

DSPLAB 專題(二) 以 KNN 實現病理嗓音偵測實作

系級：電機甲 3A 學號：1070557 姓名：古心好

Q1： 嘗試調整 mfcc_v2.m 中的參數 sample_rate、frame_time、frame_move_time，說明

mfcc 檔對於聲音擷取特徵的數值變化和意義，調整參數說明對變化和意義。(20%)

參數說明：

lsample_rate：一段音訊的取樣頻率 (單位：HZ)

lframe_time：一個音框的取樣時間長度 (單位：毫秒)

lframe_move_time：一個音框與取下一個音框重疊的時間 (單位：毫秒)

```
% 呼叫 mfcc_v2.m 改 sample_rate、frame_time、frame_move_time
% 初始：sample_rate=44100, frame_time=30, frame_move_time=15
% 只更改 sample_rate=16000(取樣頻率變小)
% 只更改 sample_rate=80000(取樣頻率變大)
% 只更改 frame_time=100, frame_move_time=50
% 只更改 frame_time=10, frame_move_time=5
```

▲ 問題一的程式碼解說

- 初始：sample_rate=44100, frame_time=30, frame_move_time=15

```
complete
complete
complete
Elapsed time is 139.747100 seconds.
```

- 只更改 sample_rate=16000(取樣頻率變小)

```
complete
complete
complete
Elapsed time is 269.128674 seconds
```

- 只更改 sample_rate=80000(取樣頻率變大)

```
complete
complete
complete
Elapsed time is 90.774188 seconds.
```

- 只更改 frame_time=100, frame_move_time=50

```
complete
complete
complete
Elapsed time is 59.004782 seconds.
```

- 只更改 frame_time=10, frame_move_time=5

```
complete
complete
complete
Elapsed time is 308.204840 seconds.
```

根據觀察，會發現 sample_rate(取樣頻率)變小時，取的樣本較多、樣本間間隔較小，因此耗時會較久，若將取樣頻率調大，取的樣本較少、樣本間間隔較大，較不耗時。若將 frame_time、frame_move_time 調大，一個音框的取樣時間長度較長，雖然重疊的多，但是因為一個音框包含的時間較長，整體的完成速度依然很快。但若將 frame_time、frame_move_time 調小，在上面我的 code 中，frame_time=10、frame_move_time=5，等於我一個音框一半會跟前面重疊，另一半會跟後面重疊，跑的時間大幅提升。

Q2： 完成下表，嘗試調整 K 值設為 3,5,7 奇數，調整 K 為多少時準確值較佳？(20%)

```
path = 'E:/2020 DSP專題二/Train_Test/normal_train';% 輸入mfc檔路徑資料夾
cd(path) % 進入選取的資料夾
file=dir([path '*.*mfc']);
Data= [];
for i = 1:length(file)
    cep_data = load(file(i).name); % 讀入每個mfc檔
    number_person(i,:) = length(cep_data(:,1)); % 每個mfc檔的長度
    Data= [Data;cep_data]; % 與上個mfc檔進行串聯
end
lable=zeros(length(Data),1)
Data_normal_train=[lable Data]

path = 'E:/2020 DSP專題二/Train_Test/im_train';% 輸入mfc檔路徑資料夾
cd(path) % 進入選取的資料夾
file=dir([path '*.*mfc']);
Data= [];
for i = 1:length(file)
    cep_data = load(file(i).name); % 讀入每個mfc檔
    number_person(i,:) = length(cep_data(:,1)); % 每個mfc檔的長度
    Data= [Data;cep_data]; % 與上個mfc檔進行串聯
end
lable=ones(length(Data),1)
Data_im_train=[lable Data]

Data_train=[Data_normal_train;Data_im_train]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

▲ 問題二的程式碼 - labeling

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
path = 'E:/2020 DSP專題二/Train_Test/normal_test';% 輸入mfc檔路徑資料夾
cd(path) % 進入選取的資料夾
file=dir([path '*.mfc']);
Data= [];
for i = 1:length(file)
    cep_data = load(file(i).name); % 讀入每個mfc檔
    number_person_normal(i,:) = length(cep_data(:,1)); % 每個mfc檔的長度
    Data= [Data;cep_data]; % 與上個mfc檔進行串聯
end
lable=zeros(length(Data),1)
Data_normal_test=[lable Data]

path = 'E:/2020 DSP專題二/Train_Test/im_test';% 輸入im_CYST_test檔路徑資料夾
cd(path) % 進入選取的資料夾
file=dir([path '*.mfc']);
Data= [];
for i = 1:length(file)
    cep_data = load(file(i).name); % 讀入每個mfc檔
    number_person_im(i,:) = length(cep_data(:,1)); % 每個mfc檔的長度
    Data= [Data;cep_data]; % 與上個mfc檔進行串聯
end
lable=ones(length(Data),1)
Data_im_test=[lable Data]

Data_test=[Data_normal_test;Data_im_test]
Person_test=[number_person_normal;number_person_im]

