm on the raw word counts for three, four, five and six genres, and the test accuracies were 46.38%, 39.74%, 36.12%, and 33.54% respectively, which are lower than the test accuracies of the SVM for all the datasets except the 3-genre dataset.

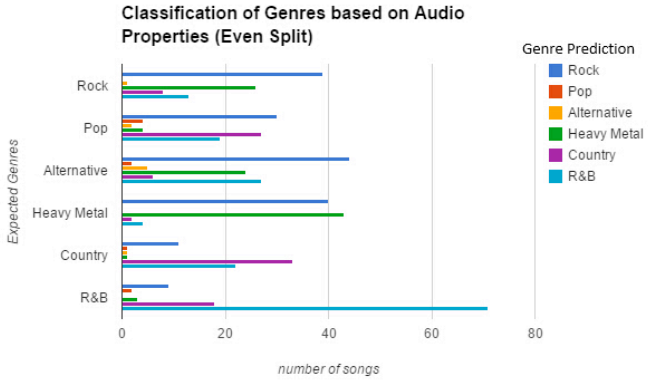
3.2.2.2 Analysis on Audio Data

For the audio properties data, the results resemble those in the lyrics data. For the dataset without evenly split process, the test accuracies goes down from 44.15% with C set to 10 for a 3 genre data set to around 39.6% for the 4, 5, and 6 genres data with C set to 0.1.

From the figure below, we can see how each of the genres are classified by the trained SVM model with C set to 0.1 after cross validation. Most of the Heavy metal, Country, and R&B songs are classified correctly while Rock, Pop, and Alternative are hard to distinguish using SVM trained with Audio data.



After the dataset was even split with equal numbers of songs in each categories, the average accuracy increases by 2% for the 3 genres case, while the accuracy decreases in the other cases to roughly 36%. The predicted results is shown in the figure below:



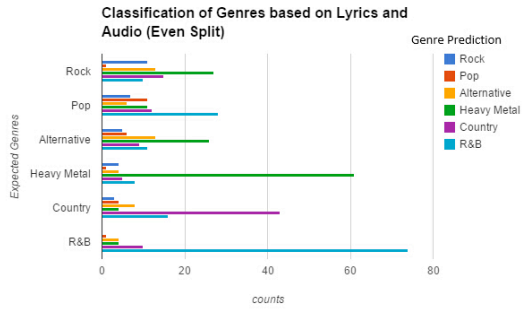
3.2.2.3 Analysis on Combined Data (Lyrical and Audio Data)

For the combined lyrical and audio data set, we also performed cross validation on different values of C for the top three through six genres. With combined data, the accuracy was 49.50% (C = 1000), 39.83% (C = 10), 43.0% (C= 1000), 42.39% (C = 10000) for the top three, four, five, and six genres respectively. Compared to the evenly splitted data, the accuracy for the six genres increases to 44.75%.

Furthermore, we looked at the words with the highest weights for each of the 6 genres in the model we have trained with SVM. In the table below, the significant indicators for Heavy Metal, Country and R&B is shown.

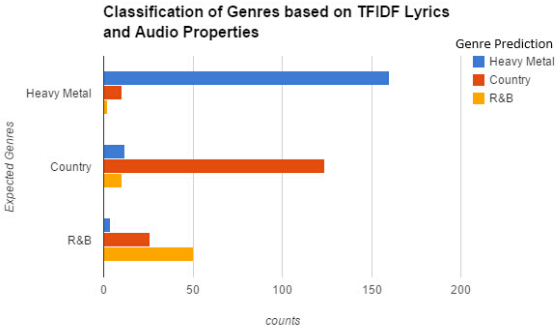
	Heavy Metal	Country	R&B
10 highest weights	energy (audio)	a	speechiness(audio)
	the	home	babi
	of	he	danceability(audio)
	death	again	you
	duration (audio)	every	yeah
	evil	her	love
	pain	around	girl
	az	s	wanna
	dark	at	duration (audio)
	master	lord	got

The prediction results for the six-genre data set with C = 10000 is shown in the following graph.



Again, we decided to take the most distinctive genres -- heavy metal, R&B, and country, to classify based on both audio and lyric features. With C =10000, the SVM had the best validation accuracy of 83.87%, and a test accuracy of 83.92% for a total of 398 songs.

The prediction results are shown in the following figure.



3.3 Discussion

3.3.1 Lyrics versus Audio Input

Classification of genre based on lyric features seems to be more effective than that based on audio features for all of the datasets (top 3-6 genres either evenly split or unfiltered).

For most of our datasets, classification based on lyric features performs better than classification based on audio features by roughly 5% (e.g. 36% for audio vs. 41.43% for lyrics in the 6-genre even split case). Thus, lyrics seem to provide a better indicator of the genre of a song in general.

More specifically, when looking into how audio and lyric inputs work for classifying songs of different genres, it seems that certain genres can be better classified by lyrics than audio (Heavy Metal, Country), whereas others are classified better by audio than lyrics (Rock), and still others are classified well by either dataset (R&B/Soul).

However, note that we did have many more lyric features (5000) than audio features (10), and so the algorithm would have a more detailed understanding of the lyrics of a song compared to its audio. Taking the number of features into account, SVM performing only 5% worse on audio data than on lyric data seems to suggest that given more information about a song's audio, it may potentially provide a stronger indication of its genre than its lyrics.

