AMATH 482 Homework 4: Music Classification

Rose Ju

March 6, 2020

Abstract

The goal of this project is to classify a given piece of music by sampling 5 second clips. I will be constructing the music data input from different songs. Then I will use three different machine learning algorithms, Linear Discriminant Analysis and Naïve Bayes, to classify the short clips in three different cases. For each case, I will use the input matrix to perform SVD and create the training set and testing set. In the end, I will build the classifier and compute its accuracies for each case.

I. Introduction and Overview

The objective of the project is to create an classifier for the genre and the artist of a music piece by using different machine learning algorithms. I will be using Linear Discriminant Analysis, and Naïve Bayes to create this identifier and calculate each method's accuracy respectively.

We will be performing three test cases for the music classification: Band Classification, the Case for Seattle and Genre Classification. In the first case, I will choose three different bands. The artists picked are Michael Jackson, Soundgarden and Beethoven. The second test is for the case of Seattle. Therefore, we chose three local 90s Seattle bands, Soundgarden, Alice in the Chains and Pearl Jam. The last test is for genre classification and I will use the same pieces from

Michael Jackson and Beethoven as the first test. In addition, I will choose three pieces from AC/DC. As a result, the music pieces are each from pop, classic and rock genres respectively.

II. Theoretical Background

i. Singular Value Decomposition

Singular Value decomposition splits a matrix into produces of three matrices, defined as

$$A = U\Sigma \mathrm{V}^{\mathrm{T}} \tag{1}$$

The columns of U and V are orthonormal and the matrix Σ is a diagonal matrix with positive real entries sorted from largest to smallest. By performing an SVD on matrix A, it first applied a rotation with V^T then stretches by the diagonal matrix Σ. Lastly, it rotates back using the orthonormal matrix U. The resulted matrix from SVD would reduce redundancy and find the maximal variance signal. By the properties of SVD, every matrix A has an SVD if the proper basis of the range and domain, U and V, are chosen. The resulted matrix will then be useful for simplified calculations.

ii. Machine Learning Algorithms

Machine learning is used to perform a specific task without writing out the explicit instructions. It relies on statistical inferences and observed patterns. In this case, I will be using two different supervised machine learning algorithms to optimize the data and classify each music piece.

Linear Discriminant Analysis (LDA) is an algorithm that finds the linear combinations of the clear separation between the data distribution. There are a few assumptions made by LDA in order to simplify the procedure. The algorithm assumes the data is Gaussian or normal, which means that when plotting the data will result in a bell curve. In addition, it assumes a constant variance for each variable. With these optimizations, I will be constructing the model to estimate the mean and variances in each game. LDA separates the probability distribution functions for each case and conducts the highest probability as the result.

Naïve Bayes Algorithm is built on Bayes' theorem for conditional probability. The theorem states the following,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

It is way to calculate the conditional probability given the conditional probability of the reversed. This algorithm applies Bayes' theorem with strong independence assumptions between the variables. Naïve Bayes classifier is an easy algorithm to understand and it estimates the test results based on prior conditional probabilities.

III. Algorithm Implementation and Development

- Load in the audio data file for 5 seconds of each song. Calculate the length and the number of points (or sampling rate) for the audio file. Since the original data is in a column form, we need to transpose the data.

(the next four steps will be done internally by matlab function)

- Before applying the fast Fourier Transform, we need to rescale the frequency domain by $\frac{2\pi}{L}$ because the algorithm is 2π periodic signals. Then, we apply `fftshift()` to get the shifted frequencies.
- Set up a time-slide vector that translates the filter for the sliding window.
- Use a loop to translate a function `func` with the time-slide vector. The function will be set to be either a Gaussian window
- After successfully setup the filter, we multiply it by the audio signal and perform a Fourier transform. Then we take the absolute value to the multiplication and apply `fftshift()` to the absolute value since we used Fourier transform. Finally, we get the matrix for the resulted Gábor transform spectrogram at each time step.
- Lower the sampling rate of the audio file and clip 5 seconds of each song. Then combine all the 5 second segments together for each case and form a matrix
- Performs SVD on the spectrogram matrix to save its $U,\ \Sigma$ and V matrices for each song.
- Use the V matrix from the SVD to construct the training set and test set. In order to have different sets, we randomly split up the data for each turn by creating a vector with random permutation
- Perform LDA and Naïve Bayes classifier to the model for each permutation and calculate the mean accuracy
- Repeat the same process for test case 2 and 3 with different song inputs.

