查询到的特征值提取后处理的程序

特征值提取程序

function [shiyu]= feature2(x)

N = length(x);

%1 均值

p1 = mean(x);

% 2方根幅值

p2 = mean(**sqrt**(**abs**(x)))^2;

% 3均方根值

p3 = **sqrt**(mean(x.^2));

% 4最大值

p4 = max(**abs**(x));

% 5标准差

p5 = std(x);

% 6偏斜度

p6 = sum((x-p1).^3)/((N-1)\*p5^3);

% 7峭度

p7 = sum((x-p1).^4)/((N-1)\*p5^4);

% 8峰值因子

p8 = p4/p3;

% 9裕度因子

p9 = p4/p2;

% 10波形因子

p10 = p3/mean(**abs**(x));

% 11脉冲因子

p11 = p4/mean(**abs**(x));

%12方差

p12=var(x);

shiyu=[p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11];

%shiyu=[p2 p3 p10 p11 p5 p7];

%shiyu=[p3 p5 p7 p8 p10];

【读代码时的函数查询】

size（）：获取矩阵的行数和列数

（1）s=size(A),

当只有一个输出参数时，返回一个行向量，该行向量的第一个元素时矩阵的行数，第二个元素是矩阵的列数。

（2）[r,c]=size(A),

当有两个输出参数时，size函数将矩阵的行数返回到第一个输出变量r，将矩阵的列数返回到第二个输出变量c。

1. size(A,n)如果在size函数的输入参数中再添加一项n，并用1或2为n赋值，则 size将返回矩阵的行数或列数。其中r=size(A,1)该语句返回的时矩阵A的行数， c=size(A,2) 该语句返回的时矩阵A的列数。

这周主要还是分析上几周的内容，随着这几周看论文和查阅资料后，对语音的相关特征提取原因与步骤有了更加深刻的了解，以老师在群里给的论文为依据查阅相关资料对SVM做到深入熟悉与了解。

论文：《基于SVM的情感分类》

**步骤：**

1. 情感语音数据库的建立；
2. 语音信号情感特征的提取；
3. 语音情感分类器的选择与优化。

**第一部分：情感语音数据库的建立**

前提：

语音的特征可以表现出说话人的情感，此为语音情感，而映射到音乐中我们也可以根据音乐的特征来提取出音乐包含的情感，在其他的论文中往往借助音乐的高潮部分来作为实验来源提取特征后进行分类。其实关于特征提取的论文都是依据语音的特征，包括韵律，节奏等，这些都可以通过在时域和频域不同的组合表现出来。在时域上会提取时域上短时能量，短时幅度，短时过零率，短时自相关（也就是基频），这些都是根据我们人类的发声规律为参考，用于区分浊音与清音。频域上含有频谱，相位谱，幅度谱等，以及梅尔倒谱系数MFCC，频率上往往需要经过傅里叶变换。

素材来源：

德语 EMO-DB 情感语料库；中科院自动化所的 CASIA 中文语料库。为语音情感识别模型的建立提供训练数据与测试数据。这两个资源库不能直接通过网站进行下载。

利用空间维度进行情感描述的方式是将情感描述为一个情感空间。在这个空间中，每一个维度表示情感的一个心理学特性。因此很多研究人员认为情感具有几个基本的类型，并且能够进行相应的模型建立，而其他情感的形成就是这几种基本情感的不同比例的组合，从而形成新的情感。

**第二部分特征提取过程：**

（1）语音的预处理：通过老师发的论文再一次强化理解，感觉以前不是很了解其意义

1.为什么将语音分为短时？

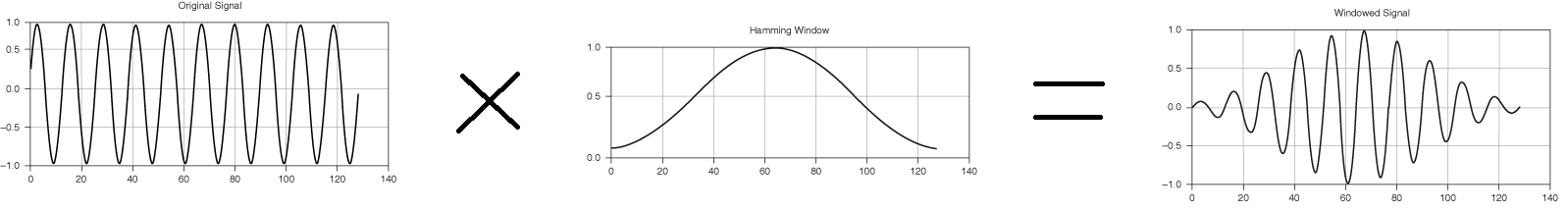
因为大多数语音都是非平稳的信号，因此不能直接用我们数字信号处理中的分析方法去分析，但是根据声带发生的规律我们可以知道语音信号在较短一段时间内近似于平稳，因此可以将其分为一小段一小段的语音信号。

