Classifying Song Genre by Lyrics and Sounds

Jayen Ram (jjram@stanford.edu) Daniel Salz (dasalz@stanford.edu)

Motivation

Both of us are huge music in general, and as a result, we were inspired by the spam detection problem from the class to create a genre detection classifier using various techniques from the class. We started with rap (binary) classification and moved towards multi class classification (3 genres).

Data

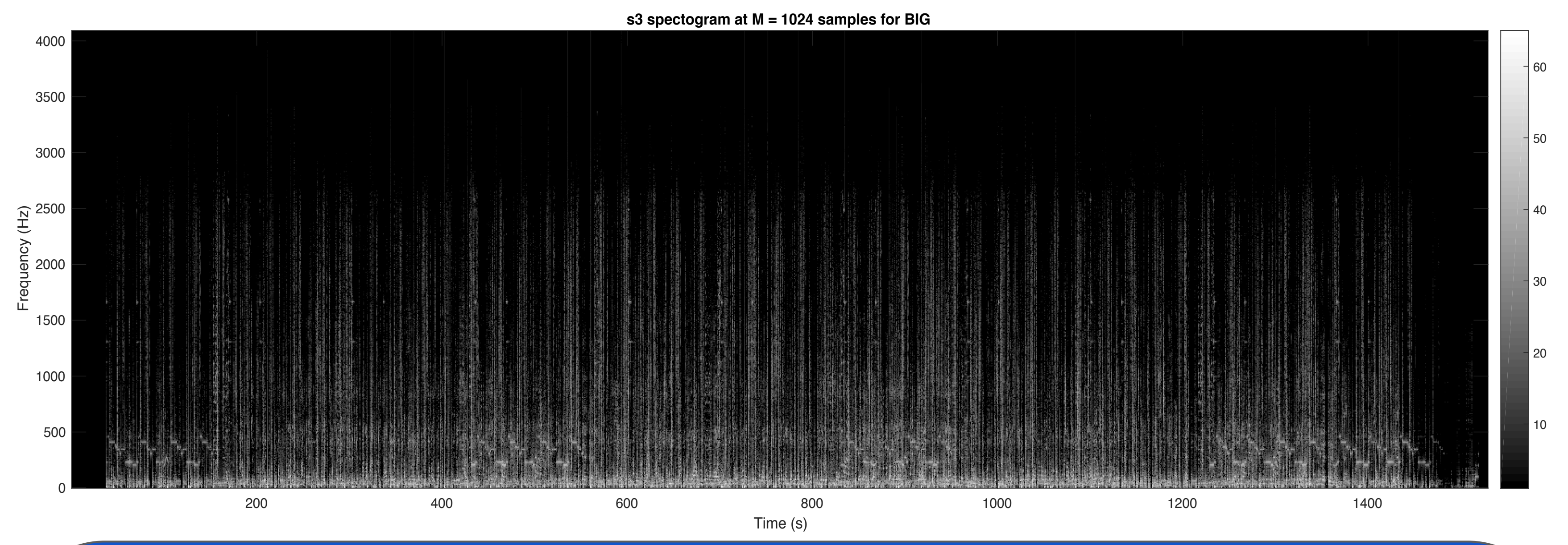
a. Lyrics

We used a dataset from Kaggle that has the lyrics to 55,000 songs This dataset mapped artist to song to lyrics, and we individually classified artists

b. Sounds

We downloaded all the songs from the lyrics dataset and ran them all through the fft function in matlab. We then took only the 2,000,000 indices (as frequency ranges above this seemed redundant) and then sampled every 8 values to reduce our input vector. We also had a different data set consisting of several song properties.

Figure 2. Spectrogram of Rap Song: Big Poppa by Notorious B.I.G.



Features

a. Lyrics

Counted occurrences of all the words in the list and found words that were found a medium amount of times (1000 < x < 3000) and were not well-distributed.

b. Sounds

Frequency: The features are the full 250,000 fft values of the FFT dataset. Different voices, instruments and beats that are represented by different frequencies in an FFT vector.

Lyrics Classification

a. Methodology

We applied Naive Bayes, an SVM with an RBF Kernel, and a linear regression on our feature vectors from values that were found 1000 < x < 3000 times. Naive Bayes had smoothing of λ =1.0, Linear Regression used square loss $(L(x) = \frac{1}{2}\sum_{i=1}^{m}(w^T x - y)^2$, and SVM used RBF kernel with squared loss. We implemented other iterations of these, but only added the most successful results.

b. Results

The SVM performed the best on both the multi-class type and the binary classification. Across the board, training and testing error was extremely similar. There were certain words that indicated rap, country, and rock, which we cannot put on this poster.

Figure 2. Accuracy Table of Lyric Classifiers

	Binary Train Acc.	Binary Test Acc.	3-class Train Acc.	3-class Train Acc.
SVM	94.4	93.2	73.1	72.8
Naive Bayes	92.3	91.6	70.2	69.8
Linear Reg	89.7	88.5	69.5	66.4

c. Analysis

After seeing the songs that our predictor found were wrong, we found that the SVM would find differences between country and rock better than the Naive Bayes. This was a result of very few words between country and rock being particular to each. The Naive Bayes were relatively effective for binary because there were certain words that were only found in rap songs.

d. Future Work

We want to expand to a greater number of classes (instead of 3 classes) and determine key-words, which we have done for rap) that indicate certain genres.

