Machine Learning

Hoyt Thomas

3/6/20

I. Abstract

A method of Machine Learning is demonstrated to classify music using spectrograms, Principle Component Analysis, and Linear Discriminate Analysis.

II. Introduction

Machine learning is an alternate approach to data analysis where predictions and classifications are made from learned or inferred trends. There exist two main categories of machine learning: supervised and unsupervised. Supervised learning takes training set of data in which the data is known and classified beforehand for code to analyze with goal of being able to classify new data. In unsupervised learning is the code itself learns to classify and group data.

Supervised machine learning will be used to classify music based upon three different input scenarios. In the first test the training data will include music from three different artists each from a different genre and then asked to determine the artist given a song from one of the artists. The second test is similar to the first except the three artists are from the same genre to make it more difficult. Finally, the third test will use multiple artists within three genres as the training data then test whether the function can correctly determine the genre of music piece from the given genres.

III. Theoretical Background

The theoretical background for the assignment can be broken into three sections: creating spectrograms of the music clips using Short Time Fourier Transform (STFT), Principle Component Analysis (PCA) to see how the clips differ from each other, and Linear Discriminate Analysis (LCA) to find a threshold to separate the groups. Spectrograms will give us a distinct signature for each clip, the STFT create ‘windows’ in the time domain, where only the signal inside is considered. Within each window the FFT is applied, then the window is slid across the signal to capture the entire range of frequencies in the signal. The STFT is defined as:

$$\tilde{f}(\tau,\omega) = \int_{-\infty}^{\infty} f(t)g(t-\tau)e^{-i\omega t}dt \text{ (EQ. 1)}$$

where f(t) is the signal, g(t-τ) is the filter function, and $e^{-i\omega t}$ is the Fourier transform. The filter function creates the window for the FFT to be performed, a real and symmetric function is needed for the filter. The most common filter used is a Gaussian filter:

$$g(t-\tau) = e^{-a(t-\tau)} \text{ (EQ. 2)}$$

where a controls the width of the gaussian window and τ controls where the center of gaussian window is in time. Unlike spectral filtering, this filter is applied to the signal in the time the STFT filters the signal in the time domain.

PCA uses singular value decomposition (SVD) to diagonalize a matrix or reduce a matrix to its *ideal* basis. The SVD of matrix, A, is defined as:

$$A = U\Sigma V^* \text{ (EQ 3)}$$

The ideal basis is given by the singular values, which are the diagonal values in the sigma matrix. Singular values tell us how much variance is in each of the dimensions, the number of non-zero singular values is the rank of A and are ordered from largest to smallest. The principle components are contained in the columns of U or the left singular vectors.

Once we have our Principle Components we will use LDA to determine a projection which minimizes the intra-class variation of PCs while maximizing the inter-class variation of PCs. This projection will help us determine thresholds for classifying groups. The intraclass variation is given by:

$$S_w = \sum_{j=1}^{n}(x - \mu_j)(x - \mu_j)^T \text{(EQ. 4)}$$

where n is the number of clips in each group, x is the matrix of PCs, mu_j is the mean PC of the jth clip. The interclass variation is given by:

$$S_B = \sum_{j=1}^{n}(\mu_j - \mu)(\mu_j - \mu)^T \text{(EQ. 5)}$$

where n is the number of groups being tested, mu_j is the mean PC of the jth group, and mu is mean PC across all groups. By maximizing a ratio of Sb to Sw a projection is found to see the how the groups are most separated.

IV. Algorithm Development

To start, load in the audio clips and downsample the signal by a factor of 4 to reduce the computational load of the code. A function was created to return a matrix of spectrograms given a matrix of audio signals of the same length. This function (band_spec) used the built-in Matlab spectrogram feature (EQ. 1&2), took the absolute values of the spectrogram to eliminate complex numbers from the array, and then was re-shaped into a column vector.

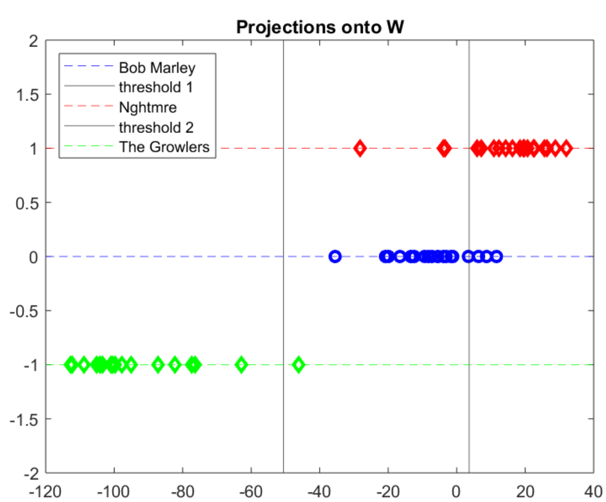

Figure 1: Test 1 Projections of PC onto W

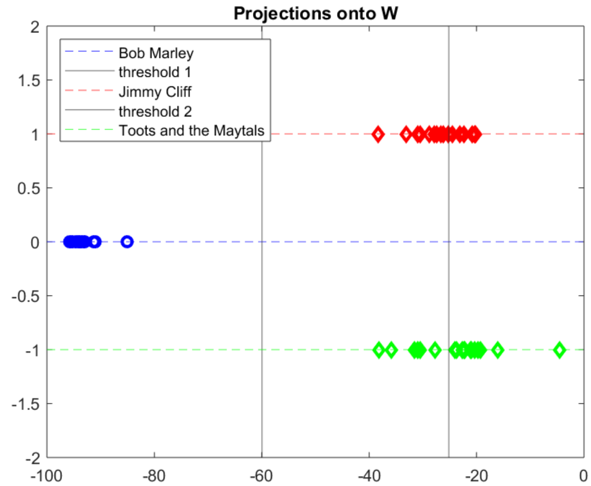

Figure 2: Test 2 Projection of PC on to W

Once spectrograms of the three each training groups were completed, they were passed with the number of features to the training function to determine optimal projection and threshold values. This function(band_trainer) starts by conducting an SVD on an array of all the spectrograms (EQ. 3). The Principle components are reconstructed by multiplying sigma by V' and separated into out previous groups again. Means of each group and all groups were calculated to pass to inter-class and intra-class variation (EQ. 4&5). The optimal projection was found by eigen-decomposing Sw, Sb, finding the largest non-zero eigen-value and using the associated eigen vector. This eigenvector is normalized by the L-2 norm to become our final projection(w). The PCs are then projected onto our vector (w) by multiplying PCs w' to get a numerical representation of the groups(vband). To visualize how groups are clustered each group was plotted on its own axis (-1,0,1) along with the threshold values (Fig. 1,2,3).