```

▲ 問題二的程式碼 - labeling

```

% Frame
tested = trainedModel.predictFcn(Data_test(:,2:27)); % k=1;
True_frame=0;

for i=1:21785 % 總音框數
    if Data_test(i,1)==tested(i,1); % 檢查預測是否正確
        True_frame=True_frame+1; % 若正確, True_frame+1
    end
end
Frame=(True_frame/21785)*100; % 計算百分比
fprintf('k=1: %2.2f %%', Frame); % 顯示百分比

% Person
last_frame=0; % 初始化
Correct=0; % 初始化
True_person(1:10,1)=0, True_person(11:20,1)=1; % 正解
for person=1:20 % 總共20人
    count=last_frame+1; % count從上一輪結束開始算
    last_frame=last_frame+Person_test(person,1); % 得知計算到何處為一個人的

    if sum(tested(count:last_frame,1))>Person_test(person)/2 % 若有病音框>一半
        predict(person,1)=1; % 若大於一半, predict存1
    else
        predict(person,1)=0; % 若小於一半, predict存0
    end

    if True_person(person,1)==predict(person,1) % 比較預測人跟實際人是否相同
        Correct=Correct+1; % 若相同, Correct+1
    end
end
Correct=(Correct/20)*100; % 計算百分比
fprintf('k=1: %d %%', Correct); % 顯示百分比

