

به نام خدا



تمرین سری سوم درس BCI

محمد مهدی حبیبی

شماره دانشجویی: 401617033

استاد: دکتر شالچیان

پاییز- زمستان 1402

فهرست مطالب

4	سوال یک
8	سوال سوم برای ارزیابی CCA
8	نتیجه بدست آمده با استفاده از مدل CCA
9	سوال دوم
10	ترکیب CCA با ML
10	مدل KNN
11	سوال سوم برای ترکیب CCA با KNN
12	مدل SVM
13	سوال سوم برای ترکیب CCA با SVM

تمرین ۴

- در این تمرین مراحل پیش پردازش، استخراج ویژگی و طبقه بندی سیگنال‌ها مورد ارزیابی قرار می‌گیرد. (مهلت ۱۰ بهمن)
- لطفاً با توجه به توضیحات زیر، فایل اسکریپت نوشته شده (m.) و گزارش (pdf) خود را در زمان تعیین شده در سامانه LMS ثبت بفرمایید.
- گزارش تمرین باید شامل تصاویر خروجی و توضیحات مربوط به آن و توضیحات مربوط به اسکریپت و فرآیند طی شده در برنامه باشد. (نیازی به بارگذاری دیتاست در فایل زیپ گزارش نیست.)
- تمام شبیه‌سازی‌های زیر را روی دیتاست SSVEP_DATA (پتانسیل برانگیخته بینایی) انجام دهید.
- این مجموعه داده حاوی سیگنال‌های EEG برای ۳ فرکانس به ترتیب ۱۳، ۲۱ و ۱۷ هرتز است. (به ترتیب دقت نمایید)
- فرکانس نمونه برداری ۲۵۶ هرتز است.
- استفاده از ۲ هارمونیک کافی است.
- می‌توانید صرفاً از کانال‌های اول و هشتم استفاده کنید.
- بازه زمانی سیگنال ۵ ثانیه است.

۱- سیگنال‌های سینوسی و کسینوسی را تا هارمونیک دوم ایجاد کرده و با استفاده از تابع myCCA، مسئله طبقه بندی را به صورت مستقیم حل نمایید.

۲- مرحله قبل را تکرار کنید؛ با این تفاوت که از خروجی‌های تابع myCCA، بعنوان ویژگی استفاده کنید و این بار با استفاده از بردار ویژگی، مسئله طبقه بندی را برای مثال با کلاسیفایر SVM حل نمایید.

۳- برای ارزیابی، از روش k-fold cross validation استفاده کنید و برای هر دو رویکرد فوق، ماتریس درهم ریختگی و میانگین صحت عملکرد مدل را گزارش کنید.

سوال یک

۱- سیگنال‌های سینوسی و کسینوسی را تا هارمونیک دوم ایجاد کرده و با استفاده از تابع myCCA، مسئله طبقه بندی را به صورت مستقیم حل نمایید.

اول دیتاست مورد نظر که مربوط به ssvep می باشد رو load میکنیم :

```
1 clear;
2 clc;
3 close all
4
5 load('SSVEP_Data.mat');
6
7
8 fstimulus(1)= 13; % we know frequency of stimulus 1 is 13 hz
9 fstimulus(2)= 21; % we know frequency of stimulus 1 is 13 hz
10 fstimulus(3)= 17; % we know frequency of stimulus 1 is 13 hz
11
12 fs=256;% sampling rate is 256 hz
```

با توضیحاتی که از دیتاست داریم، می دانیم سوژه هر سری به یکی از محرک های 13 و 21 و 17 هرتز نگاه کرده هست در نتیجه یک مسئله ی سه کلاسه داریم که کلاس یک برای محرک 13 هرتز، کلاس دوم برای محرک 21 هرتز و در نهایت کلاس سوم برای محرک 17 هرتز می باشد، همچنین فرکانس نمونه برداری نیز برابر 256 هرتز می باشد.

```
%% creating nice structure for dataset ssvep
data= cat(3,data1,data2,data3);
label= [ones(1,size(data1,3)), 2*ones(1,size(data2,3)),3*ones(1,size(data3,3))];
```

