Music Classification

Jiaqi Zhu
Amath 482, Winter 2020

March 12, 2020

Abstract

This project aims to use SVD, machine learning algroithms, Naive Bayes models to train and classify five-seconds short clips of music by different artists of different genres. Then we could finally build a classifer to identify music based on different musiciens and genres.

I. Introduction and Overview

This project mainly focuses on the music classfication. There are three tests in this project. The first test is the classfication among three different bands of different genres. I chose works of Beethoven, Michael Jackson and Soundgarden. The second one focuses on the classifcaition among three different bands but of the same genre. I chose works of Derek Clegg, Miller and Sasser and Thorn Shout which are representative artistes of Country music. The thirs test is the classification among various bands within each of the three genres. I chose three different genres Classical (works of Beethoven, Alan Piljak and Yakov Golman), Hiphop (works of Ebsa, Jonas and Scott Nice) and Country (works of Derek Clegg, Miller and Sasser and Thorn Shout).For each test, I choose to use Naive Bayes model to identify music from different musiciens and genres. After the whole procedure, we intend to build a classifer which can identify the music of different artisits from different genres based on their frequencies through supervised learning.

II. Theoretical Background

Singular Value Decomposition (SVD)

A singular value decomposition (SVD) is a factorization of a matrix into the product of three matrices. In linear algebra, SVD of a matrix A could be written as

$$A = U\mathbf{\Sigma}V \tag{1}$$

where both U and V are unitary matrices and $\mathbf{\Sigma}$ is a rectangular diagonal matrix.

Naive Bayes Model

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Naive Bayes is a technique for supervised learning. It is based on applying Bayes' theorems with strong (naïve) independence assumptions between the features and the computation of conditional probabilities. A Naive Bayes model can predict the label of a new dataset based on the analysis of the given dataset. One advantage of naive bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

III. Algorithm Implementation and Development

Note that there are three tests in this project but the algorithm impementation for each test is basically the same. The only difference is that each test will use different datasets to analyze. The first step is to load the music file of each artists I chose and divided each song into several 5-seconds clips and labeled each one with its corresponding artist and music genre. Then, I will apply the spectrogram method to create the spectrogram of each 5-seconds music clip of each artist. After adding each spectrogram into a data matrix V, I performed SVD on the data matrix to get dominiant modes associated with bands, in other words, is to extract the significant feature or characteristic of each artist. Then we randomly splitted the data into 50% for training sets and 50% for testing sets. Next, I will perform Naive Bayes model to train the model and predict the testing data and then compared the predicted label with the given label to compute the accuracy of the testing data.

IV. Computational Results

Test 1

For the test1, we basically focus on three different bands of different genres. I chose works of Beethoven, Michael Jackson and Soundgarden. I divided the testing and training data in half and applied a Naive Bayes model to compute the accuracy. We got the accuracy of the Naive Bayes model is 69.89%, which is pretty high. One possible reason is that the difference between the genres and features of these three different musiciens are very significant, so it is very to easy to predict the music.

Test 2

For the test2, we basically focus on three different bands from the same genre. I chose works of Derek Clegg, Miller and Sasser and Thorn Shout which are representative artistes of Country music. I did the same procedure as Test 1 and the accuracy of the Naive Bayes model is 62.5%, which is lower than the first test. One possible reason is that within the same genre, the music of different bands may have some similar features so that it's a little difficult to distinguish the music between musiciens within one same genre.

Test 3

For the test3, we basically focus on various bands within each of three different genres. I chose three different genres Classical (works of Beethoven, Alan Piljak and Yakov Golman), Hiphop (works of Ebsa, Jonas and Scott Nice) and Country (works of Derek Clegg, Miller and Sasser and Thorn Shout). I did the same procedure as Test 1 and the accuracy of the Naive Bayes model is 63.33%. We can see that the accuracy is lower than test 1 but higher than test 2. One possible reason is that music of different genres is supposed to have very significant difference; however, the singers may share some similar music styles with the others within the same genres so that it would be harder than test 1 but a little bit easier than test 2.

V. Summary and Conclusions

This project aims to build a classifer which can identify the music of different artisits from different genres based on their frequencies through supervised learning. I choose Naive Bays model to conduct all three tests. Among all three tests we did, we can see that the accuracy of test 1 is highest of all and test 2 has the lowest accuracy, which is reasonable because for the first test, we focus on three different bands from different genres, the difference between each genre and artist is very obvious; for the second time, we basically focus on three different bands from the same genre, musiciens with same genre may share some similar music style or characteristic, thus making the prediction a little bit harder. Note that, we only use various 5 -seconds music clips, which is a relatively small amount data to train the model. If we use a longer-time peice of music, the accuracy may be higher because we can gather and collect more concrete and comprehensive features of different music to do the prediction.

Appendix A. MATLAB functions used and brief implementation explanation

dir: dir('Beethoven') returns the attributes of the file named Beethoven.

audioread:$[Y, FS[= audioread(ab)$ reads the audio file named ab and returns data Y and frequency of its audio file FS.

randperm:randperm(n) returns a row vector containing a random permutation of the integers from 1 to n without repeating elements.

spectrogram: spectrogram(A(i,:)) returns the short-time Fourier transform of the input signal, (A(i,:)).

fitcnb: classifier = fitcnb(real(xtrain),ytrain) returns a multiclass naive Bayes model.

predict: predict(classifier, real(xtest)) returns a vector of predicted class labels for the predictor data in the table or matrix, based on the trained discriminant analysis classification model.