```

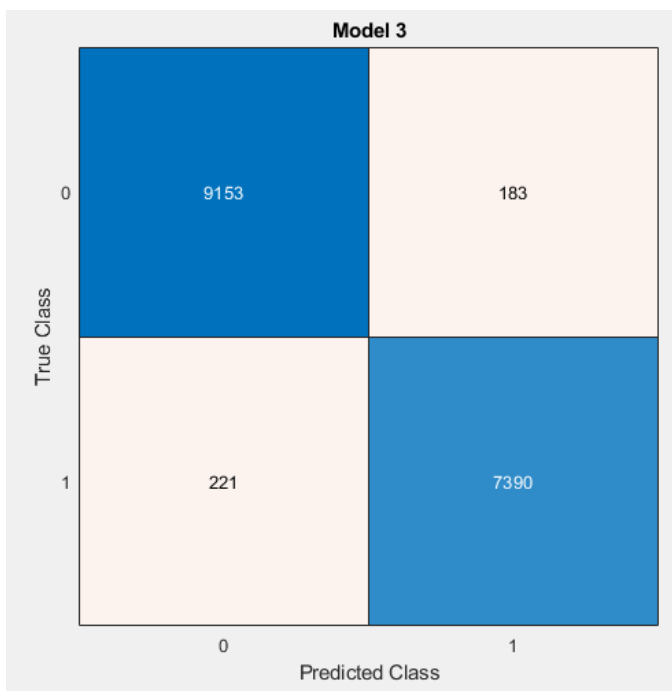
▲ 問題二的程式碼 - Accuracy



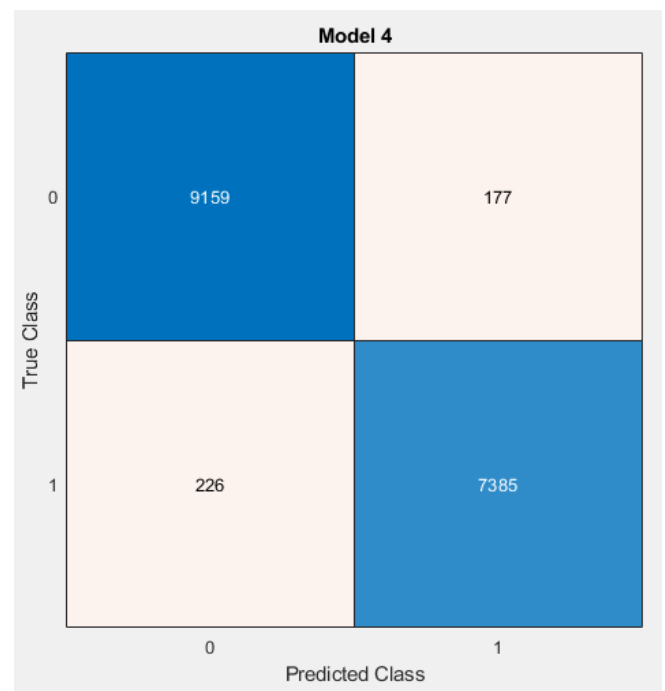
▲ K=1 時



▲ K=3 時



▲ K=5 時



▲ K=7 時

K	1	3	5	7
Accuracy (frame)	78.68%	79.18%	79.50%	79.84%
Accuracy (person)	80.00%	80.00%	80.00%	80.00%

▲ K 值對 Data_test 的 Accuracy 的影響

K 值代表，該筆資料的分類為何，是由該資料點的位置附近的 K 筆資料去決定，若它的資料附近正大於負，則該筆資料屬於正。根據觀察，可能因為資料量不大，因此 Accuracy of person 的差異性並不大，皆是等於 80%，但是在數萬多個音框中，可以觀測到準確率或多或少有受到影響，但是為正向的成長。

但是個人認為準確率固然重要，但是在醫學其實靈敏度會更重要，為了防止有病的人沒被檢測出，因此在混淆矩陣上會採用 $TP/(TP+FN)$ ，若數據更多預測應該可以更加準確。(表格為 Data_train 的 Sensitivity)

K	1	3	5	7
Sensitivity(frame)	97.75%	97.47%	97.09%	97.03%

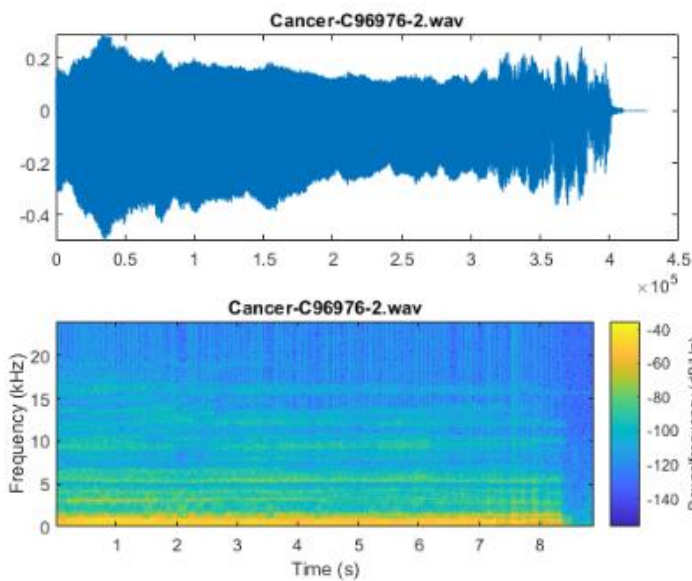
▲ K 值對 Data_train 的 Sensitivity 的影響

Q3： 嘗試畫出生病與正常聲音（各挑選一個）的頻譜圖和波形圖（參考 LAB6），說明之間的差異？(20%)