Sound Classification

a. Methodology

We implemented a Neural Net using batch Adam gradient descent and softmax output layer: $\frac{e^x}{\sum_{i=1}^{m} e^{x_i}}$. Parameters were 2 hidden layers w/ 100 neurons each, and a final learning rate of .05.

b. Results

Initially the Neural Net achieved at most 76% accuracy, but after revisiting our dataset and model parameters it went up to 92% accuracy on binary classification for rap based purely on the FFT vector.

c. Analysis

After initially running the NN on binary classification we attempted to expand the problem to 8 classes (genres), however our accuracy dropped to the 20-30% range. After seeing these results we decided to focus on binary classification with the NN and were able to dramatically improve its prediction accuracy.

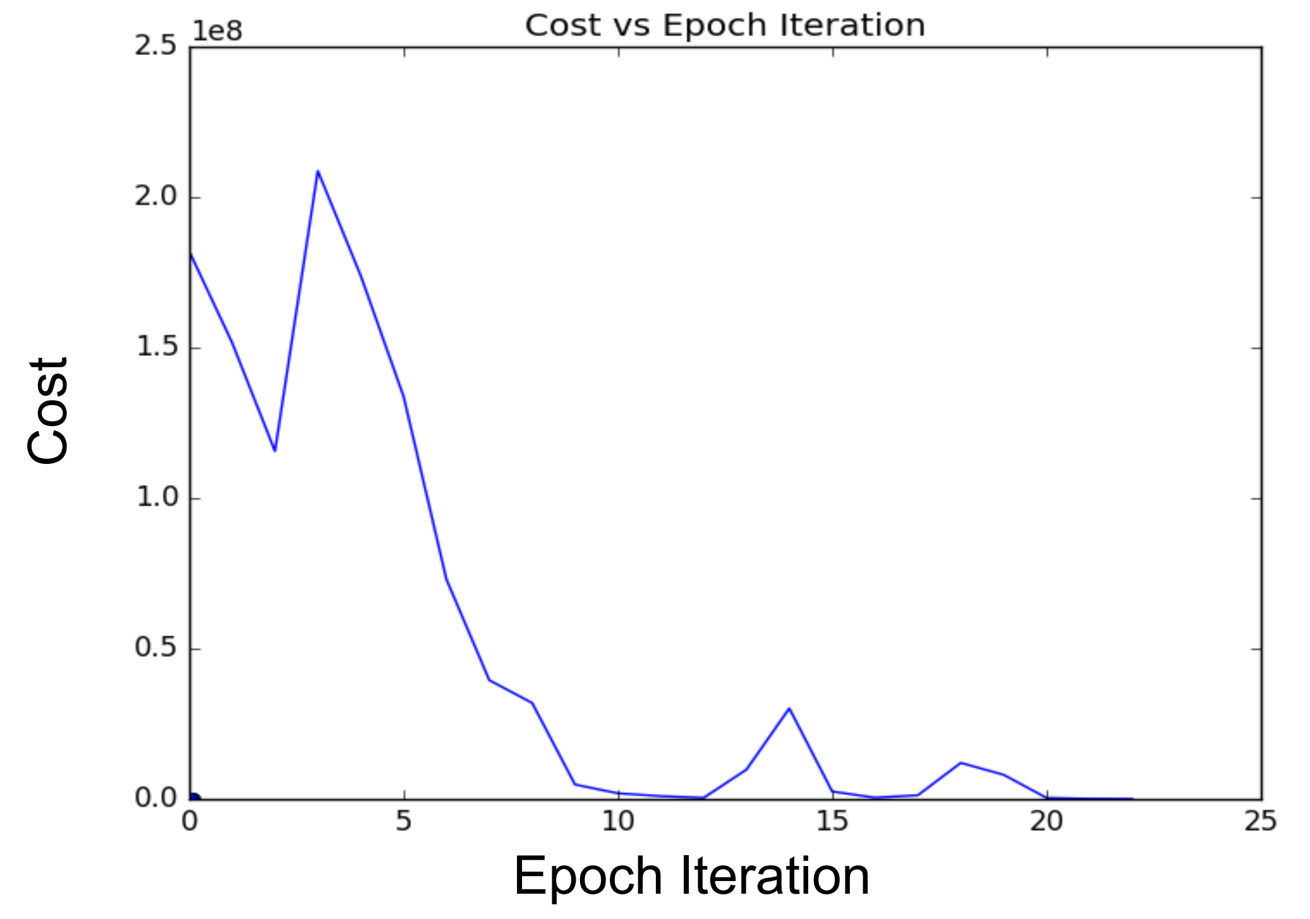
d. Future Work

We want to combine the confidence of both the sound and lyrics models to improve overall accuracy. We want to expand our feature vector to include song properties such as tempo, timbre, pitch, duration, etc. and expand the NN to identify more genres.

Figure 3. Learning Rate vs Train & Test Error

	Learning Rate = 0.1	Learning Rate = 0.05	Learning Rate = 0.01	Learning Rate = 0.005
Test Error	88.4%	92.2%	90.1%	86.8%
Train Error	90.3%	94.5%	91.3%	87.8%

Figure 4. Cost with Learning Rate of 0.05


Cost vs Epoch Iteration
1e8
Cost
Epoch Iteration