2.分帧：何为分帧？以及如何确定及帧长？

将上述所说的语音信号分成一小段一小段就是分帧，而每一段的帧长该如何确定呢？我们从上述中也能知道，如果语音分段的时长较长，即帧长过长，就会导致每一帧不能接近平稳的信号。若如果语音分段的时长较短，即帧长过短，就会导致每一帧包含的语音信息过少，频率分辨率降低。而人声的基频范围下限在100 Hz左右（有些男声会更低），换算成周期是10 ms，所以一般帧长都选在20 ~ 50 ms。

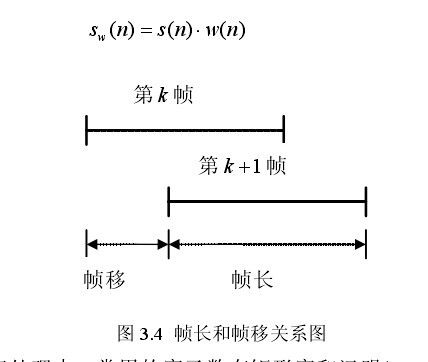
3.加窗：何为加窗？

我们可以想到学习到的数字信号处理中的截断效应知识，对语音信号进行加窗，即让语音信号乘以一个窗函数，窗函数两端变为0的过程尽可能要平滑变到0，尽量不要突变，减小截断效应。因此，加窗的作用是为了能够尽量的增大旁瓣的衰减，从而使得在分帧时，滤波器能够得到较为平坦的通带幅度响应以及较小的波纹衰减。



4.帧移：何为帧移及其作用？

从上图中我们可以看出来当一个信号乘以窗函数是，两端的信号有可能发生衰减。因此加窗的代价就是一帧信号两端的部分被削弱了，没有像中央的部分那样得到重视。这个时候就需要帧移了，帧移就是分帧的时候要相互重叠的截取，通常截取上一帧的一半或者10ms。



（2）特征的提取：就需要在相关代码进行实现

帧移

加窗

分帧

如图所示，经过上述的处理后，我们就可以得到一个接近于平稳的信号，接下来就是可以利用数字信号处理技术将其进行时域和频域的特征提取。提取的特征有短时能量，短时过零率，短时平均幅度，短时自相关函数，梅尔倒谱系数等在以前的周报中都有详细的介绍，而且老师发的论文中的解释同样通俗易懂。

线性可分支持向量机

1. **分分类器的选择与优化：**

近似线性可分支持向量机

分类支持向量机

非线性可分支持向量机

支持向量机

线性回归支持向量机

回归支持向量机

非线性回归支持向量机

以上则为支持向量机的分类

（1）何为支持向量机

在以前的周报中有详细的基础解释

（2）支持向量机算法的提出

为了进一步提高支持向量机的分类能力

1. 根据其特点支持向量机的特点提出一些快速学习算法将支持向量机进行优化训练；
2. 而针对分类支持向量机又有一些算法的提出用来优化支持向量机的的参数（如遗传算法优化支持向量机，蚁群算法优化支持向量机，粒子群算法优化支持向量机）。其中论文中就使用到经粒子群优化算法（PSO）优化后的支持向量机（SVM）作为情感识别分类器。

（3）支持向量机参数的提出原因

因为我们将数据进行分类时，会存在线性可分和线性不可分两种的情况：面对线性可分，可以在低维度对其进行处理；当线性不可分时是需要通过核函数将其映射到高维，然而映射到高维后也依旧存在线性不可分情况，这时就需要合理的设置支持向量机的参数。