data1	1280x8x160 dou...
data2	1280x8x160 dou...
data3	1280x8x160 dou...
fs	256
fstimulus	[13,21,17]

```

slecHarmonic = 2;% first and second harmonic
channel_number = [1,8];

time=5; % 5 seconds tool keshideh baray recording
TimeAlltrails=time * 256;
t= linspace(0,time,TimeAlltrails);
Y1=[]; % first refrence
Y2=[]; % second refrence
Y3=[]; % third refrence
for n=1:slecHarmonic

    tp1(1,:)= sin(2*pi*(n*fstimulus(1))*t);% reference signal for fstimulus(1) is 13 hz
    tp1(2,:)= cos(2*pi*(n*fstimulus(1))*t);% reference signal for fstimulus(1) is 13 hz
    Y1=[Y1;tp1];

    tp2(1,:)= sin(2*pi*(n*fstimulus(2))*t);% reference signal for fstimulus(2) is 21 hz
    tp2(2,:)= cos(2*pi*(n*fstimulus(2))*t);% reference signal for fstimulus(2) is 21 hz
    Y2=[Y2;tp2];

    tp3(1,:)= sin(2*pi*(n*fstimulus(3))*t);% reference signal for fstimulus(3) is 17 hz
    tp3(2,:)= cos(2*pi*(n*fstimulus(3))*t);% reference signal for fstimulus(3) is 17 hz
    Y3=[Y3;tp3];

end

```

سیگنال های
رفرنس

y_1
 y_2
 y_3

از اونجایی که دیتاست گفته سوژه به سه چراغ با فرکانس های 13 و 17 و 21 هرتز نگاه میکند بنابراین برای اینکه تشخیص بدیم در هر ترایال به کدام چراغ نگاه کرده کافیه سه رفرنس بنام های Y_1, Y_2, Y_3 بسازیم که در واقع سه سیگنال فرکانسی بصورت $\sin(2\pi f t), \cos(2\pi f t)$ می سازیم که هر بار f را برای Y_1 برابر با 13 هرتز و برای Y_2 برابر با 21 هرتز و برای Y_3 برابر با 17 هرتز قرار می دهیم.

همین طور چون گفته 5 ثانیه نگاه می کنیم t را با استفاده از `linspace` بدست بیاریم، به این صورت که طول کل ثبت مون برابر با 5 ثانیه می باشد و فرکانس نمونه برداری نیز 256 هرتز می باشد در نتیجه می تونیم با ضرب این دو، کل نمونه ها را بدست بیاریم

$$\text{duration} = 5 \text{ s}$$

$$\text{length} = 5 \text{ s} \times 256 \text{ Hz}$$

$$t = \text{linspace}(0, 5, \text{length})$$

سیگنال را از یک فیلتر ناچ عبور می‌دهیم تا نویز برق شهر را حذف کنیم

```
[b,a]= butter(3,[49.5 50.5]/(fs/2), 'stop'); % design notch filter for removing noise 50 hz
```

