The Biomedical Signal Project Four

Independent Component Analysis (ICA)

Zhang Bing 171848

**Purpose：**

1. Study the concept and methods of ICA in the handout and the text book

2. Explore the audio recording the processing functions of Matlab

3. Record mixed signals using one of the following methods:

a) Select at least 3 audio sources and use the equivalent number of microphones at different positions to record the multiple channels of mixed audio signals for 5 ~ 10 seconds simultaneously.

b) Select at least 3 audio sources and use one microphone to record the audio signal from each source for 5 ~ 10 seconds, then generate the equivalent number of mixed signals with different linear combinations of the recorded signals

Note: method a) is preferred, because it is more “real”.

4. Apply ICA on the mixed audio signals and see whether the original

signals can be recovered successfully

successfully

5, Plot all mixed signals and recovered signals, use PowerPoint to present

the pictures and attach the corresponding audio file to each signal

**Conculsion：**

快速ICA确实可以起到声音分离的效果，但是效果不是很理想。

尝试1：多加一个白噪音或者方波信号，在多加一个方波信号时曾经出现过分离效果比较好的结果。

尝试2：用音乐或者敲击声分别代替其中两个人说话的声音，效果会比分辨三个人说话好很多。

**Attach file：**

医学信号作业4\_张冰\_声音.pptx

**Attach code：**

%文件名称 : BS\_Project4\_zhangbing\_171848

%实现功能 : 快速ICA分离混合声音

%

%

%参考资料 : 快速ICA算法

%作者信息 : 171848-张冰

% 537405288@qq.com

% 18795969032

%修订时间 : 2018年5月17日17点21分

%调用格式 : 无

%参数释义 : 无

%项目路径

addpath(genpath(pwd));

clc;

%设置要读的data-set 路径

PATH = '';

SAMPLE = '';

disp("是否需要新的音频录音：");

iflag = input('“是”输入1，“否”输入0:\n');

if iflag == 1

t = input('请输入要录音的时间（秒）：\n');

%==========读取信号信息

for i = 1:3

myVoice(i) = audiorecorder(44100,16,1);

disp('开始录音');

recordblocking(myVoice(i), t);

disp('录音结束请按“回车”开始下一段录音');

x = input('');

end

disp("录音数据获取");

myVoice1 = getaudiodata(myVoice(1));

myVoice2 = getaudiodata(myVoice(2));

myVoice3 = getaudiodata(myVoice(3));

disp("录音数据写入文件");

audiowrite('myVoice1.wav',myVoice1,44100);

audiowrite('myVoice2.wav',myVoice2,44100);

audiowrite('myVoice3.wav',myVoice3,44100);

elseif iflag == 0

disp("直接读取文件数据");

[myVoice1,~] = audioread('myVoice1.wav');

[myVoice2,~] = audioread('myVoice2.wav');

[myVoice3,~] = audioread('myVoice3.wav');

else

disp("Wrong Number");

end

%==========画单独声音图==========

figure('NumberTitle', 'off', 'Name', '原声');

clf, box on, hold on;

s(1) = subplot(4,1,1);

s(2) = subplot(4,1,2);

s(3) = subplot(4,1,3);

s(4) = subplot(4,1,4);

plot(s(1),myVoice1);

plot(s(2),myVoice2);

plot(s(3),myVoice3);

%叠加声音

% t=1:10\*44100;

% whiteSgn=square(100\*t,50)/20;

% whiteSgn = rand(1,t\*44100)/20;

%simpleMix = horzcat(myVoice1,myVoice2,myVoice3,whiteSgn');

simpleMix = horzcat(myVoice1,myVoice2,myVoice3);

plot(s(4),simpleMix);

disp(size(simpleMix));

audiowrite('simpleMix.wav',simpleMix,44100);

%对叠加声音进行非线性混合

A = rand(3,3);

%A = rand(4,4);

myVoiceMix= simpleMix \* A;

myVoiceMix1 = myVoiceMix(:,1);

myVoiceMix2 = myVoiceMix(:,2);

myVoiceMix3 = myVoiceMix(:,3);

%myVoiceMix4 = myVoiceMix(:,4);

audiowrite('myVoiceMix1.wav',myVoiceMix1,44100);

audiowrite('myVoiceMix2.wav',myVoiceMix2,44100);

audiowrite('myVoiceMix3.wav',myVoiceMix3,44100);

%audiowrite('myVoiceMix4.wav',myVoiceMix4,44100);

%==========画模拟话筒混合后每个声音图==========

figure('NumberTitle', 'off', 'Name', '混合后声音');

clf, box on, hold on;

s(1) = subplot(4,1,1);

s(2) = subplot(4,1,2);

s(3) = subplot(4,1,3);

s(4) = subplot(4,1,4);

plot(s(1),myVoiceMix1);

plot(s(2),myVoiceMix2);

plot(s(3),myVoiceMix3);

%plot(s(4),myVoiceMix4);

%==========利用FastICA进行解混==========

X = myVoiceMix';

[M, T] = size(X);

%矩阵平均值=0

X = X - mean(X')' \* ones(1);

% average = mean(X')';

% for i = 1:M

% X(i, :) = X(i, :) - average(i) \* ones(1,T);

% end

%白化矩阵

Cx = cov(X',1);

[eigvector,eigvalue] = eig(Cx);

W = eigvalue^(-1/2)\*eigvector;

Z = W \* X;

Maxcount = 100000;

Critical = 0.00001;

m = M;

W = rand(m);

for n = 1:m

WP = W(:, n);

count = 0;

LastWP = zeros(m,1);

W(:,n) = W(:,n)/norm(W(:,n));

while abs(WP - LastWP)&abs(WP + LastWP) > Critical

count = count + 1;

LastWP = WP;

%计算非高斯性

for i = 1:m

WP(i) = mean(Z(i,:).\*(tanh((LastWP)'\*Z)))-(mean(1-(tanh((LastWP))'\*Z).^2)).\*LastWP(i);

end

WPP = zeros(m,1);

for j = 1:n-1

WPP = WPP + (WP'\*W(:,j))\*W(:,j);

end

%迭代

WP = WP - WPP;

WP = WP/(norm(WP));

if count == Maxcount

disp("未找到信号");

end

end

W(:,n) = WP;

end

Z = (W'\* Z)';

disp("ICA结束将数据存入文件");

audiowrite('myVoiceSep1.wav',Z(:,1),44100);

audiowrite('myVoiceSep2.wav',Z(:,2),44100);

audiowrite('myVoiceSep3.wav',Z(:,3),44100);

%audiowrite('myVoiceSep4.wav',Z(:,4),44100);

%==========画解混后图==========

figure('NumberTitle', 'off', 'Name', '解混后声音波形');

clf, box on, hold on;

s(1) = subplot(4,1,1);

s(2) = subplot(4,1,2);

s(3) = subplot(4,1,3);

%s(4) = subplot(4,1,4);

plot(s(1),Z(:,1));

plot(s(2),Z(:,2));

plot(s(3),Z(:,3));

%plot(s(4),Z(:,4));