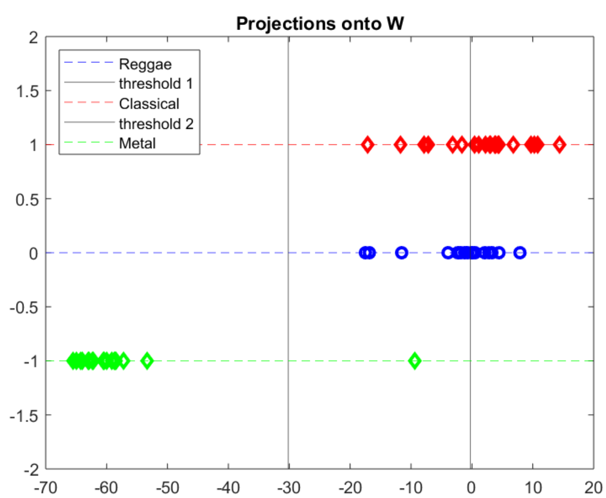

Figure 3: Test 3 Projection of PC on to W

Threshold values were determined by taking the means of each group and finding the midpoint between those means. The function outputs: projection(w), threshold values, and sorted vbands.

The thresholds were tested by creating spectrograms of new clips belonging to the three groups using band_spec. The PCs were computed and projected onto w' and using threshold values, classified.

V. Computational Results

For test 1 20 clips were each taken from songs by Bob Marley, Nghtmre, and The Growlers. The classifier was then tested against 3 clips from each artist (9 total) and had 44% success rate of classifying these clips correctly.

For test 2 20 clips were each taken from songs by Bob Marley, Jimmy Cliff, and Toots and the Maytals. The classifier was then tested against 3 clips from each artist (9 total) and had 44% success rate of classifying these clips correctly.

For test 3 20 clips were each taken from songs from Classical, Metal, and Reggae artists. The classifier was then tested against 3 clips from each genre (9 total) and had a 66% success rate of classifying these clips correctly.

VI. Conclusion

Machine Learning can be a powerful tool to save time and create a method to classify music. The classifier was not accurate in classifying new clips, the accuracy could be improved by increasing the amount clips in the training data and trying different methods of determining threshold values.

Appendix A: Matlab Commands

downsample(y,F) – downsamples signal (y) by factor of F

[s,w,t] = spectrogram(x,window,noverlap,w) – creates spectrogram using STFT

Inputs- x- signal, window- width of gabor window, overlap-overlap of gabor window