（4）支持向量机的核函数的参数

关于核函数介绍的文章

<http://www.cnblogs.com/xiaohuahua108/p/6146118.html>

<http://blog.csdn.net/batuwuhanpei/article/details/52354822>

1. 支持向量机自身参数：惩罚参数 c——表现了支持向量机模型对于误差的容忍程度，c越小的话说明容忍度越大，则产生的误差就越多，分类的正确率就越低；反之，容忍度大，准确性越高，但又不可过于小。
2. 支持向量机核函数：支持向量机有很多的核函数可以进行高维度的分类，在映射过程中我们可以根据训练样本的准确性才判断使用哪种核函数更好,，但是网上说核函数的选择是一个难题，通常高斯核函数会比较好用。核函数的参数有：

（a）线性核函数参数：主要用于线性可分的情况，其参数少速度快。

（b）多项式核函数参数：可以实现将低维的输入空间映射到高纬的特征空间，多项式核函数的参数多，当多项式的阶数比较高的时候，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度会大到无法计算。

（c）高斯核函数中核宽度γ：语音情感特征是线性不可分的，高斯径向基核函数在解决线性不可分问题是由较好的表现，且需要的决定参数较少。γ表示了支持向量机的关联程度，如果 γ 很小，支持向量之间的联系比较松弛，如果γ很大，支持向量之间的影响会过于强大，预测模型难以达到足够的精度。

（d）Sigmoid 核函数中的δ：采用sigmoid核函数，支持向量机实现的就是一种多层神经网络。

（5）核函数的选择方法

* 如果特征的数量大到和样本数量差不多，则选用LR或者线性核的SVM；
* 如果特征的数量小，样本的数量正常，则选用SVM+高斯核函数；
* 如果特征的数量小，而样本的数量很大，则需要手工添加一些特征从而变成第一种情况。

（6）SVM 分类器优化方面

在 SVM 分类器优化方面，使用粒子群优化算法（PSO）对 SVM分类器的惩罚参数 c

核函数参数 γ 的组合进行寻优，找出能够使 SVM 分类器分类精度达到最大的参数组合 从而使语音情感识别的识别率得到提升。

梅尔倒谱系数(mfcc)代码,只是再做一次理解

[x fs]=wavread('snoring1.wav');

bank=melbankm(24,256,fs,0,0.4,'t');%Mel滤波器的阶数为24，fft变换的长度为256，采样频率为16000Hz

%归一化mel滤波器组系数

bank=full(bank);

bank=bank/max(bank(:));

**for** k=1:12          %归一化mel滤波器组系数

n=0:23;

dctcoef(k,:)=**cos**((2\*n+1)\*k\*pi/(2\*24));

end

w=1+6\***sin**(pi\*[1:12]./12);%归一化倒谱提升窗口

w=w/max(w);%预加重滤波器

xx=**double**(x);

xx=filter([1-0.9375],1,xx);%语音信号分帧

xx=enframe(xx,256,80);%对x 256点分为一帧

%计算每帧的MFCC参数

**for** i=1:size(xx,1)

y=xx(i,:);

s=y'.\*hamming(256);%汉明窗

    t=**abs**(fft(s));%fft快速傅立叶变换

t=t.^2;

    c1=dctcoef\***log**(bank\*t(1:129));

c2=c1.\*w';

m(i,:)=c2';

end

%求取一阶差分系数

dtm=zeros(size(m));

%获取矩阵的行数和列数

**for** i=3:size(m,1)-2

dtm(i,:)=-2\*m(i-2,:)-m(i-1,:)+m(i+1,:)+2\*m(i+2,:);

end

dtm=dtm/3;

%求取二阶差分系数

dtmm=zeros(size(dtm));

**for** i=3:size(dtm,1)-2

dtmm(i,:)=-2\*dtm(i-2,:)-dtm(i-1,:)+dtm(i+1,:)+2\*dtm(i+2,:);

end

dtmm=dtmm/3;

%合并mfcc参数和一阶差分mfcc参数

ccc=[m dtm dtmm];

%去除首尾两帧，因为这两帧的一阶差分参数为0

ccc=ccc(3:size(m,1)-2,:);

ccc

subplot(2,1,1)

ccc\_1=ccc(:,1);

plot(ccc\_1);title('MFCC');ylabel('幅值');

[h,w]=size(ccc);

A=size(ccc);

subplot(212)

plot([1,w],A);

xlabel('维数');

ylabel('幅值');

title('维数与幅值的关系')