Appendix B. MATLAB codes

```
clear; close all; clc;
%%
folderb = dir('Beethoven');
Ab = [];
for i = 4:length(folderb)
    [Y,FS] = audioread(['Beethoven/', folderb(i).name]);
    Ab = [Ab;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        Ab=[Ab;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

folderSG = dir('SG');
ASG = [];
for i = 3:length(folderSG)
    [Y,FS] = audioread(['SG/', folderSG(i).name]);
    ASG = [ASG;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        ASG=[ASG;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

folderMJ = dir('MJ');
AMJ = [];
for i = 4:length(folderMJ)
    [Y,FS] = audioread(['MJ/', folderMJ(i).name]);
    AMJ = [AMJ;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AMJ=[AMJ;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

n = 124;
Ab=Ab(1:n,:); Ab=Ab(randperm(n),:);
AMJ=AMJ(1:n,:);AMJ=AMJ(randperm(n),:);
ASG=ASG(1:n,:);ASG=ASG(randperm(n),:);
A=[Ab;AMJ;ASG];

spec = [];
for i = 1:size(A,1)
    s = spectrogram(A(i,:));
    spec(i,:) = s(:);
end

[u,s,v] = svd(A - mean(A(:)), 'econ');
%plot(diag(s)/sum(diag(s)),'ro')
%xlabel('Singular Values')
%ylabel('Energey(%)')
%title('Test 1 Singular Value Spectrum(Percentage)')


xtrain=[v(1:62,:);v(125:186,:);v(249:310,:)];
xtest=[v(63:124,:);v(187:248,:);v(311:372,:)];
ytrain=[zeros(62,1)+1;zeros(62,1)+2;zeros(62,1)+3];
ytest=[zeros(62,1)+1;zeros(62,1)+2;zeros(62,1)+3];

classifier = fitcnb(real(xtrain),ytrain);
predicted = predict(classifier, real(xtest));
accuracy1 = sum(predicted==ytest)/length(ytest);

%% test 2
folderM = dir('Miller');
AM = [];
for i = 3:length(folderM)
    [Y,FS] = audioread(['Miller/', folderM(i).name]);
    AM = [AM;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AM=[AM;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

folderD = dir('Derek');
AD = [];
for i = 3:length(folderD)
    [Y,FS] = audioread(['Derek/', folderD(i).name]);
    AD = [AD;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AD=[AD;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

folderT = dir('Thorn');
AT = [];
```

```
for i = 3:length(folderT)
    [Y,FS] = audioread(['Thorn/', folderT(i).name]);
    AT = [AT;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AT=[AT;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

n = 96;
AM=AM(1:n,:); AM=AM(randperm(n),:);
AD=AD(1:n,:);AD=AD(randperm(n),:);
AT=AT(1:n,:);AT=AT(randperm(n),:);
A=[AM;AD;AT];

spec = [];
for i = 1:size(A,1)
    s = spectrogram(A(i,:));
    spec(i,:) = s(:);
end

[u,s,v] = svd(A - mean(A(:)), 'econ');


xtrain=[v(1:16,:);v(33:48,:);v(65:80,:)];
xtest=[v(17:32,:);v(49:64,:);v(81:96,:)];
ytrain=[zeros(16,1)+1;zeros(16,1)+2;zeros(16,1)+3];
ytest=[zeros(16,1)+1;zeros(16,1)+2;zeros(16,1)+3];

classifier = fitcnb(real(xtrain),ytrain);
predicted = predict(classifier, real(xtest));
accuracy2 = sum(predicted==ytest)/length(ytest);

%% TEST 3
folderC = dir('Classical');
AC = [];
for i = 4:length(folderC)
    [Y,FS] = audioread(['Classical/', folderC(i).name]);
    AC = [AC;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AC=[AC;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end
```

```
folderCO = dir('Country');
ACO = [];
for i = 3:length(folderCO)
    [Y,FS] = audioread(['Country/', folderCO(i).name]);
    ACO = [ACO;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        ACO=[ACO;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

folderH = dir('Hiphop');
AH = [];
for i = 3:length(folderH)
    [Y,FS] = audioread(['Hiphop/', folderH(i).name]);
    AH = [AH;Y(1:FS*5)];
    for jj=2:length(Y)/FS/5-1
        AH=[AH;Y(FS*5*(jj-1)+1:FS*5*jj)];
    end
end

n = 240;
AC=AC(1:n,:); AC=AC(randperm(n),:);
ACO=ACO(1:n,:);ACO=ACO(randperm(n),:);
AH=AH(1:n,:);AH=AH(randperm(n),:);
A=[AC;ACO;AH];

spec = [];
for i = 1:size(A,1)
    s = spectrogram(A(i,:));
    spec(i,:) = s(:);
end

[u,s,v] = svd(A - mean(A(:)), 'econ');


xtrain=[v(1:40,:);v(81:120,:);v(161:200,:)];
xtest=[v(41:80,:);v(121:160,:);v(201:240,:)];
ytrain=[zeros(40,1)+1;zeros(40,1)+2;zeros(40,1)+3];
ytest=[zeros(40,1)+1;zeros(40,1)+2;zeros(40,1)+3];

classifier = fitcnb(real(xtrain),ytrain);
```

```
predicted = predict(classifier, real(xtest));
accuracy3 = sum(predicted==ytest)/length(ytest);
```

Reference

Kutz, Jose Nathan. Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data. Oxford University Press, 2013.