

武汉大学教学实验报告

电子信息学院 电子信息工程专业 2021 年 10 月 12 日

实验名称: 男女声音识别系统 指导教师: 张兰、陈旻哲

组员: 曾鹏宇-2019302120151 年级: 2019 成绩: _____

目录

一、实验目的.....	2
二、实验设计.....	2
2.1 总体思路.....	2
2.2 特征提取.....	2
2.3 分类器设计.....	4
2.4 测试与蒙特卡罗.....	4
三、实验过程.....	5
3.1 系统介绍.....	5
3.2 代码实现.....	8
四、实验结果.....	10
4.1 普通模式实验结果.....	10
4.2 蒙特卡罗分析结果.....	11
五、结论分析.....	12
六、参考文献.....	13
附录 A 组员分工.....	错误!未定义书签。

一、实验目的

1. 实现系统进行男女声音识别；
2. 实现声音时域图与频谱图的显示；
3. 实现声音的加减速、升高降低音量、倒放等功能；
4. 添加新的声音特征提取，丰富描述声音的特征；
5. 添加新的分类器，加入逻辑回归分类器；
6. 结果采用蒙特卡罗方式显示；
7. 实现 GUI 界面，提高人机交互功能。

二、实验设计

2.1 总体思路

1. 首先对数据集中的音频提取特征，特征需要能准确反映男女声音的区别。使用特征与原始标签训练分类器，得到训练完成的分类器参数，同时进行测试得到准确率和召回率以描述分类器性能。
2. 使用训练好的分类器，提取用户选择或录入的音频的特征，作为分类器的输入得到预测的结果进行输出。
3. 同时增加蒙特卡罗统计测试，对分类器进行重复训练与测试，得到多组准确率与召回率数据，进行各种特征与分类器组合效果的比较分析。

2.2 特征提取

1. LPCC

线性预测倒谱系数 (Linear Prediction Cepstrum Coefficient, LPCC) 基于语音信号为自回归信号的假设，获得能反映声道信息的倒谱系数。

所谓自回归信号即信号内部不同数据点之间存在相关关系，因此可以使用少量已知时间内的信号反映将产生的信号。该过程类似于人发声的机制，人发声的过程近似于阶跃响应，阶跃信号经过声道之后，已知时间点的幅值进行线性相加，获得预测时间点的幅值。

线性相关预测，即 LPC (Linear Prediction Coefficient) 便采用该原理获得已知时间点线性相加过程中的系数。

LPC 编码的基本思想：“一个语音取样的现在值可以用若干个语音取样过去值的加权线性组合来逼近”，用过去 p 个样本点预测当前值：

$$\hat{x}[n] = \sum_{k=1}^p a_k x[n-k]$$

在线性组合中的加权系数 a_k 称为预测器系数。通过使实际语音抽样和线性预测抽样之间差值的平方和达到最小值，能够决定唯一的一组预测器系数。

求取 a_p 最常用的一个方法就是最小化真实信号与预测信号之间的均方误差 (Mean Squared Error, MSE)。MSE 期望可写作：

$$e(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{i=1}^p a_i x(n-i)$$

a_i 求偏导可得:

$$\sum_n x(n)x(n-j) = \sum_{i=1}^p a_i \sum_n x(n-i)x(n-j)$$

$$E = \sum_n [x(n)]^2 - \sum_{i=1}^p a_i \sum_n x(n)x(n-i)$$

Yule-Walker 方程):

$$R(j) = - \sum_{i=1}^p a_i R(j-i) \quad 1 \leq j \leq p$$

Toeplize 矩阵, 利用 Durbin 算法求解此矩阵, 即可算出滤波器系数 a_i 。

LPCC 则先提取 LPC 系数, 再转换到倒谱上。LPCC 可以反映声道的特征, 对元音描述能力较好。

LPCC 的求取方法: 由于通过自相关法求得的 LPC 系数保证了系统的稳定性, 使得下式所对应的声道模型传输函数具有最小相位。

$$H(z) = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}}$$

利用这一特点, 可推导出语音信号的倒谱和 LPC 系数之间的递推关系:

$$\begin{cases} c(1) = a_1 \\ c(n) = a_n + \sum_{k=1}^{n-1} \left(a - \frac{k}{n}\right) a_k c(n-k), 1 < n \leq p \\ c(n) = \sum_{k=1}^p \left(a - \frac{k}{n}\right) a_k c(n-k), n > p \end{cases}$$