Appendix B: Code

```
%HW4 - AMATH 482-Test 1
%Machine Learning
clear all;close all;clc;

%load and sort training data
%Bob Marley
[y_1,~] = audioread('bob_clip_1.wav');
y_1_1 = y_1(1:220500,1);
reg_1 = downsample(y_1_1,4);
```

```
[y_2,~] = audioread('bob_clip_2.wav');
y_2_1 = y_2(1:220500,1);
reg_2 = downsample(y_2_1,4);

[y_3,~] = audioread('bob_clip_3.wav');
y_3_1 = y_3(1:220500,1);
reg_3 = downsample(y_3_1,4);

[y_4,~] = audioread('bob_clip_4.wav');
y_4_1 = y_4(1:220500,1);
reg_4 = downsample(y_4_1,4);

[y_5,~] = audioread('bob_clip_5.wav');
y_5_1 = y_5(1:220500,1);
reg_5 = downsample(y_5_1,4);

[y_6,~] = audioread('bob_clip_6.wav');
y_6_1 = y_6(1:220500,1);
reg_6 = downsample(y_6_1,4);

[y_7,~] = audioread('bob_clip_7.wav');
y_7_1 = y_7(1:220500,1);
reg_7 = downsample(y_7_1,4);

[y_8,~] = audioread('bob_clip_8.wav');
y_8_1 = y_8(1:220500,1);
reg_8 = downsample(y_8_1,4);

[y_9,~] = audioread('bob_clip_9.wav');
y_9_1 = y_9(1:220500,1);
reg_9 = downsample(y_9_1,4);

[y_10,~] = audioread('bob_clip_10.wav');
y_10_1 = y_10(1:220500,1);
reg_10 = downsample(y_10_1,4);

[y_11,~] = audioread('bob_clip_11.wav');
y_11_1 = y_11(1:220500,1);
reg_11 = downsample(y_11_1,4);

[y_12,~] = audioread('bob_clip_12.wav');
y_12_1 = y_12(1:220500,1);
reg_12 = downsample(y_12_1,4);

[y_13,~] = audioread('bob_clip_13.wav');
y_13_1 = y_13(1:220500,1);
reg_13 = downsample(y_13_1,4);
```

```
[y_14,~] = audioread('bob_clip_14.wav');
y_14_1 = y_14(1:220500,1);
reg_14 = downsample(y_14_1,4);

[y_15,~] = audioread('bob_clip_15.wav');
y_15_1 = y_15(1:220500,1);
reg_15 = downsample(y_15_1,4);

[y_16,~] = audioread('bob_clip_16.wav');
y_16_1 = y_16(1:220500,1);
reg_16 = downsample(y_16_1,4);

[y_17,~] = audioread('bob_clip_17.wav');
y_17_1 = y_17(1:220500,1);
reg_17 = downsample(y_17_1,4);

[y_18,~] = audioread('bob_clip_18.wav');
y_18_1 = y_18(1:220500,1);
reg_18 = downsample(y_18_1,4);

[y_19,~] = audioread('bob_clip_19.wav');
y_19_1 = y_19(1:220500,1);
reg_19 = downsample(y_19_1,4);

[y_20,~] = audioread('bob_clip_20.wav');
y_20_1 = y_20(1:220500,1);
reg_20 = downsample(y_20_1,4);

reg =
[reg_1,reg_2,reg_3,reg_4,reg_5,reg_6,reg_7,reg_8,reg_9,reg_10,..
.

reg_11,reg_12,reg_13,reg_14,reg_15,reg_16,reg_17,reg_18,reg_19,r
eg_20];

%Nghtmre
[x_1,~] = audioread('nght_clip_1.wav');
x_1_1 = x_1(1:220500,1);
dub_1 = downsample(x_1_1,4);

[x_2,~] = audioread('nght_clip_2.wav');
x_2_1 = x_2(1:220500,1);
dub_2 = downsample(x_2_1,4);

[x_3,~] = audioread('nght_clip_3.wav');
x_3_1 = x_3(1:220500,1);
```

```
dub_3 = downsample(x_3_1,4);

[x_4,~] = audioread('nght_clip_4.wav');
x_4_1 = x_4(1:220500,1);
dub_4 = downsample(x_4_1,4);

[x_5,~] = audioread('nght_clip_5.wav');
x_5_1 = x_5(1:220500,1);
dub_5 = downsample(x_5_1,4);

[x_6,~] = audioread('nght_clip_6.wav');
x_6_1 = x_6(1:220500,1);
dub_6 = downsample(x_6_1,4);

[x_7,~] = audioread('nght_clip_7.wav');
x_7_1 = x_7(1:220500,1);
dub_7 = downsample(x_7_1,4);

[x_8,~] = audioread('nght_clip_8.wav');
x_8_1 = x_8(1:220500,1);
dub_8 = downsample(x_8_1,4);

[x_9,~] = audioread('nght_clip_9.wav');
x_9_1 = x_9(1:220500,1);
dub_9 = downsample(x_9_1,4);

[x_10,~] = audioread('nght_clip_10.wav');
x_10_1 = x_10(1:220500,1);
dub_10 = downsample(x_10_1,4);

[x_11,~] = audioread('nght_clip_11.wav');
x_11_1 = x_11(1:220500,1);
dub_11 = downsample(x_11_1,4);

[x_12,~] = audioread('nght_clip_12.wav');
x_12_1 = x_12(1:220500,1);
dub_12 = downsample(x_12_1,4);

[x_13,~] = audioread('nght_clip_13.wav');
x_13_1 = x_13(1:220500,1);
dub_13 = downsample(x_13_1,4);

[x_14,~] = audioread('nght_clip_14.wav');
x_14_1 = x_14(1:220500,1);
dub_14 = downsample(x_14_1,4);

[x_15,~] = audioread('nght_clip_15.wav');
```

```
x_15_1 = x_15(1:220500,1);
dub_15 = downsample(x_15_1,4);

[x_16,~] = audioread('nght_clip_16.wav');
x_16_1 = x_16(1:220500,1);
dub_16 = downsample(x_16_1,4);

[x_17,~] = audioread('nght_clip_17.wav');
x_17_1 = x_17(1:220500,1);
dub_17 = downsample(x_17_1,4);

[x_18,~] = audioread('nght_clip_18.wav');
x_18_1 = x_18(1:220500,1);
dub_18 = downsample(x_18_1,4);

[x_19,~] = audioread('nght_clip_19.wav');
x_19_1 = x_19(1:220500,1);
dub_19 = downsample(x_19_1,4);

[x_20,~] = audioread('nght_clip_20.wav');
x_20_1 = x_20(1:220500,1);
dub_20 = downsample(x_20_1,4);

dub =
[dub_1,dub_2,dub_3,dub_4,dub_5,dub_6,dub_7,dub_8,dub_9,dub_10,..
.

dub_11,dub_12,dub_13,dub_14,dub_15,dub_16,dub_17,dub_18,dub_19,d
ub_20];

%The Growlers
[z_1,~] = audioread('growl_clip_1.wav');
z_1_1 = z_1(1:220500,1);
surf_1 = downsample(z_1_1,4);

[z_2,~] = audioread('growl_clip_2.wav');
z_2_1 = z_2(1:220500,1);
surf_2 = downsample(z_2_1,4);

[z_3,~] = audioread('growl_clip_3.wav');
z_3_1 = z_3(1:220500,1);
surf_3 = downsample(z_3_1,4);

[z_4,~] = audioread('growl_clip_4.wav');
z_4_1 = z_4(1:220500,1);
surf_4 = downsample(z_4_1,4);
```

```
[z_5,~] = audioread('growl_clip_5.wav');
z_5_1 = z_5(1:220500,1);
surf_5 = downsample(z_5_1,4);

[z_6,~] = audioread('growl_clip_6.wav');
z_6_1 = z_6(1:220500,1);
surf_6 = downsample(z_6_1,4);

[z_7,~] = audioread('growl_clip_7.wav');
z_7_1 = z_7(1:220500,1);
surf_7 = downsample(z_7_1,4);

[z_8,~] = audioread('growl_clip_8.wav');
z_8_1 = z_8(1:220500,1);
surf_8 = downsample(z_8_1,4);

[z_9,~] = audioread('growl_clip_9.wav');
z_9_1 = z_9(1:220500,1);
surf_9 = downsample(z_9_1,4);

[z_10,~] = audioread('growl_clip_10.wav');
z_10_1 = z_10(1:220500,1);
surf_10 = downsample(z_10_1,4);

[z_11,~] = audioread('growl_clip_11.wav');
z_11_1 = z_11(1:220500,1);
surf_11 = downsample(z_11_1,4);

[z_12,~] = audioread('growl_clip_12.wav');
z_12_1 = z_12(1:220500,1);
