

浙江大学

课程名称：数字视音频处理

姓 名：

学 院：计算机学院

专 业：数字媒体技术

学 号：

指导教师：杨莹春

2020 年 01 月 07 日

浙江大学实验报告

课程名称：____数字视音频处理____实验类型：____综合____

实验项目名称：____语音画像____

学生姓名：____专业：____数字媒体技术____学号：____

同组学生姓名：____无____指导老师：____杨莹春____

实验地点：____实验日期：____2020____年____01____月____07____日

一、实验目的和要求

1. 每周录音 1 次以上（录制内容：诗歌朗诵），每次分别用正常、快速、慢速三种方式录制，采集 15 周共 15 次以上语音。保持同一电脑和麦克风，安静环境。
2. 使用 Matlab 的 VOICEBOX 工具箱作 MFCC, GMM 训练与测试。其中第一周的正常语速录音用于训练模型，所有录音用于测试。
3. 找出识别得分偏低的语句，用 PRAAT 分析其与模板语音之间的听感、特征等差异。
4. 通过数据处理和创意设计，展示语音的时间变化趋势。

二、实验器材

操作系统：Windows 10

开发工具：Matlab R2019a, Praat

三、实验步骤

1. 录音

- (1) 录音环境：寝室
- (2) 录音器材：声卡（focusrite2i2），动圈麦（Superlus top248）
- (3) 录音及切分音频所用软件：Adobe Audition 2019CC
- (4) 录音参数：采样率 8000Hz，16bit，单声道，wav 格式

(5) 录音内容:

- ①登鹳雀楼 王之涣
- ②白日依山尽
- ③黄河入海流
- ④欲穷千里目
- ⑤更上一层楼
- ⑥八千里路云和月

2. 训练模型与测试

首先配置 VOICEBOX 工具箱，配置相关路径

(1) 模型训练

首先通过 `audioread()` 函数读取 `training/a/` 文件夹下的六个 wav 文件（第一周录制的六个常速 wav 文件），然后通过 `melcepst()` 函数提取特征位 `train_feature`（12 位 MFCC），最后通过 `gmm_estimate()` 函数训练模型（16 阶 GMM），模型训练结束后将三个分量数据（`mu`, `sigma`, `weight`）保存在 `model/a/` 文件夹下。

具体代码如下：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%%  
  
%本代码适用于单人的说话人确认  
  
clear all;  
close all;  
  
MFCC_size=12;%mfcc 的维数  
GMMM_component=16;%GMM component 个数  
  
  
mu_model=zeros(MFCC_size,GMMM_component);%高斯模型 分量 均值  
sigma_model=zeros(MFCC_size,GMMM_component);%高斯模型 分量 方差  
weight_model=zeros(GMMM_component);%高斯模型 分量 权重  
  
  
train_file_path='.\training\';%模型训练文件路径  
test_file_path='.\testing\';%测试文件路径
```

```

all_train_feature=[];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%train model

FileList=dir(train_file_path);%读取该路径下的所有文件

model_num=1;%注册模型的个数

%该路径下是否是文件夹

for i=1:length(FileList)

    if(FileList(i).isdir==1&&~strcmp(FileList(i).name, '.')&&~strcmp(FileList(i).name, '..'))

        all_model_name(model_num,1)=FileList(i).name;%存储模型名称

        fprintf('Train:%s\n',all_model_name(model_num,1));

        one_train_file_path=[train_file_path

all_model_name(model_num,1) '\'];

    all_train_file=dir(fullfile(one_train_file_path,'/*.wav'));% 读取该路径下的所有文件

    for j=1:length(all_train_file)

        file_name=all_train_file(j).name;%wav 文件名

        train_file=[one_train_file_path file_name];

        fprintf(' train file:%s\n',train_file);

        [wav_data ,fs]=audioread(train_file);%读取音频

        train_feature=melcepst(wav_data ,fs);%提取特征

        all_train_feature=[all_train_feature;train_feature];

    end

    dirName=['.\model\' all_model_name(model_num,1) '\'];

    [mu_model,sigma_model,weight_model]=gmm_estimate(all_train_fea

```

```

ture',GMMM_component); %训练模型

    if ~exist( dirName, 'dir')

        mkdir(dirName);

    end

    save([dirName 'mu_model.mat'],'mu_model');

    save([dirName 'sigma_model.mat'],'sigma_model');

    save([dirName 'weight_model.mat'],'weight_model');

    model_num=model_num+1;

end

end

save('..\model\all_model_name.mat','all_model_name');

```

(2) 测试

首先通过 `audioread()` 函数读取 `testing/` 文件夹下的所有 wav 文件（测试文件）作为测试文件输入，然后通过 `melcepst()` 函数提取特征位 `test_feature`，最后将测试特征和上面训练模型得到的三个参数传入 `lmultigauss()` 函数，得到 `YM`，`Y`，对 `Y` 求 `mean` 即可得到测试语音与对应的模型比对识别得到的得分。

具体代码如下：