2. MFCC

梅尔倒谱系数(Mel Frequency Cepstrum Coefficient, MFCC)考虑人类的听觉特征, 人对各种频率成分的敏感程度并非呈线性关系。MFCC 先将原频谱映射到基于听觉感知的 Mel 非线性频谱中, 然后转换到倒谱上。

梅尔倒谱系数可以较为准确的反映人耳对声音的敏感度, 能反映人声辅音的特点。

3. 基音频率

人的发声根据声带震动的方式的不同, 将声音信号分为清音和浊音。其中浊音需要声带周期性震动, 所以具有明显的周期性, 这种声带振动的频率称为

基音频率，相应的周期就成为基音周期。而清音没有明显的周期性。一般来说，男性说话者的基音频率较低，而女性说话者的基音频率相对较高。

2.3 分类器设计

1. 朴素贝叶斯原理

朴素贝叶斯分类器是分类算法集合中基于贝叶斯原理的一种算法。贝叶斯理论在数学上可表示为：

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

朴素贝叶斯通过获得训练过程中数据集中每类特征在对应标签中的频率作为进行预测时的概率，当输入新的预测对象时，提取特征，并由特征对应的概率进行求和，计算出对应各个标签的概率，而可能性最高的标签即为预测标签。

2. 逻辑回归原理

逻辑回归是一种广义线性回归，通过构造归一化概率函数与误差函数，对比实际标签计算得到误差，使用梯度下降法修改权重参数，最终得到训练完成的权重参数。分类器将提取到的特征与对应的权重参数运算得到标签的概率。

2.4 测试与蒙特卡罗

在分类器训练过程中，会将数据集划分为训练样本与测试样本，在训练完成后，将测试样本输入到分类器得到预测结果，通过预测结果与真实结果进行计算获得该分类器的准确率与召回率。

准确率（Precision）即指：所有预测结果与实际结果一样的样本/所有样本。反映预测结果的准确性。

召回率（Recall）即指：预测为某类标签的数量/样本中该类标签真正的数量。反映对某类标签的识别能力。

蒙特卡罗是一类随机方法的统称。这类方法的特点是，可以在随机采样上计算得到近似结果，随着采样的增多，得到的结果是正确结果的概率逐渐加大。即通过多次重复实验得到较为稳定、准确的结果。本项目采用蒙特卡罗方法对各种特征与分类器的组合的准确率以及召回率进行统计分析。

