

Spectral Entropy

信息熵

对于一个离散的随机变量 X ，设其概率分布为 $f(X) = P(X = x_i) = p_i$

信息熵

$$H(x) = -\sum_{i=1}^n p_i \log_2 p_i$$

Spectral Entropy 谱熵

计算每一帧语音信号的Spectral Entropy

第 n 帧语音信号 x_n 帧长为 l

1、求功率谱

- (1) 离散傅里叶变换
- (2) 取模 平方
- (3) 除以帧长 得到功率谱

2、求能量概率分布 p

- (1) 功率谱除以总能量

此时 $\sum p_n(i) = 1$

3、带入信息熵公式

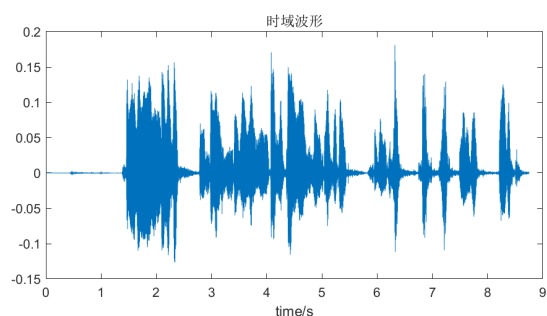
反应信号能量关于频率分布的复杂度。

大概是噪声谱熵高、元音谱熵低。我也不清楚。

Code

读取音频

```
[x, fs] = audioread("audio1.wav");
```



🔍📄🔍📄

分帧

```
%% 分帧
len_win = fs / 40;           % 窗宽      25ms
len_frame = len_win;         % 帧长      25ms
len_inc = floor(len_frame * 0.8); % 帧移      20ms
win = hamming(len_win);       % 海明窗
x_frame = enframe(x, win, len_inc); % 分帧
```