```

scores = zeros(15,15);

all_model_name=importdata('..\model\all_model_name.mat');

model_num=length(all_model_name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%test

FileList=dir(test_file_path);%读取该路径下的所有文件

%该路径下是否是文件夹

for i=1:length(FileList)

    if(FileList(i).isdir==1&&~strcmp(FileList(i).name, '.')&&~strcmp(FileList(i).name, '..'))

        test_name=FileList(i).name;
    end
end

```

```

        one_test_file_path=[test_file_path test_name '\'];

all_test_file=dir(fullfile(one_test_file_path,'/*.wav'));% 读取
该路径下的所有文件

        fprintf('测试类型: %s\n',test_name);
        for j=1:length(all_test_file)

            file_name=all_test_file(j).name;%wav 文件名
            test_file=[one_test_file_path file_name];

            [wav_data ,fs]=audioread(test_file);%读取音频
            test_feature=melcepst(wav_data ,fs);%提取特征

            fprintf('Test: %s\n',test_file);

            for k=1:model_num

                model_path=['.\model\' all_model_name(k,1) '\'];
                mu_model=importdata([model_path 'mu_model.mat']);
                sigma_model=importdata([model_path
'sigma_model.mat']);

                weight_model=importdata([model_path
'weight_model.mat']);

                [lYM, lY] = lmultigauss(test_feature', mu_model,
sigma_model, weight_model);

                score(j,k) = mean(lY);

                scores(i,j) = score(j,k);

                fprintf('                                Model:%s
score:%f\n',all_model_name(k,1),score(j,k));

            end

        end

        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %result

        [max_score,max_id]=max(score(:,k));

```

```

        [min_score,min_id]=min(score(:,k));

        fprintf('Max      score:%f          file:%s\nMin      score:%f
file:%s\n\n',max_score,all_test_file(max_id).name,min_score,al
l_test_file(min_id).name);

    end

end

```

3. 语音得分可视化分析

通过运行代码可以得到所有周的所有语句的分数以及每周中的最高分和最低分。统计所有的数据，以周为横坐标，得分为纵坐标展示每周所有的语句的得分，通过可视化分析可以更直观地发现并记录得分较低的语句。




4. PRAAT 分析识别得分较低的语句

选出比较低的分数所对应的语句，在 Praat 中将其与其对应的模板语句进行对比分析语音特征差异。

四、实验结果与分析

1. 训练模型与测试

模型训练结束后在 model/a/ 文件夹下生成模型的三个分量数据：

 mu_model.mat
 sigma_model.mat
 weight_model.mat

测试结束后生成以周为单位的录制的 18 个文件（6 个快速文件，6 个常速文件，6 个慢速文件）的分数以及周中最高分和最低分，这里选取第 1 周的数据举例：

```

测试类型: 3170104579-W1

.....