surf_12 = downsample(z_12_1,4);

[z_13,~] = audioread('growl_clip_13.wav');
z_13_1 = z_13(1:220500,1);
surf_13 = downsample(z_13_1,4);

[z_14,~] = audioread('growl_clip_14.wav');
z_14_1 = z_14(1:220500,1);
surf_14 = downsample(z_14_1,4);

[z_15,~] = audioread('growl_clip_15.wav');
z_15_1 = z_15(1:220500,1);
surf_15 = downsample(z_15_1,4);

[z_16,~] = audioread('growl_clip_16.wav');
z_16_1 = z_16(1:220500,1);
surf_16 = downsample(z_16_1,4);
```

```
[z_17,~] = audioread('growl_clip_17.wav');
z_17_1 = z_17(1:220500,1);
surf_17 = downsample(z_17_1,4);

[z_18,~] = audioread('growl_clip_18.wav');
z_18_1 = z_18(1:220500,1);
surf_18 = downsample(z_18_1,4);

[z_19,~] = audioread('growl_clip_19.wav');
z_19_1 = z_19(1:220500,1);
surf_19 = downsample(z_19_1,4);

[z_20,Fs] = audioread('growl_clip_20.wav');
z_20_1 = z_20(1:220500,1);
surf_20 = downsample(z_20_1,4);

surf =
[surf_1,surf_2,surf_3,surf_4,surf_5,surf_6,surf_7,surf_8,surf_9,
surf_10,...

surf_11,surf_12,surf_13,surf_14,surf_15,surf_16,surf_17,surf_18,
surf_19,surf_20];



%%
feature = 45;
bobdata = band_spec(reg,Fs);
nghtdata = band_spec(dub,Fs);
growldata = band_spec(surf,Fs);

[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(bobdata,nghtdata,growldata,feature);

%%

figure(2)
subplot(2,1,1)
plot(diag(S),'ko','Linewidth',2)
subplot(2,1,2)
semilogy(diag(S),'ko','Linewidth',2)
%%
figure(3)
for k=1:3
    subplot(3,3,3*k-2)
```

```
    plot(1:20,V(1:20,k),'ko-')
    subplot(3,3,3*k-1)
    plot(21:40,V(21:40,k),'ko-')
    subplot(3,3,3*k)
    plot(41:60,V(41:60,k),'ko-')
end
subplot(3,3,1), set(gca,'Fontsize',12), title('Bob')
subplot(3,3,2), set(gca,'Fontsize',12), title('Nghtmre')
subplot(3,3,3), set(gca,'Fontsize',12), title('Growlers')

%% Projection onto W
figure(4)
plot(sortband1,zeros(20),'ob','Linewidth',2,'HandleVisibility','
off')
yline(0,'b--','DisplayName','Bob Marley');
xline(threshold_1,'DisplayName','threshold 1');
hold on
plot(sortband2,ones(20),'dr','Linewidth',2,'HandleVisibility','o
ff')
yline(1,'r--','DisplayName','Nghtmre');
xline(threshold_2,'DisplayName','threshold 2');
hold on
plot(sortband3,ones(20)*-
1,'gd','Linewidth',2,'HandleVisibility','off')
yline(-1,'g--','DisplayName','The Growlers');
ylim([-2 2]);
legend('Location','northwest');
title('Projections onto W');
%% Histogram
figure(5)
subplot(1,3,1)
histogram(sortband1,15);
title('Bob');
subplot(1,3,2)
histogram(sortband2,15);
title('Nghtmre');
subplot(1,3,3)
histogram(sortband3,15);
title('Growlers');

%% Load test data
[t_1,~] = audioread('bob_clip_21.wav');
t_1_1 = t_1(1:220500,1);
test_1 = downsample(t_1_1,4);

[t_2,~] = audioread('bob_clip_22.wav');
t_2_1 = t_2(1:220500,1);
```

```
test_2 = downsample(t_2_1,4);

[t_3,~] = audioread('bob_clip_23.wav');
t_3_1 = t_3(1:220500,1);
test_3 = downsample(t_3_1,4);

[t_4,~] = audioread('nght_clip_21.wav');
t_4_1 = t_4(1:220500,1);
test_4 = downsample(t_4_1,4);

[t_5,~] = audioread('nght_clip_22.wav');
t_5_1 = t_5(1:220500,1);
test_5 = downsample(t_5_1,4);

[t_6,~] = audioread('nght_clip_23.wav');
t_6_1 = t_6(1:220500,1);
test_6 = downsample(t_6_1,4);

[t_7,~] = audioread('growl_clip_21.wav');
t_7_1 = t_7(1:220500,1);
test_7 = downsample(t_7_1,4);

[t_8,~] = audioread('growl_clip_22.wav');
t_8_1 = t_8(1:220500,1);
test_8 = downsample(t_8_1,4);

[t_9,Fs] = audioread('growl_clip_23.wav');
t_9_1 = t_9(1:220500,1);
test_9 = downsample(t_9_1,4);

test =
[test_1,test_2,test_3,test_4,test_5,test_6,test_7,test_8,test_9]
;

testData = band_spec(test,Fs);
testpval = zeros(1,9);
for k = 1:9
 testPCA = U'*testData(:,k);
 testpval(:,k) = w'*testPCA(1:feature,:);
end

for k = 1:9
     if testpval(:,k) < threshold_1
         disp('The Growlers')
     elseif threshold_1 < testpval(:,k) < threshold_2
         disp('Bob Marley')
     elseif testpval(:,k) > threshold_2
```

```
        disp('Nghtmre')
    end
end
%result: 4/9 correct - 44.4% success
%%
function bandData = band_spec(bandfile,Fs)

    [~,n] = size(bandfile);
    [Spc] = spectrogram(bandfile(:,1),302,150,302,'yaxis',
Fs/4);
    [~,y] = size(Spc);
    bandData = zeros(60*y,n);

    for k = 1:n
       [Sp, ~, ~] =
spectrogram(bandfile(:,k),302,150,302,'yaxis', Fs/4);
       Gr = abs(Sp(1:60,:));
       [l,w] = size(Gr);
       bandData(:,k) = reshape(Gr,l*w,1);
    end

end

function
[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(band1_S,band2_S,band3_S,feature)

    [U,S,V] = svd([band1_S band2_S band3_S], 'econ');
    nband1 = size(band1_S,2); nband2 = size(band2_S,2); nband3 =
size(band3_S,2);
    bands = S*V';
    band1 = bands(1:feature,1:nband1);
    band2 = bands(1:feature,nband1+1:nband1+nband2);
    band3 =
bands(1:feature,nband1+nband2+1:nband1+nband2+nband3);

    mband1 = mean(band1,2);
    mband2 = mean(band2,2);
    mband3 = mean(band3,2);

    Sw = 0; % within class variances
    for k=1:nband1
        Sw = Sw + (band1(:,k)-mband1)*(band1(:,k)-mband1)';
    end
    for k=1:nband2
        Sw = Sw + (band2(:,k)-mband2)*(band2(:,k)-mband2)';
```

```
    end
    for k=1:nband3
        Sw = Sw + (band3(:,k)-mband3)*(band3(:,k)-mband3)';
    end

    mbands = mean(bands(1:feature,:),2);
    means = [mband1,mband2,mband3];
    Sb = 0; %between class variances
    for j=1:3
        Sb = 20*(means(:,j)-mbands)*(means(:,j)-mbands)';
    end

    [V2,D] = eig(Sb,Sw); % linear discriminant analysis
    [~,ind] = max(abs(diag(D)));
    w = V2(:,ind); w = w/norm(w,2);

    vband1 = w'*band1;
    vband2 = w'*band2;
    vband3 = w'*band3;

    sortband1 = sort(vband1);
    sortband2 = sort(vband2);
    sortband3 = sort(vband3);