IV. Computational Results

i. Band Classification:

Figure 1 shows the data of the singular value spectrum. Then we perform our algorithms. I

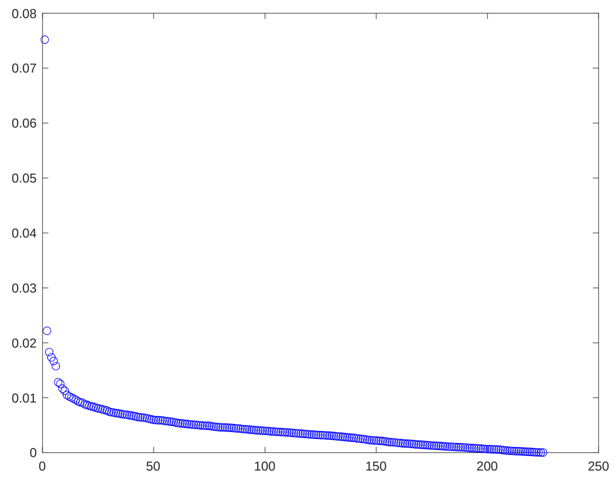

Figure 1: Singular Value Spectrum of the data used for test1

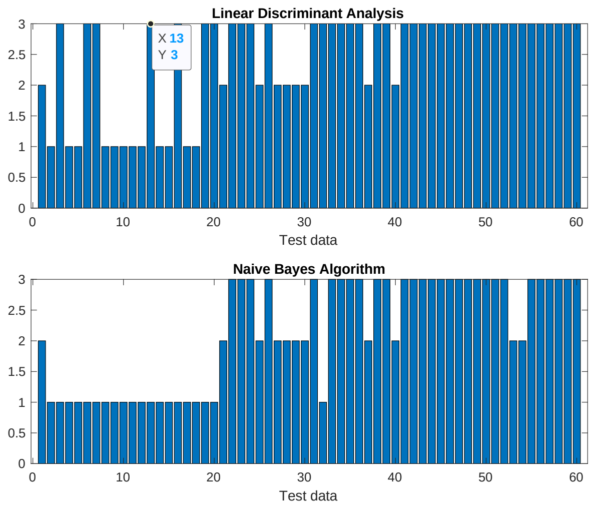

Figure 1: Test 1 results for LDA and Naive Bayes algorithms

use two algorithms, LDA and Naïve Bayes to calculate the testing set on the training data. We found the results of the algorithms to be shown in Figure 2. The accuracy calculated for the data is not ideal. The accuracy given for LDA is 0.6667 and Naïve Bayes is 0.7500. This could definitely be improved in the future. A few possibilities that may have caused low accuracy is the choice of song. They might not be distinguishable enough.

ii. The case for Seattle:

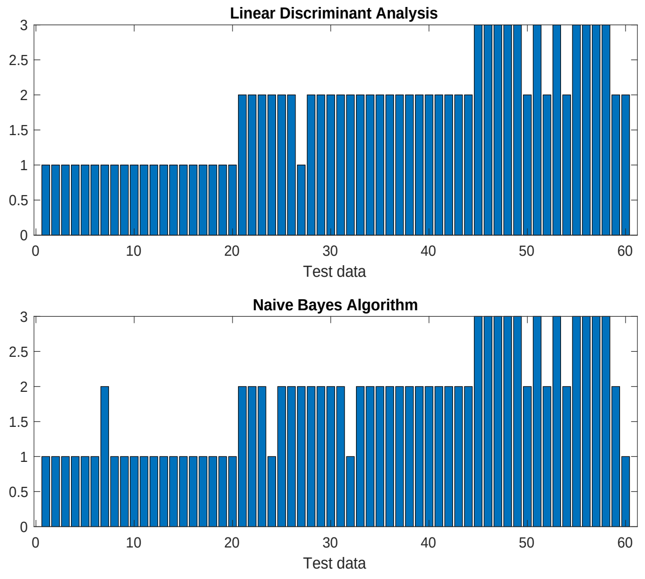

Figure 2: Test 2 results for LDA and Naive Bayes algorithms

Similar to test case 1, I use two algorithms, LDA and Naïve Bayes to calculate the testing set on the training data. We found the results of the algorithms to be shown in Figure 3. The accuracy calculated for the data shows improvements in comparison to test 1. The accuracy given for LDA is 0.8000 and Naïve Bayes is 0.83333. This is slightly higher than the results obtained from test 1. However, it is still far from being 100% accurate. For future investigations, more songs and clips can be analyzed for more accurate training set.

iii. Genre Classification:

For the last test case, LDA and Naïve Bayes algorithms were used again to calculate the testing set on the training data. We found the results of the algorithms to be shown in Figure 4. Again, the accuracy is not very high. The accuracy given for LDA is 0.7333 and Naïve Bayes is 0.6833. Unlike the other two test, LDA accuracy is higher than Naïve Bayes algorithm in this test.