三、实验过程

3.1 系统介绍

系统在 MATLAB 中的 App designer 中完成。首先分别实现各种特征提取函数与分类器算法函数，利用 UI 界面设计工具完成界面设计。

App 初始界面如下图所示：



图 1: app 初始界面

最上方为菜单栏，可以选择分类器训练设置、导入数据集与模式选择。

使用过程中先进行分类器训练设置，可选择训练/验证数据比例、分类器种类与特征。设置过程中会在左方菜单显示对应的设置。

正常模式将进行单次训练，训练完成后将显示准确率与召回率，对于逻辑回归还将显示训练迭代过程中误差下降的过程。如下图所示：

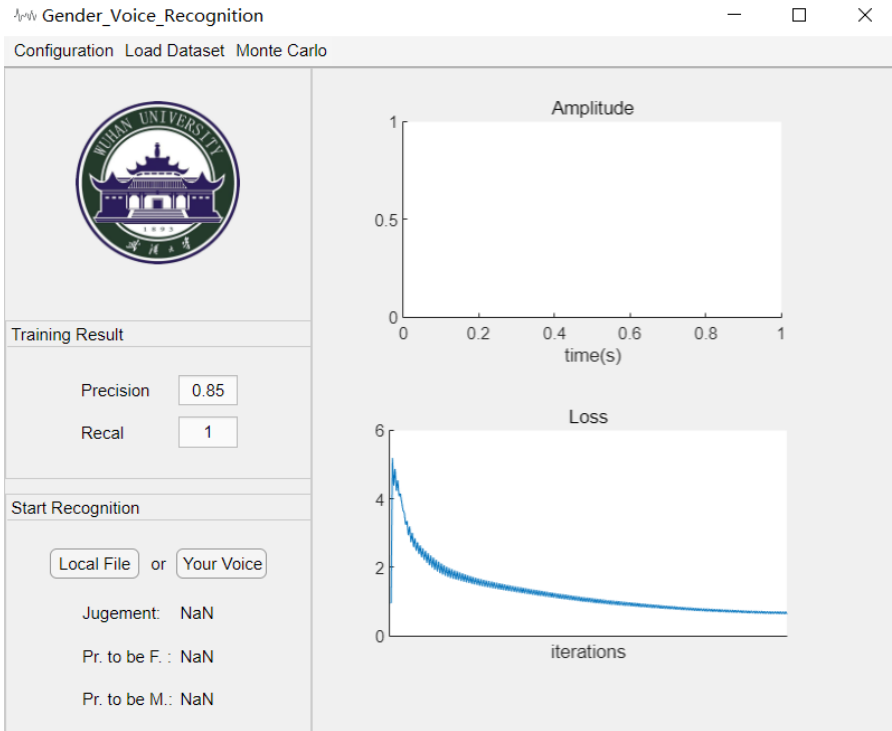


图 2：普通模式训练完成界面显示

随后可进行声音的识别，可选择导入本地文件或者录音。导入完毕或录音完毕即开始识别，将声音提取特征后输入到分类器得到分类结果。右侧将显示声音的时域波形与频谱图，同时有播放、倒放、加减速、加減音量等功能。