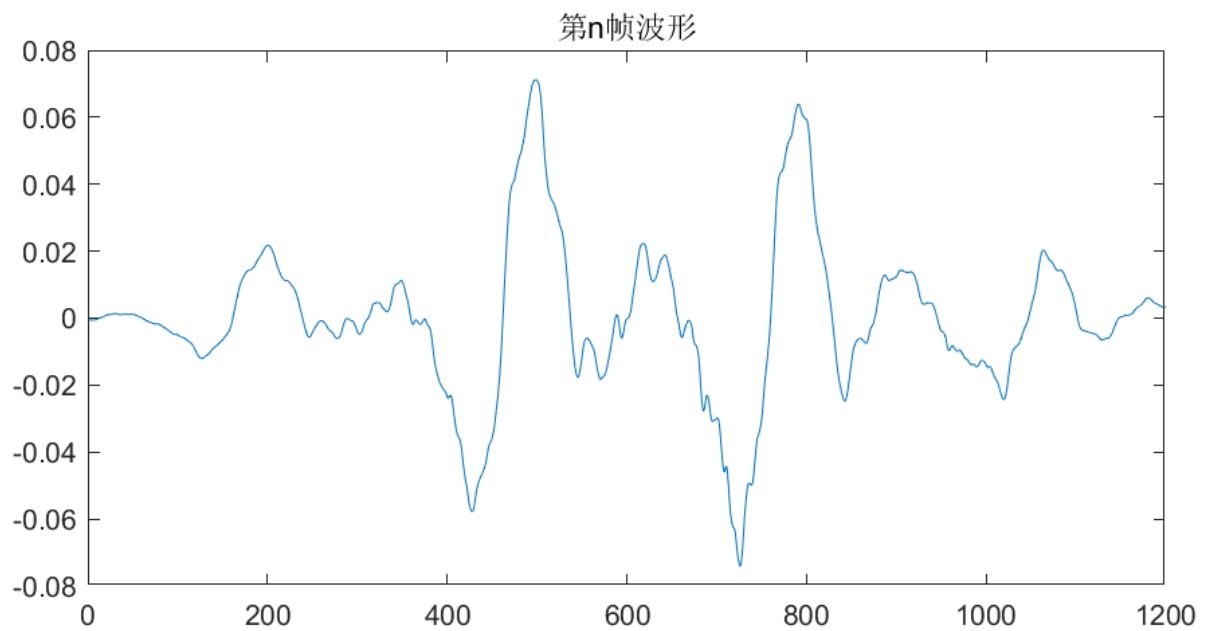
```
function [y_data ] = enframe(x,win,inc)
% in:
    % x: 语音信号
    % win: 窗函数或者帧长
    % inc: 帧移
% out:
    % 分帧后的数组（帧数×帧长）
% =====
L = length(x(:));           % 数据的长度
nwin = length(win);         % 取窗长，数字的长度是1
if (nwin == 1)               % 判断有无窗函数，若为1，表示没有窗函数
    wlen = win;              % 没有，帧长等于win
else
    wlen = nwin;             % 有窗函数，帧长等于窗长
end

if (nargin <3)               % 如果只有两个参数，inc = 帧长
    inc = len;
end

fn = floor((L - wlen)/inc) + 1; % 帧数

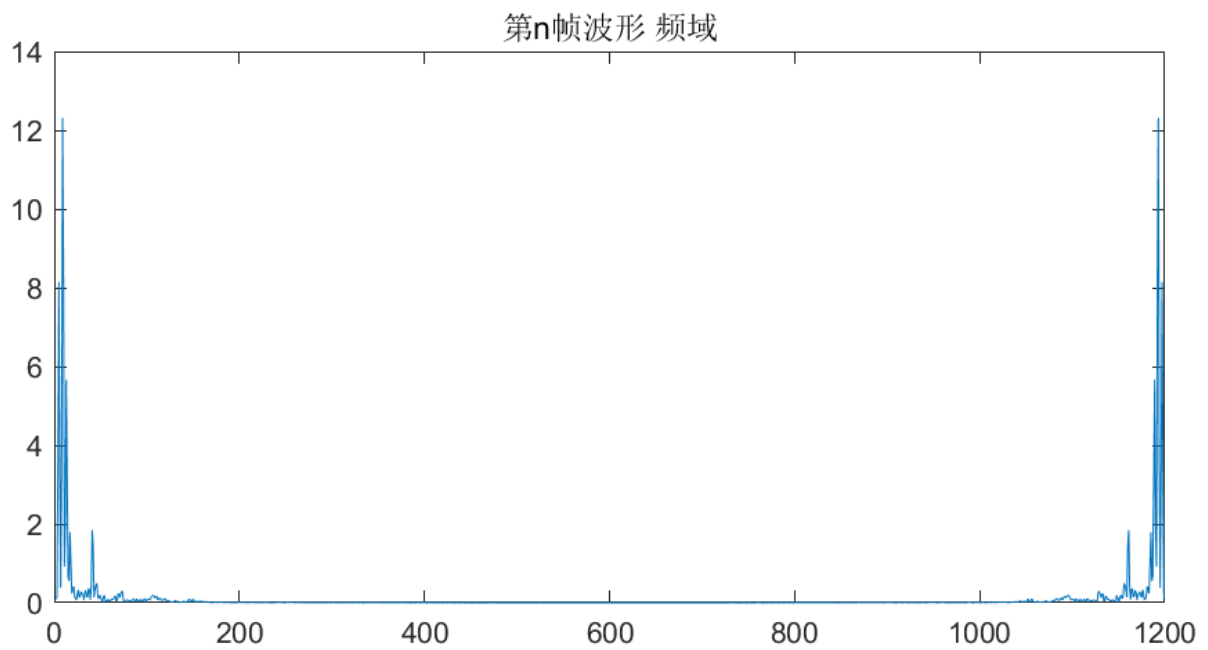
y_data = zeros(fn,wlen);     % 初始化，fn行，wlen列
indf = ((0:(fn - 1))*inc)';   % 每一帧在数据y中开始位置的指针
inds = 1:wlen;               % 每一帧的数据位置为1至wlen
indf_k = indf(:,ones(1,wlen)); % 将indf扩展乘fn*wlen的矩阵，每一列的数值都和原indf一样
inds_k = inds (ones(fn,1), : ); % 将inds扩展乘fn*wlen的矩阵，每一行的数值都和原inds一样
y_data(:) = x(indf_k + inds_k);

if (nwin >1) % 若参数中有窗函数，把每帧乘以窗函数
    w = win(:)';
    y_data = y_data.*w(ones(fn,1),:);
end
```



离散傅里叶变换

```
x_frame_freq = zeros(length(x_frame(:,1)), len_frame);  
for index = 1:length(x_frame(:,1))  
    x_frame_freq(index, :) = fft(x_frame(index, :));  
end
```