    msortband1 = mean(sortband1);
    msortband2 = mean(sortband2);
    msortband3 = mean(sortband3);

    msorts = [msortband1,msortband2,msortband3];
    msort = sort(msorts);
    threshold_1 = (msort(1)+msort(2))/2;
    threshold_2 = (msort(2)+msort(3))/2;

end


%HW4 - AMATH 482-Test 2
%Machine Learning
clear all;close all;clc;


%load and sort training data
%Bob Marley
[y_1,~] = audioread('bob_clip_1.wav');
y_1_1 = y_1(1:220500,1);
bob_1 = downsample(y_1_1,4);
```

```
[y_2,~] = audioread('bob_clip_2.wav');
y_2_1 = y_2(1:220500,1);
bob_2 = downsample(y_2_1,4);

[y_3,~] = audioread('bob_clip_3.wav');
y_3_1 = y_3(1:220500,1);
bob_3 = downsample(y_3_1,4);

[y_4,~] = audioread('bob_clip_4.wav');
y_4_1 = y_4(1:220500,1);
bob_4 = downsample(y_4_1,4);

[y_5,~] = audioread('bob_clip_5.wav');
y_5_1 = y_5(1:220500,1);
bob_5 = downsample(y_5_1,4);

[y_6,~] = audioread('bob_clip_6.wav');
y_6_1 = y_6(1:220500,1);
bob_6 = downsample(y_6_1,4);

[y_7,~] = audioread('bob_clip_7.wav');
y_7_1 = y_7(1:220500,1);
bob_7 = downsample(y_7_1,4);

[y_8,~] = audioread('bob_clip_8.wav');
y_8_1 = y_8(1:220500,1);
bob_8 = downsample(y_8_1,4);

[y_9,~] = audioread('bob_clip_9.wav');
y_9_1 = y_9(1:220500,1);
bob_9 = downsample(y_9_1,4);

[y_10,~] = audioread('bob_clip_10.wav');
y_10_1 = y_10(1:220500,1);
bob_10 = downsample(y_10_1,4);

[y_11,~] = audioread('bob_clip_11.wav');
y_11_1 = y_11(1:220500,1);
bob_11 = downsample(y_11_1,4);

[y_12,~] = audioread('bob_clip_12.wav');
y_12_1 = y_12(1:220500,1);
bob_12 = downsample(y_12_1,4);

[y_13,~] = audioread('bob_clip_13.wav');
y_13_1 = y_13(1:220500,1);
bob_13 = downsample(y_13_1,4);
```

```
[y_14,~] = audioread('bob_clip_14.wav');
y_14_1 = y_14(1:220500,1);
bob_14 = downsample(y_14_1,4);

[y_15,~] = audioread('bob_clip_15.wav');
y_15_1 = y_15(1:220500,1);
bob_15 = downsample(y_15_1,4);

[y_16,~] = audioread('bob_clip_16.wav');
y_16_1 = y_16(1:220500,1);
bob_16 = downsample(y_16_1,4);

[y_17,~] = audioread('bob_clip_17.wav');
y_17_1 = y_17(1:220500,1);
bob_17 = downsample(y_17_1,4);

[y_18,~] = audioread('bob_clip_18.wav');
y_18_1 = y_18(1:220500,1);
bob_18 = downsample(y_18_1,4);

[y_19,~] = audioread('bob_clip_19.wav');
y_19_1 = y_19(1:220500,1);
bob_19 = downsample(y_19_1,4);

[y_20,~] = audioread('bob_clip_20.wav');
y_20_1 = y_20(1:220500,1);
bob_20 = downsample(y_20_1,4);

bob =
[bob_1,bob_2,bob_3,bob_4,bob_5,bob_6,bob_7,bob_8,bob_9,bob_10,..
.

bob_11,bob_12,bob_13,bob_14,bob_15,bob_16,bob_17,bob_18,bob_19,b
ob_20];

%Jimmy Cliff
[x_1,~] = audioread('jim_clip_1.wav');
x_1_1 = x_1(1:220500,1);
jim_1 = downsample(x_1_1,4);

[x_2,~] = audioread('jim_clip_2.wav');
x_2_1 = x_2(1:220500,1);
jim_2 = downsample(x_2_1,4);

[x_3,~] = audioread('jim_clip_3.wav');
x_3_1 = x_3(1:220500,1);
```

```
jim_3 = downsample(x_3_1,4);

[x_4,~] = audioread('jim_clip_4.wav');
x_4_1 = x_4(1:220500,1);
jim_4 = downsample(x_4_1,4);

[x_5,~] = audioread('jim_clip_5.wav');
x_5_1 = x_5(1:220500,1);
jim_5 = downsample(x_5_1,4);

[x_6,~] = audioread('jim_clip_6.wav');
x_6_1 = x_6(1:220500,1);
jim_6 = downsample(x_6_1,4);

[x_7,~] = audioread('jim_clip_7.wav');
x_7_1 = x_7(1:220500,1);
jim_7 = downsample(x_7_1,4);

[x_8,~] = audioread('jim_clip_8.wav');
x_8_1 = x_8(1:220500,1);
jim_8 = downsample(x_8_1,4);

[x_9,~] = audioread('jim_clip_9.wav');
x_9_1 = x_9(1:220500,1);
jim_9 = downsample(x_9_1,4);

[x_10,~] = audioread('jim_clip_10.wav');
x_10_1 = x_10(1:220500,1);
jim_10 = downsample(x_10_1,4);

[x_11,~] = audioread('jim_clip_11.wav');
x_11_1 = x_11(1:220500,1);
jim_11 = downsample(x_11_1,4);

[x_12,~] = audioread('jim_clip_12.wav');
x_12_1 = x_12(1:220500,1);
jim_12 = downsample(x_12_1,4);

[x_13,~] = audioread('jim_clip_13.wav');
x_13_1 = x_13(1:220500,1);
jim_13 = downsample(x_13_1,4);

[x_14,~] = audioread('jim_clip_14.wav');
x_14_1 = x_14(1:220500,1);
jim_14 = downsample(x_14_1,4);

[x_15,~] = audioread('jim_clip_15.wav');
```

```
x_15_1 = x_15(1:220500,1);
jim_15 = downsample(x_15_1,4);

[x_16,~] = audioread('jim_clip_16.wav');
x_16_1 = x_16(1:220500,1);
jim_16 = downsample(x_16_1,4);

[x_17,~] = audioread('jim_clip_17.wav');
x_17_1 = x_17(1:220500,1);
jim_17 = downsample(x_17_1,4);

[x_18,~] = audioread('jim_clip_18.wav');
x_18_1 = x_18(1:220500,1);
jim_18 = downsample(x_18_1,4);

[x_19,~] = audioread('jim_clip_19.wav');
x_19_1 = x_19(1:220500,1);
jim_19 = downsample(x_19_1,4);

[x_20,~] = audioread('jim_clip_20.wav');
x_20_1 = x_20(1:220500,1);
jim_20 = downsample(x_20_1,4);

jim =
[jim_1,jim_2,jim_3,jim_4,jim_5,jim_6,jim_7,jim_8,jim_9,jim_10,..
.

jim_11,jim_12,jim_13,jim_14,jim_15,jim_16,jim_17,jim_18,jim_19,j
im_20];

%Toots and the Maytals

[z_1,~] = audioread('toot_clip_1.wav');
z_1_1 = z_1(1:220500,1);
toot_1 = downsample(z_1_1,4);

[z_2,~] = audioread('toot_clip_2.wav');
z_2_1 = z_2(1:220500,1);
toot_2 = downsample(z_2_1,4);

[z_3,~] = audioread('toot_clip_3.wav');
z_3_1 = z_3(1:220500,1);
toot_3 = downsample(z_3_1,4);

[z_4,~] = audioread('toot_clip_4.wav');
z_4_1 = z_4(1:220500,1);
toot_4 = downsample(z_4_1,4);
```

```
[z_5,~] = audioread('toot_clip_5.wav');
z_5_1 = z_5(1:220500,1);
toot_5 = downsample(z_5_1,4);

[z_6,~] = audioread('toot_clip_6.wav');
z_6_1 = z_6(1:220500,1);
toot_6 = downsample(z_6_1,4);

[z_7,~] = audioread('toot_clip_7.wav');
z_7_1 = z_7(1:220500,1);
toot_7 = downsample(z_7_1,4);

[z_8,~] = audioread('toot_clip_8.wav');
z_8_1 = z_8(1:220500,1);
toot_8 = downsample(z_8_1,4);

[z_9,~] = audioread('toot_clip_9.wav');
z_9_1 = z_9(1:220500,1);
toot_9 = downsample(z_9_1,4);

[z_10,~] = audioread('toot_clip_10.wav');