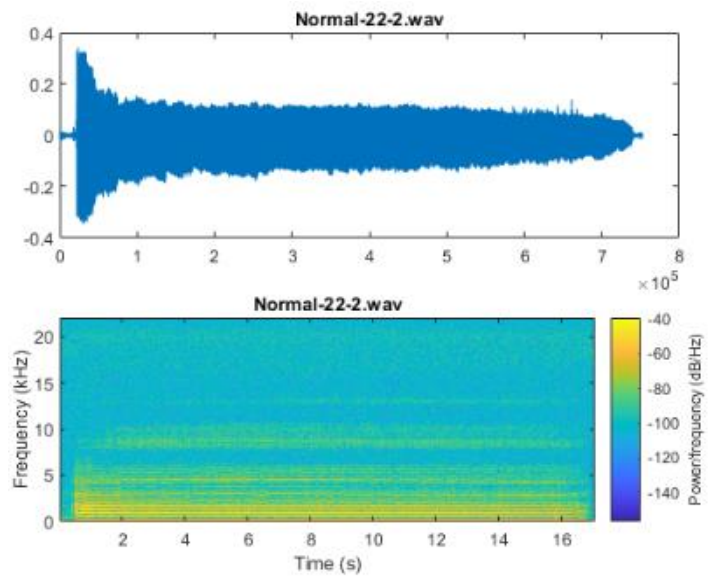
```
% Cancer
[y,fs]=audioread('C96976-2.wav');
figure;
subplot(2,1,1);
plot(y);
title('Cancer-C96976-2.wav');
subplot(2,1,2);
spectrogram(y(:,1), 1024, 1000, [], fs, 'yaxis')
title('Cancer-C96976-2.wav');

% Normal
[y,fs]=audioread('22-2.wav');
figure;
subplot(2,1,1);
plot(y);
title('Normal-22-2.wav');
subplot(2,1,2);
spectrogram(y(:,1), 1024, 1000, [], fs, 'yaxis')
title('Normal-22-2.wav');
```

▲ 問題三的程式碼



▲ 生病聲音的波型圖和頻譜圖



▲ 正常聲音的波型圖和頻譜圖

● 根據波型圖比較：

我們會發現生病聲音的波型圖較混雜，低頻聲音部分居多，正常聲音則是從頭到尾波型都很一致，沒甚麼變化，也不會有那麼多低頻聲音。

● 根據頻譜圖比較：

會更加明顯，生病聲音的頻譜圖顏色很明顯地和正常聲音完全不同，生病聲音幾乎都會是低頻的深藍色，再參雜一些綠色，看起來亂亂的。而正常聲音則是很均勻的天藍色，就算參雜綠色也是很均勻的參雜，看起來不會很混亂。

● 用聽的：

生病聲音會有很明顯的沙啞聲。

Q4： 使用實驗三、四預測在資料夾 other 中的 10 個語音檔(20%)

```

path = 'E:/2020 DSP專題二/MFCC/other';           % 輸入other的mfc檔路徑資料夾
cd(path)                                           % 進入選取的資料夾
file=dir([path '*.*mfc']);
Data= [];
for i = 1:length(file)
    cep_data = load(file(i).name);                % 讀入每個mfc檔
    Number_other(i,:) = length(cep_data(:,1));    % 每個mfc檔的長度
    Data= [Data;cep_data];                        % 與上個mfc檔進行串聯
end
Data_other=Data

tested = trainedModel7.predictFcn(Data_other(:,1:26));

% Person
last_frame=0;                                     % 初始化
Correct=0;                                         % 初始化
for person=1:10                                   % 總共10人
    count=last_frame+1;                           % count從上一輪結束開始算
    last_frame=last_frame+Number_other(person,1); % 得知計算到何處為一個人的

    if sum(tested(count:last_frame,1))>Number_other(person)/2 % 若有病音框>一半
        predict(person,1)=1;                       % 若大於一半,predict存1
    else
        predict(person,1)=0;                       % 若小於一半,predict存0
    end
end
predict

```

▲ 問題四的程式碼

predict		
10x1 double		
	1	
1		0
2		1
3		0
4		0
5		1
6		0
7		0
8		1
9		0
10		1

◀ Other 的預測結果
(採用 K=7 去預測)
(其 mfcc 檔按照名稱遞增排列)