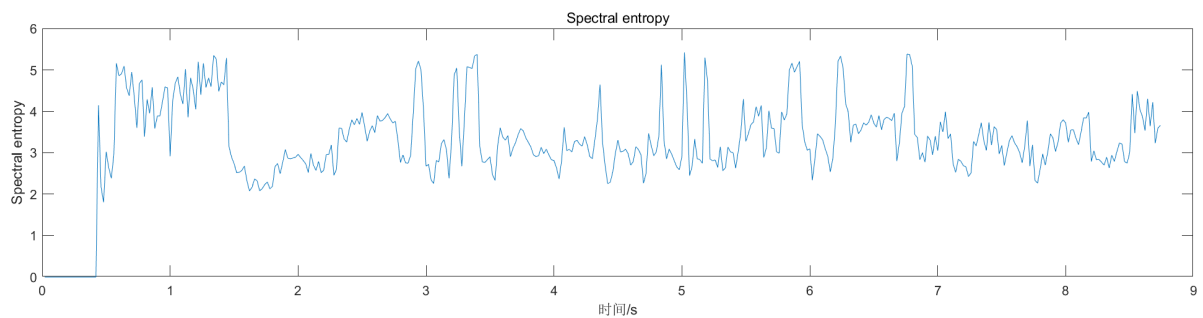


计算谱熵

```

% Spectral entropy
x_spectralEntropy = zeros(length(x_frame(:, 1)), 1);
for index = 1:length(x_frame(:, 1))
    temp = abs(x_frame_freq(index, :)); % 频谱取模
    temp = temp.^2; % 平方 能量
    if sum(temp) ~= 0
        x_pow_p = temp / sum(temp); % 计算能量分布
        for jindex = 1:length(x_pow_p) % 代入信息熵公式
            x_spectralEntropy(index) = x_spectralEntropy(index) - x_pow_p(jindex) *
log(x_pow_p(jindex));
        end
    end
end
end

```

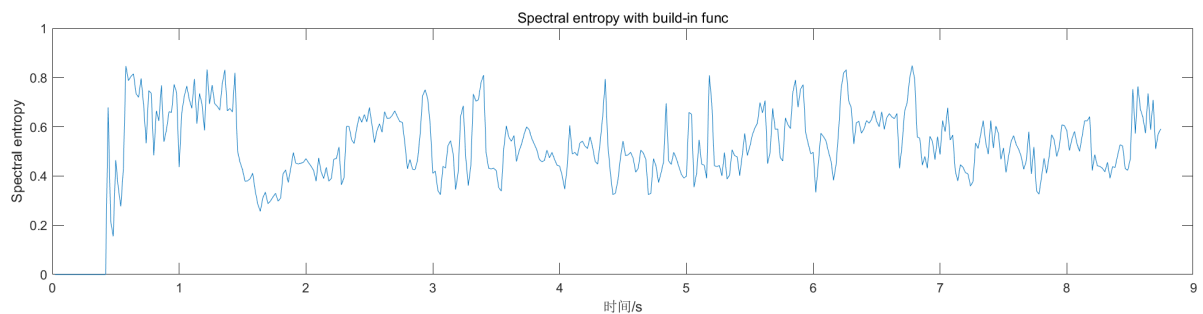


使用内建函数一步到位

```

% Spectral entropy with build-in func
x_spectralEntropy0 = spectralEntropy(x ,fs, ...
    window = win, OverlapLength = len_frame-len_inc, ...
    Range = [62.5, 5000]); % 只要62.5Hz - 5000Hz的声音

```



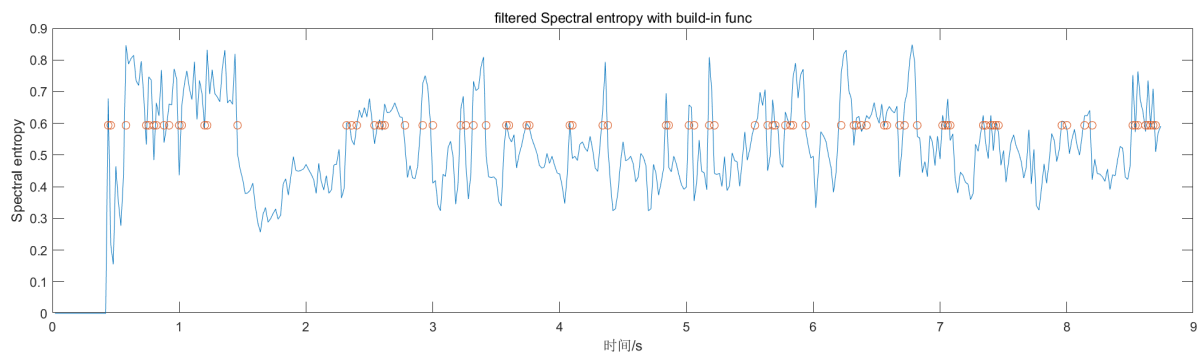
端点检测

```

gate = 0.7 * max(x_spectralEntropy2);          % 门限
x_SE_minus_gate = x_spectralEntropy2 - gate;    % 减去门限

% 计算过0的点
jindex = 1;
for index = 2:length(x_SE_minus_gate)
    if x_SE_minus_gate(index-1) * x_SE_minus_gate(index) < 0
        cross_zero(jindex) = index;
        y_cross_zero(jindex) = gate;
        jindex = jindex+1;
    end
end
x_cross_zero = cross_zero / (round(fs / len_inc));