z_10_1 = z_10(1:220500,1);
toot_10 = downsample(z_10_1,4);

[z_11,~] = audioread('toot_clip_11.wav');
z_11_1 = z_11(1:220500,1);
toot_11 = downsample(z_11_1,4);

[z_12,~] = audioread('toot_clip_12.wav');
z_12_1 = z_12(1:220500,1);
toot_12 = downsample(z_12_1,4);

[z_13,~] = audioread('toot_clip_13.wav');
z_13_1 = z_13(1:220500,1);
toot_13 = downsample(z_13_1,4);

[z_14,~] = audioread('toot_clip_14.wav');
z_14_1 = z_14(1:220500,1);
toot_14 = downsample(z_14_1,4);

[z_15,~] = audioread('toot_clip_15.wav');
z_15_1 = z_15(1:220500,1);
toot_15 = downsample(z_15_1,4);

[z_16,~] = audioread('toot_clip_16.wav');
z_16_1 = z_16(1:220500,1);
```

```
toot_16 = downsample(z_16_1,4);

[z_17,~] = audioread('toot_clip_17.wav');
z_17_1 = z_17(1:220500,1);
toot_17 = downsample(z_17_1,4);

[z_18,~] = audioread('toot_clip_18.wav');
z_18_1 = z_18(1:220500,1);
toot_18 = downsample(z_18_1,4);

[z_19,~] = audioread('toot_clip_19.wav');
z_19_1 = z_19(1:220500,1);
toot_19 = downsample(z_19_1,4);

[z_20,Fs] = audioread('toot_clip_20.wav');
z_20_1 = z_20(1:220500,1);
toot_20 = downsample(z_20_1,4);

toot =
[toot_1,toot_2,toot_3,toot_4,toot_5,toot_6,toot_7,toot_8,toot_9,
toot_10,...

toot_11,toot_12,toot_13,toot_14,toot_15,toot_16,toot_17,toot_18,
toot_19,toot_20];

%%
feature = 45;
bobdata = band_spec(bob,Fs);
jimdata = band_spec(jim,Fs);
tootdata = band_spec(toot,Fs);

[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(bobdata,jimdata,tootdata,feature);


%% Projection onto W
figure(4)
plot(sortband1,zeros(20),'ob','Linewidth',2,'HandleVisibility','
off')
yline(0,'b--','DisplayName','Bob Marley');
xline(threshold_1,'DisplayName','threshold 1');
hold on
plot(sortband2,ones(20),'dr','Linewidth',2,'HandleVisibility','o
ff')
yline(1,'r--','DisplayName','Jimmy Cliff');
xline(threshold_2,'DisplayName','threshold 2');
```

```
hold on
plot(sortband3,ones(20)*-
1,'gd','Linewidth',2,'HandleVisibility','off')
yline(-1,'g--','DisplayName','Toots and the Maytals');
ylim([-2 2]);
legend('Location','northwest');
title('Projections onto W');
%% Histogram
figure(5)
subplot(1,3,1)
histogram(sortband1,15);
title('Bob');
subplot(1,3,2)
histogram(sortband2,15);
title('Jimmy C');
subplot(1,3,3)
histogram(sortband3,15);
title('Toots');

%% Load test data
[t_1,~] = audioread('bob_clip_21.wav');
t_1_1 = t_1(1:220500,1);
test_1 = downsample(t_1_1,4);

[t_2,~] = audioread('bob_clip_22.wav');
t_2_1 = t_2(1:220500,1);
test_2 = downsample(t_2_1,4);

[t_3,~] = audioread('bob_clip_23.wav');
t_3_1 = t_3(1:220500,1);
test_3 = downsample(t_3_1,4);

[t_4,~] = audioread('jim_clip_21.wav');
t_4_1 = t_4(1:220500,1);
test_4 = downsample(t_4_1,4);

[t_5,~] = audioread('jim_clip_22.wav');
t_5_1 = t_5(1:220500,1);
test_5 = downsample(t_5_1,4);

[t_6,~] = audioread('jim_clip_23.wav');
t_6_1 = t_6(1:220500,1);
test_6 = downsample(t_6_1,4);

[t_7,~] = audioread('toot_clip_21.wav');
t_7_1 = t_7(1:220500,1);
test_7 = downsample(t_7_1,4);
```

```
[t_8,~] = audioread('toot_clip_22.wav');
t_8_1 = t_8(1:220500,1);
test_8 = downsample(t_8_1,4);

[t_9,Fs] = audioread('toot_clip_23.wav');
t_9_1 = t_9(1:220500,1);
test_9 = downsample(t_9_1,4);

test =
[test_1,test_2,test_3,test_4,test_5,test_6,test_7,test_8,test_9]
;

testData = band_spec(test,Fs);
testpval = zeros(1,9);
for k = 1:9
 testPCA = U'*testData(:,k);
 testpval(:,k) = w'*testPCA(1:feature,:);
end


for k = 1:9
     if testpval(:,k) < threshold_1
         disp('Bob Marley')
     elseif threshold_1 < testpval(:,k) < threshold_2
         disp('Jimmy Cliff')
     elseif testpval(:,k) > threshold_2
         disp('Toots and the Maytals')
     end
end

%result: 4/9 correct - 44.4% success
%%
function bandData = band_spec(bandfile,Fs)

    [~,n] = size(bandfile);
    [Spc] = spectrogram(bandfile(:,1),302,150,302,'yaxis',
Fs/4);
    [~,y] = size(Spc);
    bandData = zeros(60*y,n);

    for k = 1:n
       [Sp, ~, ~] =
spectrogram(bandfile(:,k),302,150,302,'yaxis', Fs/4);
       Gr = abs(Sp(1:60,:));
       [l,w] = size(Gr);
       bandData(:,k) = reshape(Gr,l*w,1);
```

```
    end

end

function
[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(band1_S,band2_S,band3_S,feature)

    [U,S,V] = svd([band1_S band2_S band3_S], 'econ');
    nband1 = size(band1_S,2); nband2 = size(band2_S,2); nband3 =
size(band3_S,2);
    bands = S*V';
    band1 = bands(1:feature,1:nband1);
    band2 = bands(1:feature,nband1+1:nband1+nband2);
    band3 =
bands(1:feature,nband1+nband2+1:nband1+nband2+nband3);

    mband1 = mean(band1,2);
    mband2 = mean(band2,2);
    mband3 = mean(band3,2);

    Sw = 0; % within class variances
    for k=1:nband1
        Sw = Sw + (band1(:,k)-mband1)*(band1(:,k)-mband1)';
    end
    for k=1:nband2
        Sw = Sw + (band2(:,k)-mband2)*(band2(:,k)-mband2)';
    end
    for k=1:nband3
        Sw = Sw + (band3(:,k)-mband3)*(band3(:,k)-mband3)';
    end

    mbands = mean(bands(1:feature,:),2);
    means = [mband1,mband2,mband3];
    Sb = 0; %between class variances
    for j=1:3
        Sb = 20*(means(:,j)-mbands)*(means(:,j)-mbands)';
    end

    [V2,D] = eig(Sb,Sw); % linear discriminant analysis
    [~,ind] = max(abs(diag(D)));
    w = V2(:,ind); w = w/norm(w,2);

    vband1 = w'*band1;
    vband2 = w'*band2;
    vband3 = w'*band3;
```

```
    sortband1 = sort(vband1);
    sortband2 = sort(vband2);
    sortband3 = sort(vband3);

    msortband1 = mean(sortband1);
    msortband2 = mean(sortband2);
    msortband3 = mean(sortband3);

    msorts = [msortband1,msortband2,msortband3];
    msort = sort(msorts);
    threshold_1 = (msort(1)+msort(2))/2;