هکس ترایال دایم

جدا کردن یک ترایال

و قرار دادن آن به

Signal

نفس notch را به signal اعمال می‌کنیم

cca را به ازای هر فاکتور می‌کنیم

محاسبه می‌کنیم

```
for i=1:480
    Signal= data(:,i);
    Signal= filtfilt(b,a,Signal);

    % using CCA Function
    r1 = myCCA(Signal(:,channel_number),Y1');
    r2 = myCCA(Signal(:,channel_number),Y2');
    r3 = myCCA(Signal(:,channel_number),Y3');

    pearson_coefficient1 = max(r1);
    pearson_coefficient2 = max(r2);
    pearson_coefficient3 = max(r3);

    Pearson_Coefficient = [pearson_coefficient1 , pearson_coefficient2 , pearson_coefficient3];
    [MaxValue , index] = max(Pearson_Coefficient) ;

    numberStimulus(i) = index; % result : showing subject have attention which stimulus

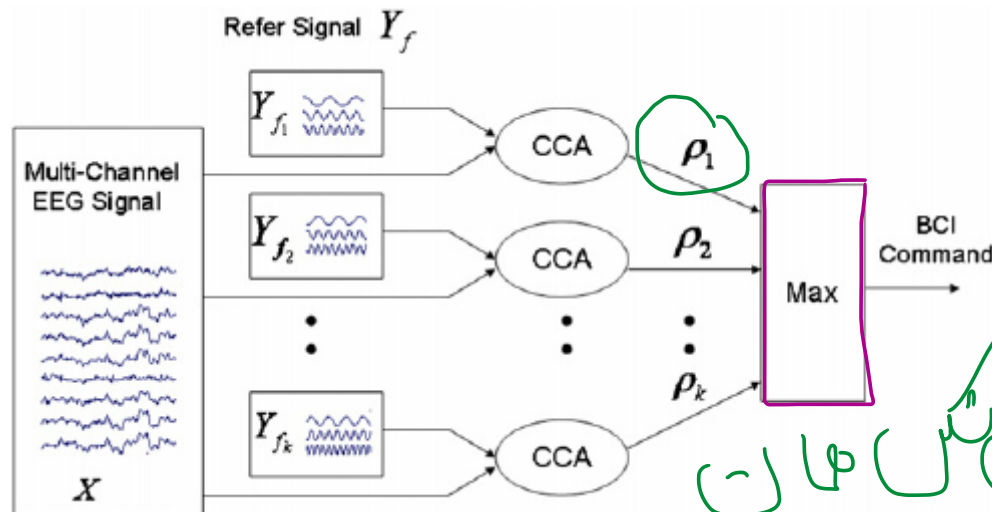
end
```

حالا در ادامه وقتی که سیگنال‌های رفرنس را تولید کردیم از تابع آماده myCCA استفاده می‌کنیم، تا correlation بین ترایال‌های بدست آمده از ثبت EEG را با رفرنس‌ها یعنی Y1,Y2,Y3 بدست بیاوریم. در نتیجه ما r1,r2,r3 خواهیم داشت

```
r1      [0.0703;0.0325]
r2      [0.0386;0.0225]
r3      [0.3089;0.0898]
```

حالا از هر کدام از این ماتریس‌های r1,r2,r3 باید بیشترین شون رو بدست بیاوریم و در نهایت نیز یک max نهایی رو اعمال می‌کنیم تا هر کدام از این ضریب‌ها بیشترین همبستگی را داشته باشد در نتیجه پی می‌بریم اون سیگنال مورد نظرمون به کدام محرک بیشترین همبستگی رو داره

```
pearson_coeff... 0.0703
pearson_coeff... 0.0386
pearson_coeff... 0.3089
```



```

% Pearson_Coefficient = 0.0000
r1 = [0.0703;0.0325]
r2 = [0.0386;0.0225]
r3 = [0.3089;0.0898]

```

از اونجایی که در مسئله مون گفته شده تنها از کانال های 1 و 8 استفاده کنیم در نتیجه وقتی r_1, r_2, r_3 را بدست می آوریم یک ماتریسی می باشد که دو تا ستون دارد که به ازای هر کانال یک ستون بدست آوردیم.

% using CCA Function

```

r1 = myCCA(Signal(:,channel_number),Y1');
r2 = myCCA(Signal(:,channel_number),Y2');
r3 = myCCA(Signal(:,channel_number),Y3');

```

با استفاده از تابع **myCCA** که در تمرین داده شده می توانیم **correlation** بین سیگنال های رفرنس Y_1, Y_2, Y_3 را با سیگنال EEG مون بدست بیاریم

```

pearson_coefficient1 = max(r1);
pearson_coefficient2 = max(r2);
pearson_coefficient3 = max(r3);

```

```

Pearson_Coefficient = [pearson_coefficient1 , pearson_coefficient2 , pearson_coefficient3];
[MaxValue , index] = max(Pearson_Coefficient) ;

```

```

numberStimulus(i) = index; % result : showing subject have attention which stimulates

```

از بین فیچر هایی که با استفاده از **CCA** بدست آوردیم و برابر با r_1, r_2, r_3 قرار دادیم حالا دنبال بیشترین **correlation** هستیم بنابراین کافیه از تمامی این فیچر ها بیشترین مقدارشون رو بدست بیاریم