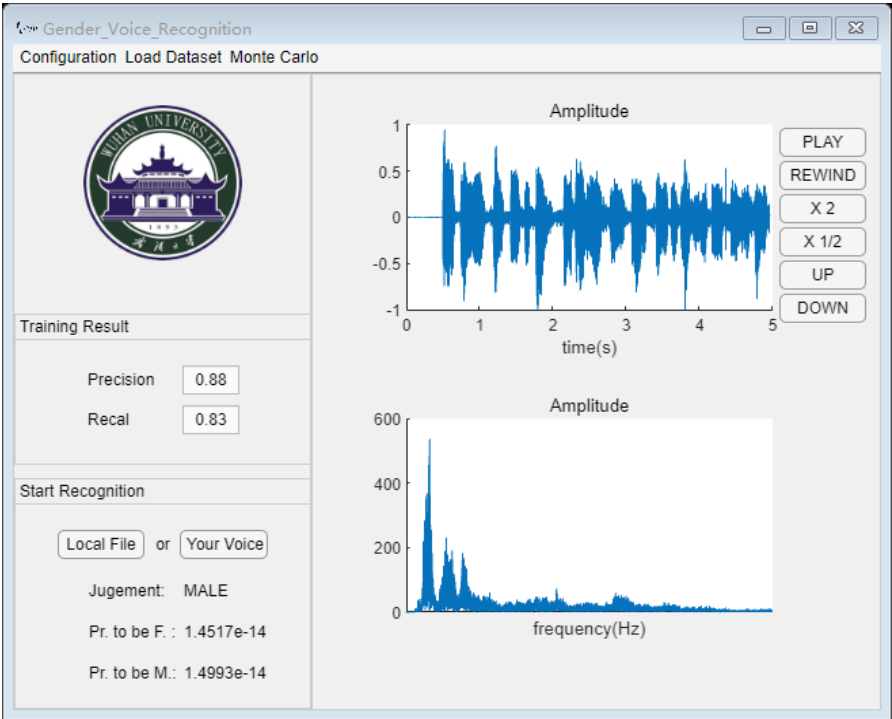


图 3：导入声音进行识别

若点击菜单选择蒙特卡罗测试模式，则界面如下所示，可填入训练次数。

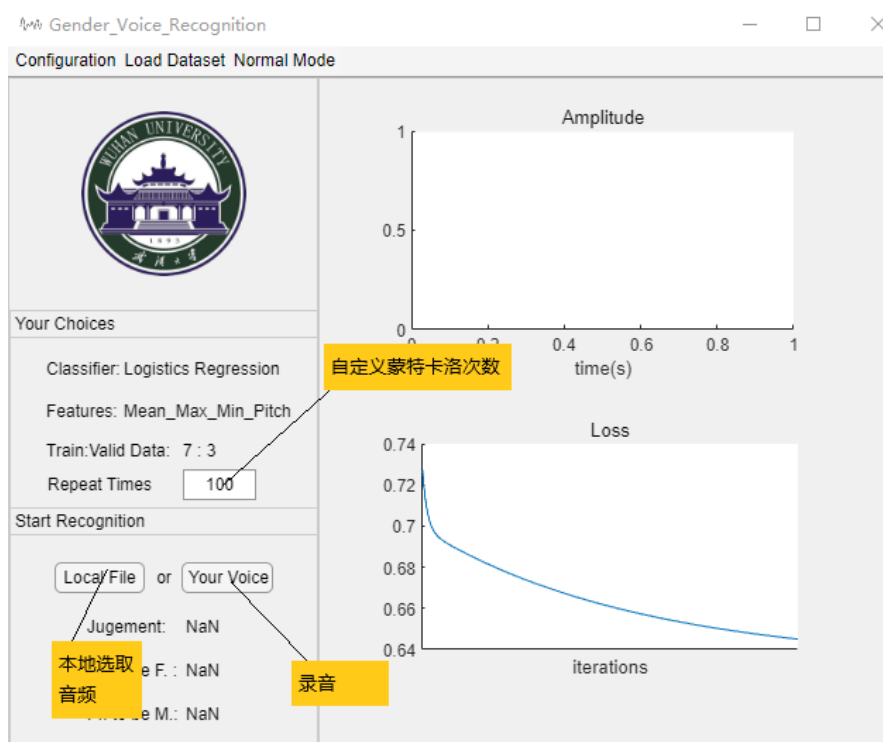


图 4：蒙特卡罗测试模式设置界面

导入数据集后开始训练，得到准确率、召回率的分布图。如下图所示

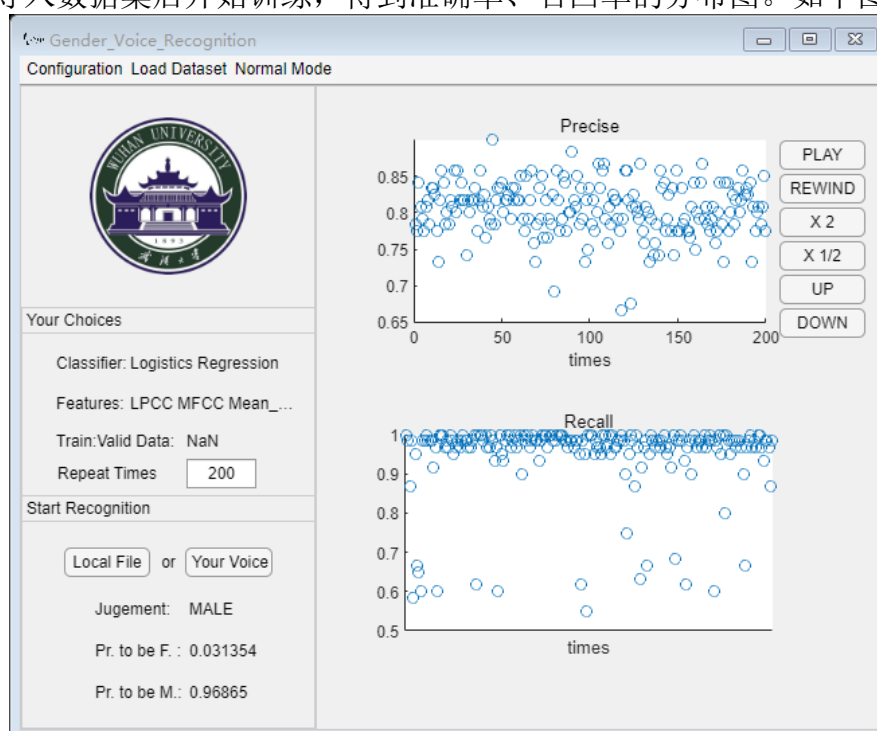


图 5：蒙特卡罗测试结果显示

可用于对比分析各种分类器的性能特点，实际应用中更好地选择特征与分类器算法的搭配。

3.2 代码实现

后端部分代码如下图所示：

```
for j=1:VOICES(i).num
    [currentvoice,fs] = audioread([app.datasetpath,'\',VOICES(i).name,'\ ',file_list{i}(j+2).name]);
    allfts = myfeature(currentvoice,fs); % 提取各种特征
    voiced = mygetvoiced(currentvoice,fs);

    % 根据用户的选择提取特征
    feature = [];
    if app.Mean_PitchCheckBox.Value==1 % 选择了 meanpitch
        feature = [feature allfts(11)];
    end

    if app.Max_PitchCheckBox.Value==1 % 选择了 maxpitch
        feature = [feature allfts(12)];
    end

    if app.Min_PitchCheckBox.Value==1 % 选择了 minpitch
        feature = [feature allfts(13)];
    end

    if app.MFCCCheckBox.Value==1
        feature = [feature allfts(14:26)];
    end

    if app.LPCCCheckBox.Value==1
        LPCC=yourfeature1(voiced,fs);
        feature = [feature LPCC(1:10) LPCC(18:27) LPCC(35:44)];
    end

    if isempty(feature)
        allWidgetEnable(handles,1);
        errordlg('错误！你至少需要选择一种特征','错误提示');
        return;
    end
    VOICES(i).data(j,:) = feature; % 转为行向量存入结构体中
    fprintf('第%d类/第%d条\n',i,j);
end
%fprintf('第%d类/共%d类语音(共%d条数据)特征提取完毕\n',i,category_num,VOICES(i).num);
end
```