Test: .\testing\3170104579-W1\N1.wav

    Model:a  score:-13.429391

Test: .\testing\3170104579-W1\N2.wav

    Model:a  score:-14.054325

```

Test: .\testing\3170104579-W1\N3.wav

Model:a score:-14.317899

Test: .\testing\3170104579-W1\N4.wav

Model:a score:-12.988226

Test: .\testing\3170104579-W1\N5.wav

Model:a score:-13.457369

Test: .\testing\3170104579-W1\N6.wav

Model:a score:-14.577237

.....

Max score:-12.988226 file:N4.wav

Min score:-16.447595 file:F3.wav

2. 语音得分可视化分析

在网页图表中，横坐标表示周，纵坐标表示得分，每一条线代表相同内容同样速度的一句话，比如：图（Fig-1:total）中高亮线代表 N4 在 15 周的录制过程中发生的变化。

（1）总体视图，可以看到第 13 周的语音得分均偏低，其中第 13 周的 N3 得分最低。



Fig-1:total

(2) 快速语音视图 (Fig-2:fast), 可以看到第 13 周的 F1 得分最低。



Fig-2:fast

(3) 常速语音视图 (Fig-3:normal), 可以看到第 13 周的 N3 得分最低。

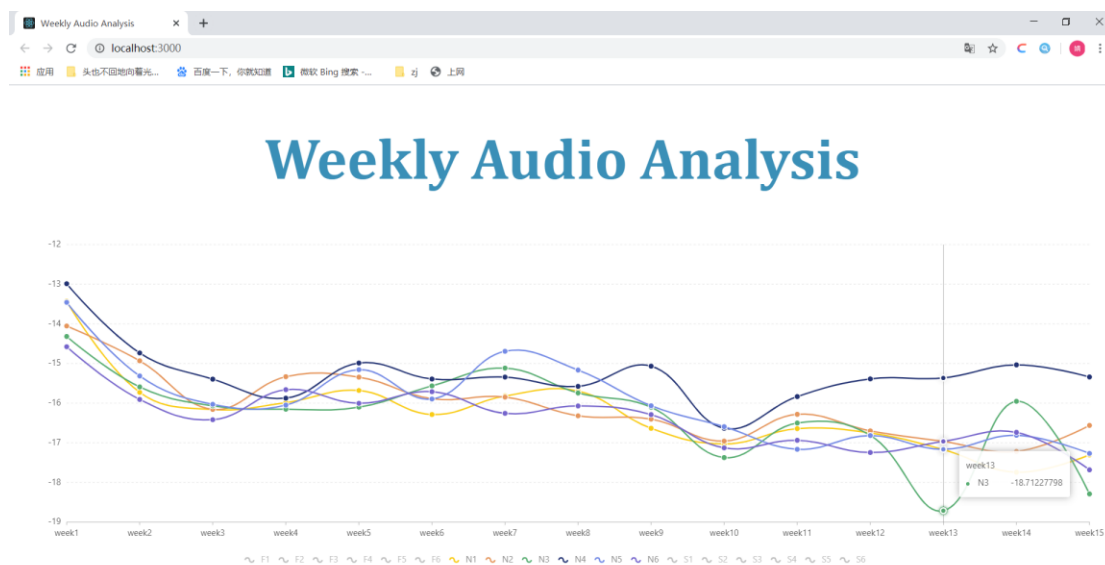


Fig-3:normal

(4) 慢速语音视图 (Fig-4:slow), 可以看到第 13 周的 S3 得分最低。

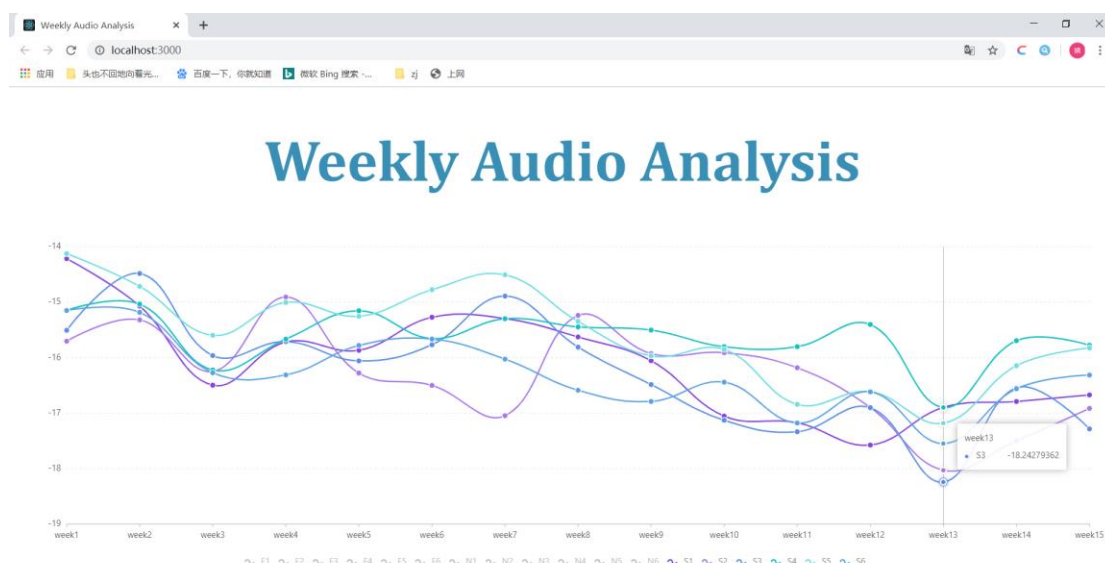