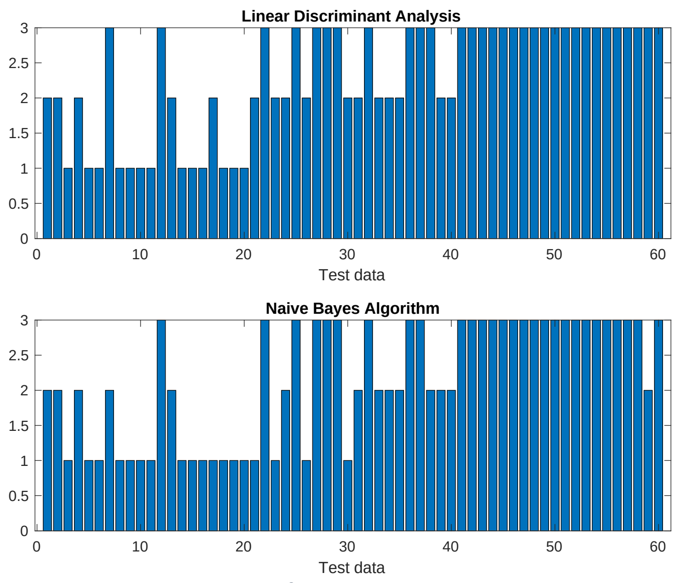

Figure 4: Test 3 results for LDA and Naive Bayes algorithms

V. Summary and Conclusions

This main goal of this project is to explore music classification, and there are three tests conducted. We will be performing three test cases for the music classification: Band Classification, the Case for Seattle and Genre Classification. In the first case, I will choose three different bands. For the second test, we chose three local 90s Seattle bands. The last test is for genre classification and we picked three different genres. For each test, to applied SVD for the matrix and generated the training set from matrix V. Then, we used two algorithms, LDA and

Naïve Bayes to estimates the test set. Unfortunately, we found our results to be not as accurate as we hoped. There is definitely space for improvements. A few possibilities that may have caused the low accuracy is the music piece. They might not be distinguishable enough to compare differences. In addition, we can also explore different training set proportions for future studies.

VI. Appendix A – MATLAB functions

`abs(X)` – calculates the absolute value of the input element. If input in complex, then returns the complex magnitude. This was used to normalize our dataset.

`audioread`– load data from MAT-file into the tensor. We used this to load the dataset into our workspace.

`randperm(X)` – generates a random number with the given value

`svd(A)`- performs a singular value decomposition to the x y matrix.

`plot()`- plot the dataset of the clusters and different algorithm histograms

VII. Appendix B – MATLAB codes

```
%% Load in Data
clc;close; clear all

testd = [];
for str={"Michael", "Beethoven", "Alice"}
    for i=1:3
        [y, Fs]=audioread(strcat(str{1},num2str(i),".mp3"));
        y = y'/ 2;
        y = y(1,:) + y(2,:);
        for j = 40:5:160
            test = y(1, Fs*j : Fs*(j+5));
            perm = abs(spectrogram(test));
            perm = reshape(perm, [1, 8*32769]);
            testd = [testd;perm];
        end
    end
end
testd = testd';
%% SVD
[U, S, V] = svd(testd - mean(testd(:)), 'econ');
figure(1)
plot(diag(S) ./ sum(diag(S)), 'bo');

true = [ones(20,1); 2*ones(20,1); 3*ones(20,1)];
rp1 = randperm(50);
rp2 = randperm(50);
rp3 = randperm(50);
michael = V(1:50, 2:4);
beethoven = V(51:100, 2:4);
alice = V(101:150, 2:4);
train = [michael(rp1(1:30), :); beethoven(rp2(1:30), :); alice(rp3(1:30),:)];
test1 = [michael(rp1(31:end), :); beethoven(rp2(31:end), :);
alice(rp3(31:end),:)];

% naive bayes
ctrain = [ones(30,1); 2*ones(30,1); 3*ones(30,1)];
nb = fitcnb(train, ctrain);
est = nb.predict(test1);
temp = [est== true];
bayes = sum(temp) / length(temp);

figure(2)
subplot(2,1,1);
bar(est)
title('Naive Bayes Algorithm');
xlabel('Test data');

% classify (Built in)
est = classify(test1, train, ctrain);
temp = [est== true];
lda = sum(temp) / length(temp);

subplot(2,1,2);
```

```
bar(est);
title('Linear Discriminant Analysis ');
xlabel('Test data');
% final results
bayes = mean(bayes);
lda = mean(lda);
result = [bayes;lda]
```