图 6：特征提取过程代码

在命令行中反馈每条语音的提取过程，可以有效反馈训练进度。


```

if app.RepeatTimesEditField.Visible=='on'
    hs = msgbox('正在进行蒙特卡罗测试','提示');
    ht = findobj(hs, 'Type', 'text');
    set(ht,'FontSize',8);
    set(hs, 'Resize', 'on');
%%
app.mc_results_p=[];
app.mc_results_r=[];
for i=1:app.mc

    category_num = size(VOICES,1);
    ttraindata = [];
    ttrainlabel = [];
    tvaldata = [];
    tvallabel = [];
    train_index_remember = [];
    for i=1:category_num
        file_num = VOICES(i).num;
        [val_index,train_index] = crossvalind('holdout',file_num,app.TrainSegRatio);
        ttraindata = [ttraindata;VOICES(i).data(train_index,:)];
        ttrainlabel = [ttrainlabel;(i)*ones(sum(train_index),1)];
        tvaldata = [tvaldata;VOICES(i).data(val_index,:)];
        tvallabel = [tvallabel;(i)*ones(sum(val_index),1)];
        train_index_remember = [train_index_remember;train_index];
    end

    % 一些参数
    male_train_num = sum(train_index_remember(1:size(VOICES(1).data,1)));
    male_val_num = size(VOICES(1).data,1) - male_train_num;
    female_train_num = sum( train_index_remember(size(VOICES(1).data,1)+1:size(VOICES(1).data,1)+size(VOICES(2).data,1) ));
    %上一行代码修改了表述,可多分类拓展
    female_val_num = size(VOICES(2).data,1) - female_train_num;

    app.traindata = ttraindata;
    app.trainlabel = ttrainlabel;
    app.valdata = tvaldata;
    app.vallabel = tvallabel;

    %% 训练
    % 开始训练分类器
    if app.classifer_selected==1
        [L,app.Logistic_Reg]=train_logistic_reg(app.traindata,app.trainlabel,2.4,300);

%% 训练
% 开始训练分类器
if app.classifer_selected==1
    [L,app.Logistic_Reg]=train_logistic_reg(app.traindata,app.trainlabel,2.4,300);

    %plot(app.UIAxes_2, L);

    predicted_label=real_logistic_reg(app.valdata,app.vallabel,app.Logistic_Reg);
    cfsmtx = mycfsmtx(app.vallabel,predicted_label);
    app.mc_results_p = [app.mc_results_p cfsmtx(end,end)]; % 准确率
    app.mc_results_r = [app.mc_results_r cfsmtx(1,end)]; % 召回率

elseif app.classifer_selected==2
    % 量化预处理
    ftsnum = size(app.traindata,2); % 特征数值
    traindata_qualify = zeros(size(app.traindata));
    valdata_qualify = zeros(size(app.valdata));
    for i=1:ftsnum
        traindata_qualify(:,i) = mydiscretization(app.traindata(:,i),20); % 特征数值量化阶ftsRange = 20;
        valdata_qualify(:,i) = mydiscretization(app.valdata(:,i),20);
    end
    % 训练测试
    [TrainingSets,ValidationSets] = myNaiveBayesTrain(traindata_qualify,app.trainlabel,...
        valdata_qualify,app.vallabel,20);
    predicted_label = myNaiveBayesValidation(TrainingSets,ValidationSets);
    app.NBTrainingSets = TrainingSets;
    app.NBValidationSets = ValidationSets;
    cfsmtx = mycfsmtx(app.vallabel,predicted_label);
    % 将识别错误的数据的序号找到
    app.mc_results_p = [app.mc_results_p cfsmtx(end,end)]; % 准确率
    app.mc_results_r = [app.mc_results_r cfsmtx(1,end)]; % 召回率
else
    errordlg('错误! 该分类器不存在','错误提示');
end
end
end

```

图 7：蒙特卡罗测试过程代码

```

X1=data;
Y1=label-ones(size(label));
m1 = size(X1,1);
Features=size(data,2); %特征个数
theta1=rand(1,Features);
%num 最大迭代次数
L=[];%记录误差
] while(num)
    dt=zeros(1,Features);
    loss=0;
]   for i=1:m1
        xx=X1(i,1:Features);
        yy=Y1(i,1);
        h=1/(1+exp(-(theta1 * xx')));%归一化概率函数
        dt=dt+(h-yy) * xx;%运用梯度下降算法、
        loss=loss+ yy*log(h)+(1-yy)*log(1-h);%构造误差函数
    end
    loss=-loss/m1;
    L=[L, loss];

    theta2=theta1 - delta*dt/m1 ;
    theta1=theta2;
    num = num - 1;

    if loss<0.01
        break;
    end
end
Loss=L;
th=theta1;

end

```

图 8：逻辑回归训练函数

实验结果

4.1 普通模式实验结果

系统人机交互友好，运算速度较快。采用逻辑回归与朴素贝叶斯分类器均可以较为准确的判断说话者的性别。其中特征采用基音频率+MFCC+LPCC 的组合可以较好的保证语音识别的准确性。