```



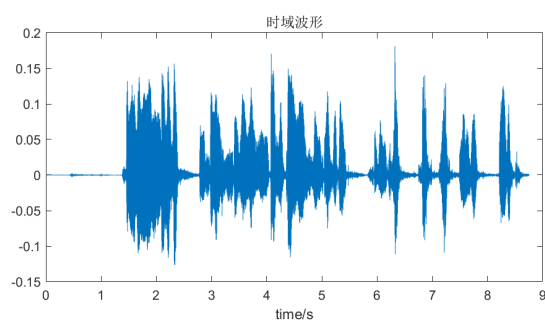
分段输出音频

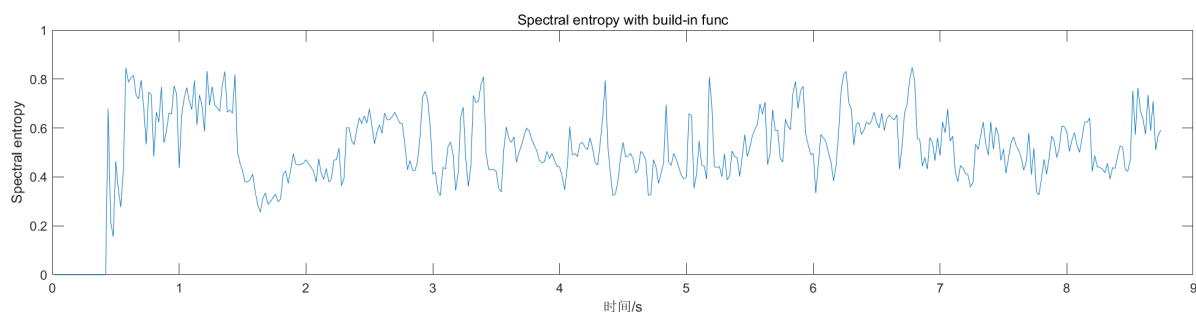
```

if false
    gate_audio = sum(x_frame_pow) / (length(x_frame_pow) * 2);
    file_nums = 1;
    for index = 2:2:length(cross_zero)-1
        temp_wave = x(cross_zero(index) * len_inc : cross_zero(index+1) *
len_inc); % 片段波形
        temp_pow = x_frame_pow(cross_zero(index):cross_zero(index+1));
        % 片段短时能量
        if (sum(temp_pow) / length(temp_pow)) > gate_audio || false
            % 判断是否是语音
            str_file_nums = num2str(file_nums);
            filename = strcat("subAudio_omega",str_file_nums,".wav");
            audiowrite(filename, temp_wave, fs);
            file_nums = file_nums + 1;
        end
    end
end
end

```

Summarize





观察时域波形和谱熵，貌似噪声的谱熵高一些。把低谱熵端输出。

示例

示例1

例句: 全民制作人们，大家
好！我是练习时长两年半的
个人练习生，蔡徐坤。喜欢
唱、跳、rap、篮球。
music!

分割：

全民制作人们大家好

ao

我
是练
习

时长
两
年半
的

个人练习
生

蔡徐
坤

喜欢

唱

跳

rap

篮球

music

示例2

例句： Ad astra
abyssosque. Welcome to
the adventurer's guild.

Ad

as

t

ra

abys

sos

que

Wel

come

to the adven

turer's

guild

示例3

例句：星と深淵に目指せ！
ようこそ冒険者協会へ。

ho shi to shin en ni me za
se you ko so bou ken
shya kyou kai e

ho

o sh

i

to

shin en ni me za s

e

e

yo

u

ko

so

bou

ken shy

a

kyou k

ai e

e

Summarize2

从汉语、日语和英语的分割情况来看，基本是按照一个音节一个音节来分割的。推测，元音的谱熵相比辅音要低一些。

到底元音辅音，清音浊音、噪音的谱熵有什么特征，等老师来解答吧。