

## Contents

- [start rearrange data and fit](#)
- [modeling in svd data and non svd data](#)
- [plot](#)

```
clear all; close all; clc;
% Test 1

% Load music data, sort and reduce sampling rate.
% Section out 5 seconds music data from each artist's song.
% Artists : "Dr.dre", "Beethoven", "The Beatles" ;
% 5 songs for each artist.

tic
% Read folder and file name.
flist = dir("music");
fname = {flist.name};
fname = fname(~strncmp(fname, '.', 1));

% Data aquare, format and reduce for each song.
musicAll = [];
labels = [];
Aname = ["Dr.Dre", "Beethoven", "The Beatles"];

for h = 1:length(Aname)
    localAname = Aname(h);
    % using artist name to sort song
    tf = startsWith(fname, localAname, 'IgnoreCase', true);
    target = fname(tf);
    for i = 1:length(target)
        % load song
        localsname = string(target(i));
        [music, Fs] = audioread(strcat("music\", localsname));
        % lower sampling rate 1/10
        Fs = Fs/10;
        music = music(1:10:end, :);
        % convert to mono tone; skip if already mono tone.
        musicSize = size(music);
        musicSize = musicSize(2);
        if musicSize == 1
            Mmono = music;
        else
            Mmono = zeros(length(music), 1);
            for j = 1:length(music)
                if (music(j, 1) == 0 || music(j, 2) == 0)
                    Mmono(j, 1) = max(music(j, :));
                else
                    Mmono(j, 1) = (music(j, 1) + music(j, 2)) / 2;
                end
            end
        end
        % remove zeros on head and tail
        Mhead = find(Mmono, 1, "first");
        Mtail = find(Mmono, 1, "last");
        Mmono = Mmono(Mhead:Mtail);
        % aquare 5 seconds clips from music; every 5sec one clip
        % round music to n * 5 seconds
        Intervals = Fs*5;
        sections = floor(length(Mmono) / (Fs*5));
        Mmono = Mmono(1:(Intervals * sections), 1);
        data5s = reshape(Mmono, [Intervals, sections]);
        musicAll = [musicAll, data5s];
        labels = [labels; repmat(localAname, size(data5s, 2), 1)];
    end
end
labels = labels.';
trainlabel = labels;

toc
```

历时 11.094174 秒。

## start rearrange data and fit

```
tic
```

```

% svd data
[u,s,v] = svd(abs(fft(musicAll)), "econ");
sig = diag(s);
lambda = sig.^2;
%plot ( cumsum ( lambda /sum ( lambda )), 'bo ')
utrunc = u(:, 1:226);
traindata = utrunc'*musicAll;

%building test data set by selection rate of 1/10 from traindata set,
%randomly selected.

sizen = floor(size(traindata,2)/(10));
testnumber = randperm (size(traindata,2), sizen);

testdata = traindata(:,testnumber);
testlabel = trainlabel(:,testnumber);

%remove test data from train data set.
traindata(:,testnumber) = [];
trainlabel(:,testnumber) = [];

%non svd train and test data sets.
nonSvdtraindata = abs(fft(musicAll));
nonSvdtestdata = nonSvdtraindata(:,testnumber);
%remove test data from train data set.
nonSvdtraindata(:,testnumber) = [];

toc

```

历时 2.942937 秒。

## modeling in svd data and non svd data

```

tic
% Naive Bayes classifier modeling(svd data)
NBCModel = fitcnb(traindata.', trainlabel);
NBCLoss = loss(NBCModel , testdata.', testlabel);

svdDataTimeElapsed = toc;

tic
% Naive Bayes classifier modeling(non svd data)
nonSvdNBCModel = fitcnb(nonSvdtraindata.', trainlabel);
nonSvdNBCLoss = loss(nonSvdNBCModel , nonSvdtestdata.', testlabel);

nonSvdDataTimeElapsed = toc;

```

```

%LDA classifier modeling(svd data)
tic
LDAttrainData = traindata.';
LDAttrainlabel = trainlabel.';
MdlLinear = fitcdiscr(LDAttrainData,LDAttrainlabel);
Testclass = predict(MdlLinear,testdata.');
ars = find(Testclass == testlabel.');
LDA_SVDrate = length(ars)/length(testlabel);

svdLDATimeElapsed = toc;

%LDA classifier modeling(non svd data)
tic
LDAttrainData = nonSvdtraindata.';
LDAttrainlabel = trainlabel.';
MdlLinear = fitcdiscr(LDAttrainData,LDAttrainlabel);
Testclass = predict(MdlLinear,nonSvdtestdata.');
ars = find(Testclass == testlabel.');
LDA_nonSVDrate = length(ars)/length(testlabel);

nonSvdLDATimeElapsed = toc;

```

## plot

```

subplot(1,2,1)
b = bar(1-[NBCLoss,nonSvdNBCLoss]);
xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(b(1).YData);

```

```

ylim ([0, 1]) ;
text(xtips1,ytips1,labels1,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom')
xticklabels (["SVD-Data","Non SVD-data"]);
ylabel (" Classification Accuracy ( Scale of 0 to 1) ");
xlabel (" Data Type ");
title (" Accuracy of Naive Bayes classifier Model of Three artists in Different Genres");
grid on
subplot(1,2,2)
b = bar([svdDataTimeElapsed;nonSvdDataTimeElapsed]);
xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(b(1).YData);
text(xtips1,ytips1,labels1,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom')
xticklabels (["SVD-Data time","Non SVD-data time"]);
ylabel (" Time(seconds) ");
xlabel (" Data Type ");
title (" Processing Time of Naive Bayes classifier Model of Three artists in Different Genres");

figure(2)
subplot(1,2,1)
b = bar([LDA_SVDrate;LDA_nonSVDrate]);
xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(b(1).YData);
ylim ([0, 1]) ;
text(xtips1,ytips1,labels1,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom')
xticklabels (["SVD-Data","Non SVD-data"]);
ylabel (" Classification Accuracy ( Scale of 0 to 1) ");
xlabel (" Data Type ");
title (" Accuracy of Linear discriminant analysis Model of Three artists in Different Genres");
grid on
subplot(1,2,2)
b = bar([svdLDATimeElapsed;nonSvdLDATimeElapsed]);
xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(b(1).YData);
text(xtips1,ytips1,labels1,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom')
xticklabels (["SVD-Data time","Non SVD-data time"]);
ylabel (" Time(seconds) ");
xlabel (" Data Type ");
title (" Processing Time of Linear discriminant analysis Model of Three artists in Different Genres");

```