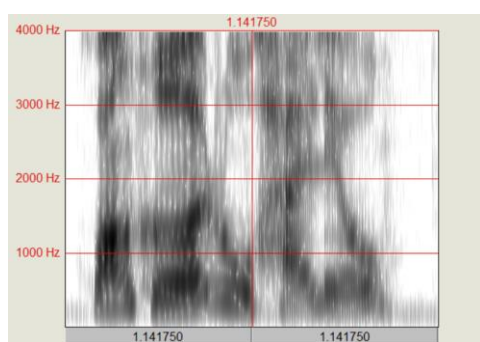


Fig-4:slow

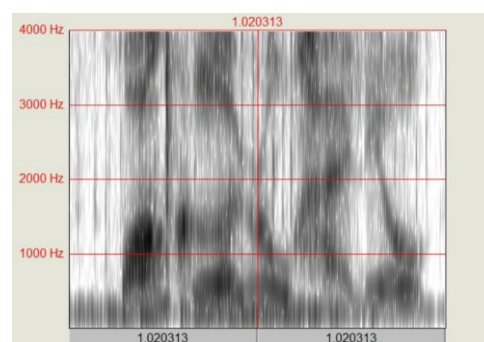
3. PRAAT 分析识别得分较低的语句

从上面的可视化分析中可以看到，第 13 周的录音普遍偏低，我听了一下 13 周的录音，大部分录音有细微的电流声，这可能是导致第 13 周录音普遍偏低的原因。这里我选择得分最低的第 13 周的常速第三条语句（“黄河入海流”）来分析。

(1) 语图



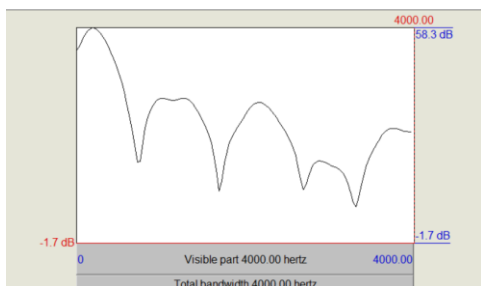
W1-N1 频谱图



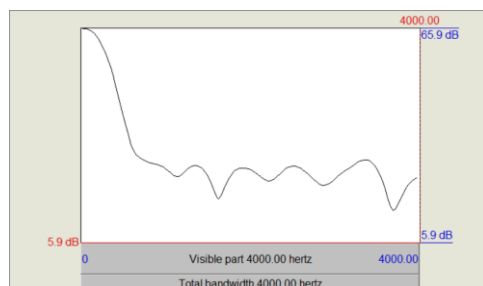
W13-N3 频谱图

根据两句语音的频谱图可以看到，其频率差异很大，整句话中测试录音的频率基本都是明显低于模板录音。

接下来截取两端录音的相同时间节点的 2 维频谱图如下：



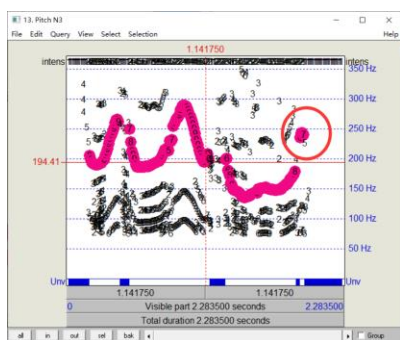
W1-N1 2 维频谱图



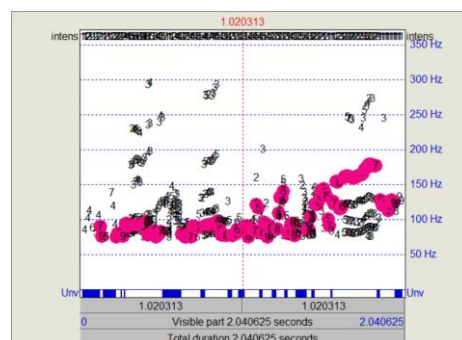
W13-N3 2 维频谱图

根据两句语音的频谱图可以看到，测试录音的基本低于模板录音，并且波动不清晰。

(2) 基频图



W1-N1 基频图

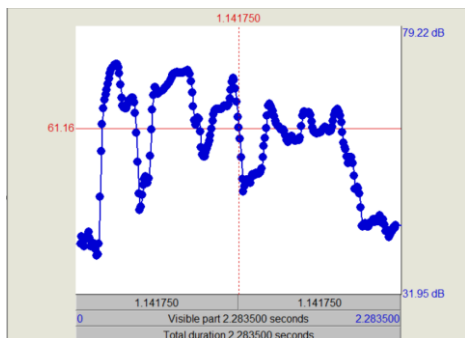


W13-N3 基频图

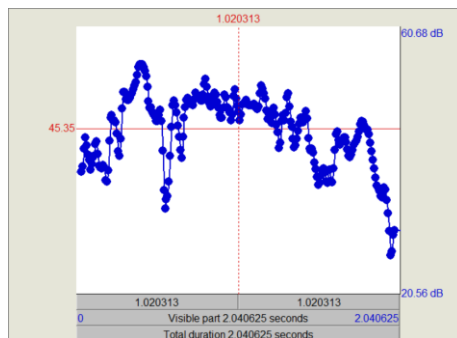
根据两句语音的基频图可以看到，差异很大，整句话中测试录音的基本都是明显低于模板录音，这应该是造成得分较低的主要原因，此外，在模板录音中还存在一个游离于整体之外的奇异点。

我调出了这两句录音的详细信息，模板录音的平均频率为 194.81132113676335 Hz，测试录音的平均频率为 107.45851583493635 Hz，可以看到明显低于模板录音；模板录音的标准差为 43.18208681412932 Hz，测试录音的标准差为 31.81349028245567 Hz，测试录音的起伏小一些，这些在基频图上也可以看出来。