3.3.2 Important Features of Audio and Lyrical Data

The features with the heaviest weights for each genre are especially interesting in that they are very similar to how we, as humans, associate a song to a particular genre. For R&B/Soul specifically, the three features with the highest weight on the combined (lyrics and audio features) dataset are speechiness, lyrics with the word "baby", and danceability, all of which are features many of us would associate with current R&B/Hip-hop songs. Similarly for Heavy Metal, some of the most important features are energy, and lyrics with words such as "death", "evil", "master". We also see in the combined dataset that in some genres (R&B/Soul, Heavy Metal), audio features have high weights, whereas in others (Country, Pop), all the features with high weights are lyric features.

3.3.3 Comparison of SVM and Naïve Bayes

For our linear SVM vs. Naive Bayes comparison, we mainly focused on lyric data, where SVM performed equally well or better than Naive Bayes for all our genre datasets (top 3-6 genres, even split and unfiltered) by 0-10%. Even for classification of year, where both learning algorithms performed poorly, SVM still performed roughly 3% better than Naive Bayes on the evenly split years dataset. This seems to suggest that the independence assumption that the Naive Bayes classifier makes does not hold for lyric feature data, which is logical since the occurrence of one word in a song often means it's more likely for another associated word to also occur in the same song.

3.3.4 Analysis of Methods

Based on our methods, we obtained interesting results for genre classification based on lyric and audio features. Furthermore, our models actually performed better when trained on evenly split datasets than on unfiltered datasets, which suggests that the model is actually learning the

features of a genre, rather than purely guessing on probability (like the models for year classification are). In comparison to a naive "guesser" on a evenly split dataset of 6 genres, who would be able to achieve an average accuracy of 16.6%, the accuracy of the SVM on the dataset of 44.75% suggests that it is doing something intelligent in genre classification. We also see this when we further analyzed the weights of the SVM and found highly weighted audio and lyric features that were similar to the features that we, as humans, would identify with a certain genre. Furthermore, we also attempted to use radial and polynomial SVMs on the combined datasets, but the radial performed poorly, whereas the polynomial or the sigmoid tanh kernels simply could not learn a model efficiently with our large feature set.

4 Future Work

In this project, one shortcoming is the use of only 10 audio features. We can potentially use a Hidden Markov model to model the sequential audio timbre data. We can assume that each genre will have a unique probability distribution, and thus a separate HMM model.

In regards of the lyrical data, we can potentially try some other natural language processing and other feature extraction methods. In this project, we mainly used word counts for each songs. In addition to word counts, we could have taken repetition of lines, and part-of-speech tag, and other NLP techniques into consideration when designing our feature vectors.

For the combined lyrical and audio data, we simply joined the two data set together according to the song's ID. However, it resulted in a very sparse and high dimensional data set. Since these two data sets are likely to be correlated, we could perform canonical correlation analysis to find a small number of linear combinations from each set of features that have the highest possible between-set correlation. Also, for either datasets, we could also have performed principal component analysis to find smaller linear combinations of a set of variables that still retains most of the information in the original data set.

5 Conclusion

We found that lyrics and audio features do not provide a good indicator of the decade of release of a song. However, for genre classification, the lyrics and audio features do provide a good indication of a song's genre. Furthermore, lyrics tend to be more indicative that a song is of a particular genre than audio is, however using the two in combination provides allows for an even better classification of a song's genre. Also, certain genres (Country, Heavy Metal, R&B) are much more distinctive than others (Rock, Pop, Alternative) based on audio and lyric features. In addition, many of the features that the SVM assign high weights to are also features that we people tend to associate with a particular genre.

Our results can also improve future research, because we have determined that the audio and lyrics features we considered are not indicative of a song's genre, and therefore future research can explore other audio features of a song that may be more indicative of a song's decade of release. Recommender applications can also use some of the highly weighted features we determined to determine which songs to recommend.

6 References

[1]

Joachims, Thorsten. "SVM-multiclass." *SVM-Multiclass: Multi-Class Support Vector Machine*. Cornell University, 14 Aug. 2008. Web. 10 Dec. 2014. <http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html>

[2]

Bertin-Mahieux, Ellis, Whitman, and Lamere. *The Million Song Dataset.* In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011),2011. <http://labrosa.ee.columbia.edu/millionsong/>

[3]

"MusiXmatch API." *MusiXmatch API*. MusiXmatch API, n.d. Web. 10 Dec. 2014. <https://developer.musixmatch.com/>

[4]

"Pyechonest." Overview — *Pyechonest 7.2.1 Documentation.* The Echo Nest, n.d. Web. 10 Dec. 2014. <http://echonest.github.io/pyechonest/index.html>

[5]

"The MusiXmatch Dataset." *The MusiXmatch Dataset.* The MusiXmatch Dataset, n.d. Web. 10 Dec. 2014. <http://labrosa.ee.columbia.edu/millionsong/musixmatch>.

[6]

"Scikit-learn." *Scikit-learn 0.15.2*. Scikit-learn, n.d. Web. 10 Dec. 2014. <http://scikit-learn.org/stable/about.html#citing-scikit-learn>.