    threshold_2 = (msort(2)+msort(3))/2;

end



%HW4 - AMATH 482-Test 2
%Machine Learning
clear all;close all;clc;


%load and sort training data
%Reggae
[y_1,~] = audioread('bob_clip_1.wav');
y_1_1 = y_1(1:220500,1);
reg_1 = downsample(y_1_1,4);

[y_2,~] = audioread('bob_clip_2.wav');
y_2_1 = y_2(1:220500,1);
reg_2 = downsample(y_2_1,4);

[y_3,~] = audioread('bob_clip_3.wav');
y_3_1 = y_3(1:220500,1);
reg_3 = downsample(y_3_1,4);

[y_4,~] = audioread('bob_clip_4.wav');
y_4_1 = y_4(1:220500,1);
reg_4 = downsample(y_4_1,4);

[y_5,~] = audioread('bob_clip_5.wav');
y_5_1 = y_5(1:220500,1);
reg_5 = downsample(y_5_1,4);

[y_6,~] = audioread('bob_clip_6.wav');
y_6_1 = y_6(1:220500,1);
```

```
reg_6 = downsample(y_6_1,4);

[y_7,~] = audioread('jim_clip_7.wav');
y_7_1 = y_7(1:220500,1);
reg_7 = downsample(y_7_1,4);

[y_8,~] = audioread('jim_clip_8.wav');
y_8_1 = y_8(1:220500,1);
reg_8 = downsample(y_8_1,4);

[y_9,~] = audioread('jim_clip_9.wav');
y_9_1 = y_9(1:220500,1);
reg_9 = downsample(y_9_1,4);

[y_10,~] = audioread('jim_clip_10.wav');
y_10_1 = y_10(1:220500,1);
reg_10 = downsample(y_10_1,4);

[y_11,~] = audioread('jim_clip_11.wav');
y_11_1 = y_11(1:220500,1);
reg_11 = downsample(y_11_1,4);

[y_12,~] = audioread('toot_clip_12.wav');
y_12_1 = y_12(1:220500,1);
reg_12 = downsample(y_12_1,4);

[y_13,~] = audioread('jim_clip_13.wav');
y_13_1 = y_13(1:220500,1);
reg_13 = downsample(y_13_1,4);

[y_14,~] = audioread('jim_clip_14.wav');
y_14_1 = y_14(1:220500,1);
reg_14 = downsample(y_14_1,4);

[y_15,~] = audioread('jim_clip_15.wav');
y_15_1 = y_15(1:220500,1);
reg_15 = downsample(y_15_1,4);

[y_16,~] = audioread('toot_clip_16.wav');
y_16_1 = y_16(1:220500,1);
reg_16 = downsample(y_16_1,4);

[y_17,~] = audioread('toot_clip_17.wav');
y_17_1 = y_17(1:220500,1);
reg_17 = downsample(y_17_1,4);

[y_18,~] = audioread('toot_clip_18.wav');
```

```
y_18_1 = y_18(1:220500,1);
reg_18 = downsample(y_18_1,4);

[y_19,~] = audioread('toot_clip_19.wav');
y_19_1 = y_19(1:220500,1);
reg_19 = downsample(y_19_1,4);

[y_20,~] = audioread('toot_clip_20.wav');
y_20_1 = y_20(1:220500,1);
reg_20 = downsample(y_20_1,4);

reg =
[reg_1,reg_2,reg_3,reg_4,reg_5,reg_6,reg_7,reg_8,reg_9,reg_10,..
.

reg_11,reg_12,reg_13,reg_14,reg_15,reg_16,reg_17,reg_18,reg_19,r
eg_20];

%Classical
[x_1,~] = audioread('class_clip_1.wav');
x_1_1 = x_1(1:220500,1);
class_1 = downsample(x_1_1,4);

[x_2,~] = audioread('class_clip_2.wav');
x_2_1 = x_2(1:220500,1);
class_2 = downsample(x_2_1,4);

[x_3,~] = audioread('class_clip_3.wav');
x_3_1 = x_3(1:220500,1);
class_3 = downsample(x_3_1,4);

[x_4,~] = audioread('class_clip_4.wav');
x_4_1 = x_4(1:220500,1);
class_4 = downsample(x_4_1,4);

[x_5,~] = audioread('class_clip_5.wav');
x_5_1 = x_5(1:220500,1);
class_5 = downsample(x_5_1,4);

[x_6,~] = audioread('class_clip_6.wav');
x_6_1 = x_6(1:220500,1);
class_6 = downsample(x_6_1,4);

[x_7,~] = audioread('class_clip_7.wav');
x_7_1 = x_7(1:220500,1);
class_7 = downsample(x_7_1,4);
```

```
[x_8,~] = audioread('class_clip_8.wav');
x_8_1 = x_8(1:220500,1);
class_8 = downsample(x_8_1,4);

[x_9,~] = audioread('class_clip_9.wav');
x_9_1 = x_9(1:220500,1);
class_9 = downsample(x_9_1,4);

[x_10,~] = audioread('class_clip_10.wav');
x_10_1 = x_10(1:220500,1);
class_10 = downsample(x_10_1,4);

[x_11,~] = audioread('class_clip_11.wav');
x_11_1 = x_11(1:220500,1);
class_11 = downsample(x_11_1,4);

[x_12,~] = audioread('class_clip_12.wav');
x_12_1 = x_12(1:220500,1);
class_12 = downsample(x_12_1,4);

[x_13,~] = audioread('class_clip_13.wav');
x_13_1 = x_13(1:220500,1);
class_13 = downsample(x_13_1,4);

[x_14,~] = audioread('class_clip_14.wav');
x_14_1 = x_14(1:220500,1);
class_14 = downsample(x_14_1,4);

[x_15,~] = audioread('class_clip_15.wav');
x_15_1 = x_15(1:220500,1);
class_15 = downsample(x_15_1,4);

[x_16,~] = audioread('class_clip_16.wav');
x_16_1 = x_16(1:220500,1);
class_16 = downsample(x_16_1,4);

[x_17,~] = audioread('class_clip_17.wav');
x_17_1 = x_17(1:220500,1);
class_17 = downsample(x_17_1,4);

[x_18,~] = audioread('class_clip_18.wav');
x_18_1 = x_18(1:220500,1);
class_18 = downsample(x_18_1,4);

[x_19,~] = audioread('class_clip_19.wav');
x_19_1 = x_19(1:220500,1);
class_19 = downsample(x_19_1,4);
```

```
[x_20,~] = audioread('class_clip_20.wav');
x_20_1 = x_20(1:220500,1);
class_20 = downsample(x_20_1,4);

class =
[class_1,class_2,class_3,class_4,class_5,class_6,class_7,class_8
,class_9,class_10,...

class_11,class_12,class_13,class_14,class_15,class_16,class_17,c
lass_18,class_19,class_20];

%Metal
[z_1,~] = audioread('metal_clip_1.wav');
z_1_1 = z_1(1:220500,1);
metal_1 = downsample(z_1_1,4);

[z_2,~] = audioread('metal_clip_2.wav');
z_2_1 = z_2(1:220500,1);
metal_2 = downsample(z_2_1,4);

[z_3,~] = audioread('metal_clip_3.wav');
z_3_1 = z_3(1:220500,1);
metal_3 = downsample(z_3_1,4);

[z_4,~] = audioread('metal_clip_4.wav');
z_4_1 = z_4(1:220500,1);
metal_4 = downsample(z_4_1,4);

[z_5,~] = audioread('metal_clip_5.wav');
z_5_1 = z_5(1:220500,1);
metal_5 = downsample(z_5_1,4);

[z_6,~] = audioread('metal_clip_6.wav');
z_6_1 = z_6(1:220500,1);
metal_6 = downsample(z_6_1,4);

[z_7,~] = audioread('metal_clip_7.wav');
z_7_1 = z_7(1:220500,1);
metal_7 = downsample(z_7_1,4);

[z_8,~] = audioread('metal_clip_8.wav');
z_8_1 = z_8(1:220500,1);
metal_8 = downsample(z_8_1,4);

[z_9,~] = audioread('metal_clip_9.wav');
z_9_1 = z_9(1:220500,1);
```

```
metal_9 = downsample(z_9_1,4);

[z_10,~] = audioread('metal_clip_10.wav');
z_10_1 = z_10(1:220500,1);
metal_10 = downsample(z_10_1,4);

[z_11,~] = audioread('metal_clip_11.wav');