4.2 蒙特卡罗分析结果

实验结果如下图所示：

1. 仅选择基音频率特征：

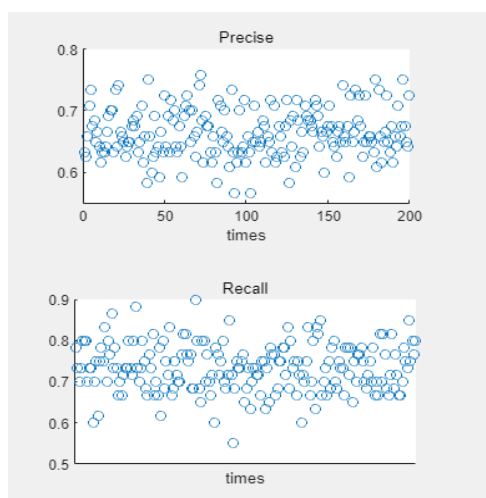


图 9(a)：逻辑回归测试结果

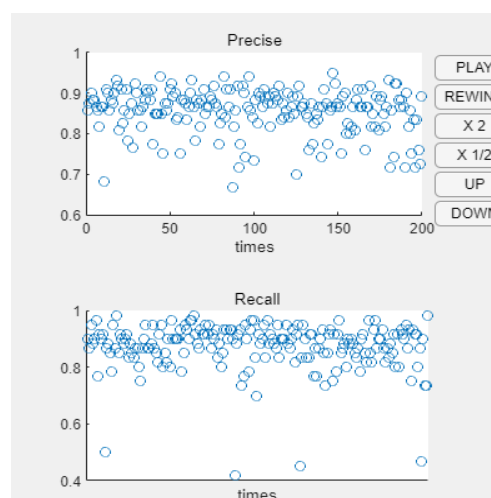


图 9(b)：朴素贝叶斯测试结果

2. 仅选择 MFCC 特征：

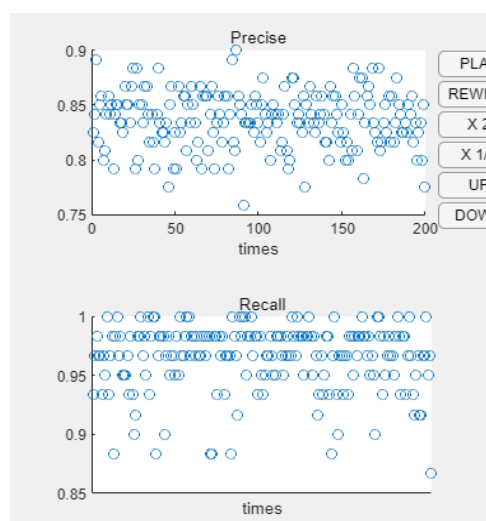


图 10(a)：逻辑回归测试结果

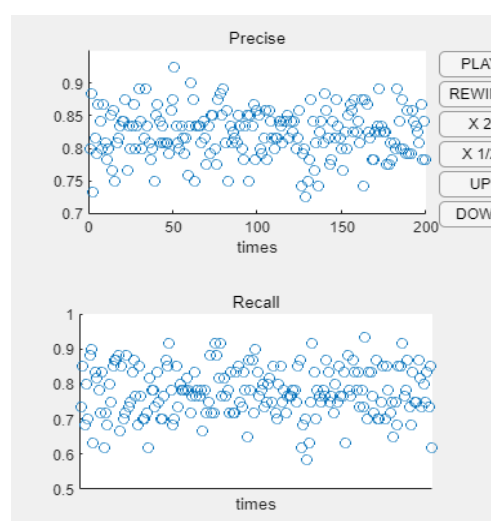


图 10(b)：朴素贝叶斯测试结果

3. 选择基音频率+MFCC 特征:

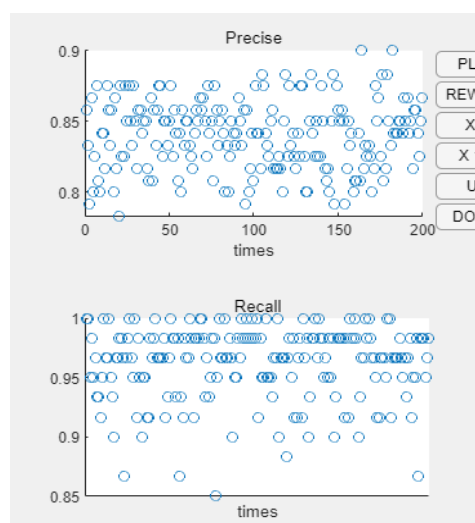


图 11(a): 逻辑回归测试结果

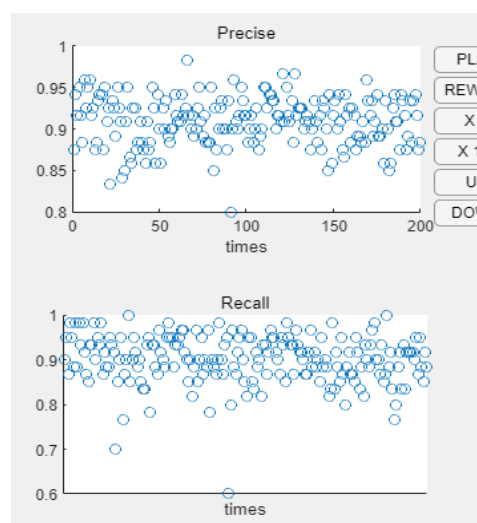


图 11(b): 朴素贝叶斯测试结果

4. 选择基音频率+MFCC+LPCC 特征:

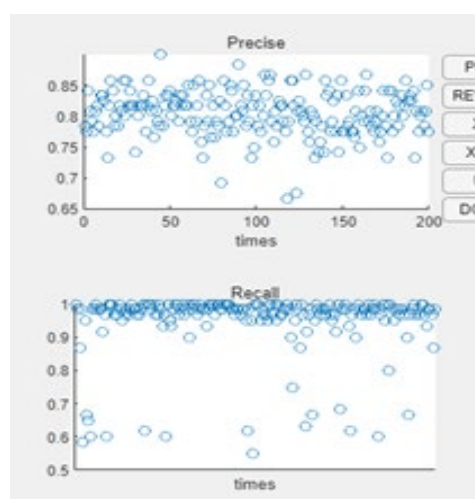


图 12(a): 逻辑回归测试结果

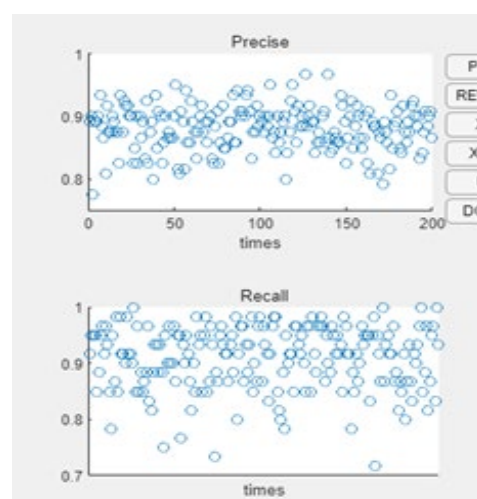


图 12(b): 朴素贝叶斯测试结果

五、结论分析

1. 从频谱图中可看出女性声音音调普遍高于男性，其高频成分多于男性。
2. 对于两种分类器，随着选取特征的增加均会使测试结果的准确率、召回率上升。在多数特征组合下，逻辑回归的召回率相较于朴素贝叶斯更优，数值更高且更为稳定集中；而朴素贝叶斯分类器的准确率较高。总体而言朴素贝叶斯效果更优，逻辑回归分类器训练过程中存在拟合线性。
3. 特征采用基音频率+MFCC+LPCC 的组合可以较好的平衡准确率与召回率，能保证语音识别的准确性。LPCC 虽然较大的提高了运算负担，但也丰富了系统对语音的特征描述能力。
4. 理论上使用 MFCC 与 LFCC 的组合可以实现对人的声音较为全面的特征描述，因此后期拟采用本系统进一步实现文字识别等功能。

六、参考文献

1. Namrata Dave, Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. Volume 1, Issue VI, July 2013
2. 语音信号处理——线性预测编码 LPC:
<https://blog.csdn.net/eqiang8848/article/details/92962867>
3. 王彪, 基于 LPCC 参数的语音识别, 电子设计工程, 2012, 4:18-19
4. 逻辑回归 (Logistic Regression) (一) :
<https://zhuanlan.zhihu.com/p/28408516>