Nialls Chavez
CS 529 Machine Learning
Spring 2015
Project 3

Classification of music based on genre
An exercise in Logistic Regression and Gradient Descent

FFT For Classification

Report:

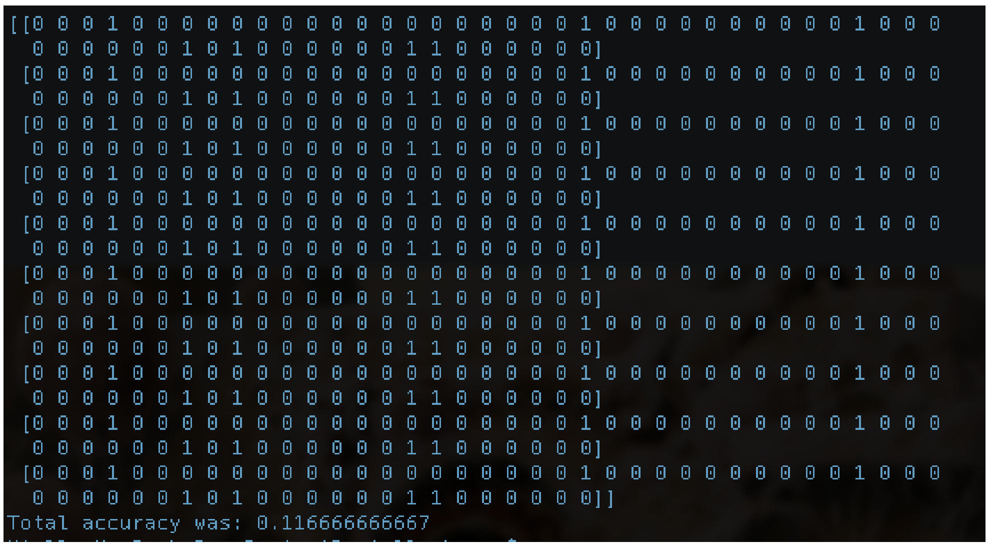
Using the standard implementation found in Mitchell's paper (Generative and Discriminative Classifiers, Naive Bayes and Logistic Regression), I was able to calculate with an average of 54.3333% accuracy the proper classification of a song. Which is about average for logistic regression under these circumstances. This percentage, I believe could be greatly improved if I had continued my calculations for more iterations. This would have helped by increasing the accuracy of the weights.

FFT with 200 classification optimization

Report:

This classification method worked by taking the standard deviation of all columns and then giving more “weight” to the values on a per column basis that were at least 1 standard deviation away. With an ordering preferring those whose values were even further from a single standard deviation. Using this method slightly boosted my average percentage up to 58.8833% accuracy. The same could be said for the previous answer which is that if I had continued my learning for more iterations I believe we could have gotten a higher accuracy rate.

MFCC For Classification



```
[[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0]]
Total accuracy was: 0.116666666667
```

Report:

According to Wikipedia the MFCC is defined as: “ a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.”(http://en.wikipedia.org/wiki/Mel-frequency_cepstrum). This allows us to get a more representative model of the data we are looking at per song. This not only improves the speed of our algorithm (due to the fact that in our example we are only looking at 13 features per song) we were able to get a fairly accurate classification per song. In the case I ran it for, the classification came out to : 11.6667%. Which is pretty bad I believe that this was due to how I parsed the data. This is definitely an error in my code that I unfortunately was not able to mitigate in the time I had to do this project.

Further Improvements:

I believe that the way to improve the classification in the future would be 2 fold. First I would combine the MFCC classification with the standard deviation classification. I believe that this would help our learning algorithm be able to even more accurately classify the data and provide interesting results.