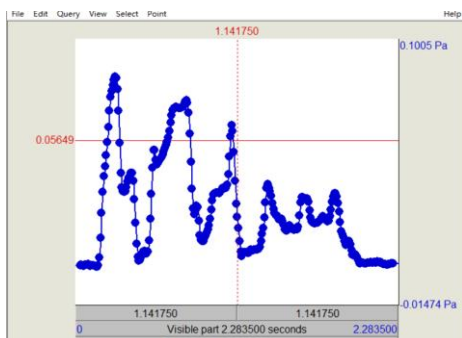
(3) 振幅强度



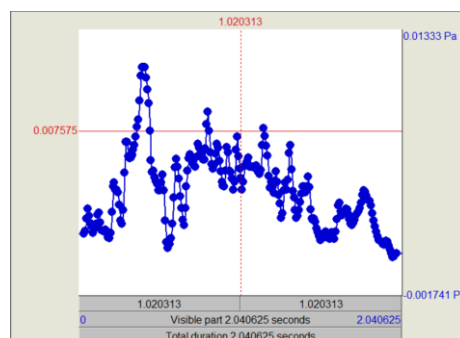
W1-N1 强度曲线



W13-N3 强度曲线



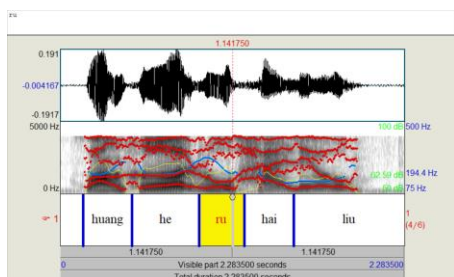
W1-N1 振幅曲线



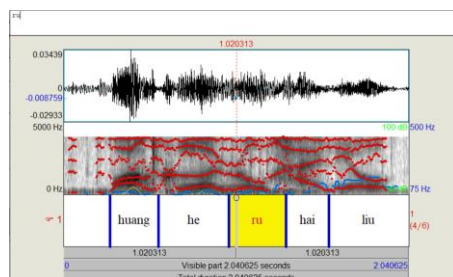
W13-N3 振幅曲线

根据两句语音的强度曲线和振幅曲线可以看到，模板录音和测试录音的大致走向有一些相似，但是具体来看是有很大的差异存在的。

(4) 共振峰分析与语音标注



W1-N1 共振峰与语音标注



W13-N3 共振峰与语音标注

根据两句语音的共振峰可以看到，测试录音与模板录音的共振峰差异也是非常大的，测试录音的音质差了一些，事实上测试录音实际听出来的确有一些明显的杂音，这可能是导致其得分很低的原因。

综上所述，13 周的录音大部分的确有细微的电流声和杂音，这应该是导致第 13 周录音普遍偏低的原因

五、 实验感想及分析

通过本次实验，最大的收获是学会了如何分析自己的语音特征，在做实验的过程中，由于要用到室友的录音设备，也对其安装和设置有了初步的了解，同时也掌握了 Au 录音和剪辑音频的基本操作，掌握了 Praat 软件的使用。

我发现当语音的信息以数字的形式表达出来有助于帮助分析语音的特征和对比关系，将得分可视化时也促使我去学习了 React 框架和 G2 库，最后在网页端写出来了可视化分析图，更加助于直观地辨析自己录音的得分和分布情况。

通过本次实验，我感觉到自己对于语音特征的分析有了更深的理解，受益匪浅。同时，钻研问题和解决问题是很快乐的，而在不断解决问题的过程中我感受到了自己的成

长，感觉很有收获，将视觉上的可视化、听觉上的音频与数字结合起来的感覺也是非常美妙的。