z_11_1 = z_11(1:220500,1);
metal_11 = downsample(z_11_1,4);

[z_12,~] = audioread('metal_clip_12.wav');
z_12_1 = z_12(1:220500,1);
metal_12 = downsample(z_12_1,4);

[z_13,~] = audioread('metal_clip_13.wav');
z_13_1 = z_13(1:220500,1);
metal_13 = downsample(z_13_1,4);

[z_14,~] = audioread('metal_clip_14.wav');
z_14_1 = z_14(1:220500,1);
metal_14 = downsample(z_14_1,4);

[z_15,~] = audioread('metal_clip_15.wav');
z_15_1 = z_15(1:220500,1);
metal_15 = downsample(z_15_1,4);

[z_16,~] = audioread('metal_clip_16.wav');
z_16_1 = z_16(1:220500,1);
metal_16 = downsample(z_16_1,4);

[z_17,~] = audioread('metal_clip_17.wav');
z_17_1 = z_17(1:220500,1);
metal_17 = downsample(z_17_1,4);

[z_18,~] = audioread('metal_clip_18.wav');
z_18_1 = z_18(1:220500,1);
metal_18 = downsample(z_18_1,4);

[z_19,~] = audioread('metal_clip_19.wav');
z_19_1 = z_19(1:220500,1);
metal_19 = downsample(z_19_1,4);

[z_20,Fs] = audioread('metal_clip_20.wav');
z_20_1 = z_20(1:220500,1);
metal_20 = downsample(z_20_1,4);
```

```
metal =
[metal_1,metal_2,metal_3,metal_4,metal_5,metal_6,metal_7,metal_8
,metal_9,metal_10,...

metal_11,metal_12,metal_13,metal_14,metal_15,metal_16,metal_17,m
etal_18,metal_19,metal_20];

%%
feature = 45;
regdata = band_spec(reg,Fs);
classdata = band_spec(class,Fs);
metaldata = band_spec(metal,Fs);

[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(regdata,classdata,metaldata,feature);

%% Projection onto W
figure(4)
plot(sortband1,zeros(20),'ob','Linewidth',2,'HandleVisibility','
off')
yline(0,'b--','DisplayName','Reggae');
xline(threshold_1,'DisplayName','threshold 1');
hold on
plot(sortband2,ones(20),'dr','Linewidth',2,'HandleVisibility','o
ff')
yline(1,'r--','DisplayName','Classical');
xline(threshold_2,'DisplayName','threshold 2');
hold on
plot(sortband3,ones(20)*-
1,'gd','Linewidth',2,'HandleVisibility','off')
yline(-1,'g--','DisplayName','Metal');
ylim([-2 2]);
legend('Location','northwest');
title('Projections onto W');
%% Histogram
figure(5)
subplot(1,3,1)
histogram(sortband1,15);
title('Reggae');
subplot(1,3,2)
histogram(sortband2,15);
title('Classical');
subplot(1,3,3)
histogram(sortband3,15);
title('Metal');
%% load test data
```

```
[t_1,~] = audioread('bob_clip_21.wav');
t_1_1 = t_1(1:220500,1);
test_1 = downsample(t_1_1,4);

[t_2,~] = audioread('jim_clip_22.wav');
t_2_1 = t_2(1:220500,1);
test_2 = downsample(t_2_1,4);

[t_3,~] = audioread('toot_clip_23.wav');
t_3_1 = t_3(1:220500,1);
test_3 = downsample(t_3_1,4);

[t_4,~] = audioread('class_clip_21.wav');
t_4_1 = t_4(1:220500,1);
test_4 = downsample(t_4_1,4);

[t_5,~] = audioread('class_clip_22.wav');
t_5_1 = t_5(1:220500,1);
test_5 = downsample(t_5_1,4);

[t_6,~] = audioread('class_clip_23.wav');
t_6_1 = t_6(1:220500,1);
test_6 = downsample(t_6_1,4);

[t_7,~] = audioread('metal_clip_21.wav');
t_7_1 = t_7(1:220500,1);
test_7 = downsample(t_7_1,4);

[t_8,~] = audioread('metal_clip_22.wav');
t_8_1 = t_8(1:220500,1);
test_8 = downsample(t_8_1,4);

[t_9,Fs] = audioread('metal_clip_23.wav');
t_9_1 = t_9(1:220500,1);
test_9 = downsample(t_9_1,4);

test =
[test_1,test_2,test_3,test_4,test_5,test_6,test_7,test_8,test_9]
;
testData = band_spec(test,Fs);
testpval = zeros(1,9);
for k = 1:9
 testPCA = U'*testData(:,k);
 testpval(:,k) = w'*testPCA(1:feature,:);
end
```

```
for k = 1:9
     if testpval(:,k) < threshold_1
         disp('Metal')
     elseif threshold_1 < testpval(:,k) < threshold_2
         disp('Reggae')
     elseif testpval(:,k) > threshold_2
         disp('Classical')
     end
end

%results: 6/9 correct 66% success rate
%%
function bandData = band_spec(bandfile,Fs)

    [~,n] = size(bandfile);
    [Spc] = spectrogram(bandfile(:,1),302,150,302,'yaxis',
Fs/4);
    [~,y] = size(Spc);
    bandData = zeros(60*y,n);

    for k = 1:n
       [Sp, ~, ~] =
spectrogram(bandfile(:,k),302,150,302,'yaxis', Fs/4);
       Gr = abs(Sp(1:60,:));
       [l,w] = size(Gr);
       bandData(:,k) = reshape(Gr,l*w,1);
    end

end

function
[U,S,V,threshold_1,threshold_2,w,sortband1,sortband2,sortband3]
= ...
    band_trainer(band1_S,band2_S,band3_S,feature)

    [U,S,V] = svd([band1_S band2_S band3_S], 'econ');
    nband1 = size(band1_S,2); nband2 = size(band2_S,2); nband3 =
size(band3_S,2);
    bands = S*V';
    band1 = bands(1:feature,1:nband1);
    band2 = bands(1:feature,nband1+1:nband1+nband2);
    band3 =
bands(1:feature,nband1+nband2+1:nband1+nband2+nband3);

    mband1 = mean(band1,2);
    mband2 = mean(band2,2);
    mband3 = mean(band3,2);
```

```
    Sw = 0; % within class variances
    for k=1:nband1
        Sw = Sw + (band1(:,k)-mband1)*(band1(:,k)-mband1)';
    end
    for k=1:nband2
        Sw = Sw + (band2(:,k)-mband2)*(band2(:,k)-mband2)';
    end
    for k=1:nband3
        Sw = Sw + (band3(:,k)-mband3)*(band3(:,k)-mband3)';
    end

    mbands = mean(bands(1:feature,:),2);
    means = [mband1,mband2,mband3];
    Sb = 0; %between class variances
    for j=1:3
        Sb = 20*(means(:,j)-mbands)*(means(:,j)-mbands)';
    end

    [V2,D] = eig(Sb,Sw); % linear discriminant analysis
    [~,ind] = max(abs(diag(D)));
    w = V2(:,ind); w = w/norm(w,2);

    vband1 = w'*band1;
    vband2 = w'*band2;
    vband3 = w'*band3;

    sortband1 = sort(vband1);
    sortband2 = sort(vband2);
    sortband3 = sort(vband3);

    msortband1 = mean(sortband1);
    msortband2 = mean(sortband2);
    msortband3 = mean(sortband3);

    msorts = [msortband1,msortband2,msortband3];
    msort = sort(msorts);
    threshold_1 = (msort(1)+msort(2))/2;
    threshold_2 = (msort(2)+msort(3))/2;

end
```