سوال سوم برای ارزیابی CCA

نتیجه بدست آمده با استفاده از مدل CCA

```
%% ACCuracy , performance
MatrixConfusion= confusionmat(label,numberStimulus);
m = diag(MatrixConfusion);
n = sum(MatrixConfusion(:));
Total_Acc= sum(m) /n * 100;
disp(['Total accuracy using CCA: ', num2str(Total_Acc), '%'])
```

تمام مولفه های ماتریس ابهام را بدست می آوریم و باهم جمع میکنیم و در داخل n قرار میدهم

ماتریس ابهام بصورت روبرو می باشد :

MatrixConfusion =

158	0	2
22	120	18
5	0	155

قطر اصلی ماتریس ابهام را بدست می آوریم

1. از کلاس اول، 158 تریال
2. از کلاس دوم تنها 120 ترا

3. و در نهایت از کلاس سوم، 155 تریال را به درستی تشخیص داده است.

صحت بدست آمده با استفاده از CCA برابر با 90.2083% میباشد:

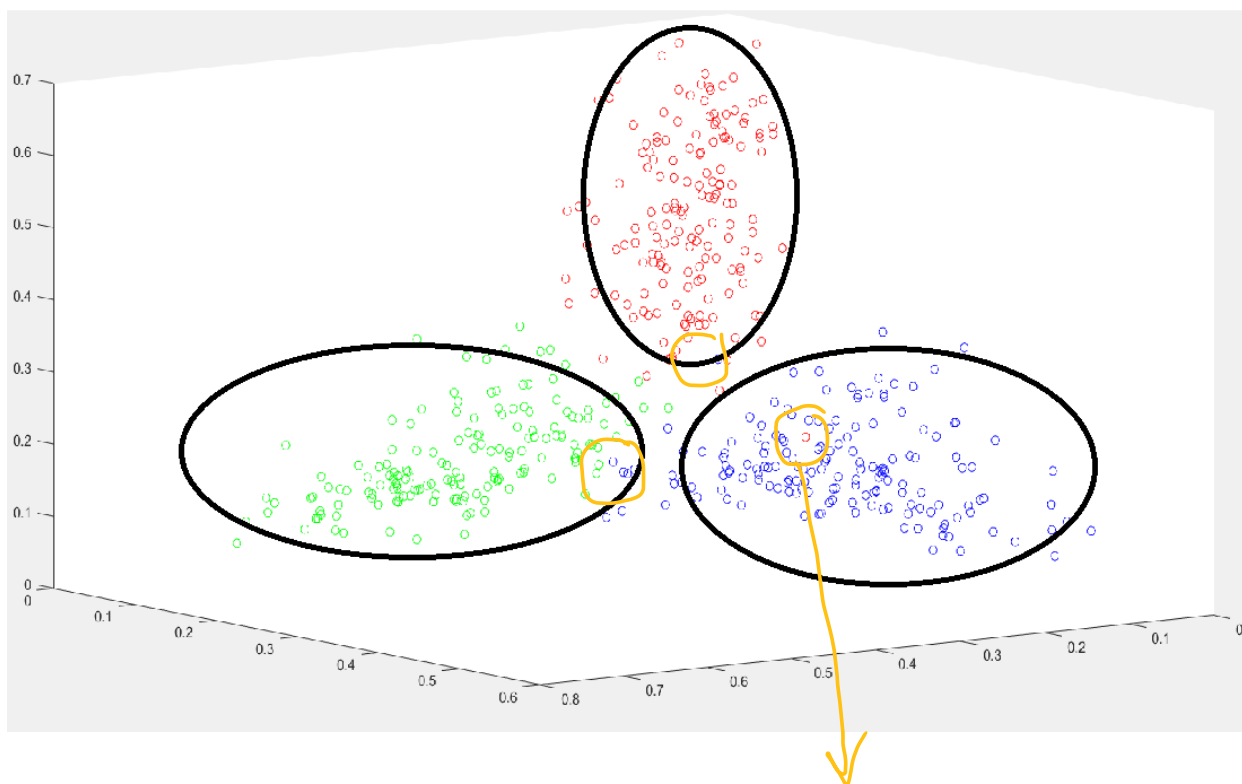
% Total accuracy using CCA: 90.2083

سوال دوم

۲- مرحله قبل را تکرار کنید؛ با این تفاوت که از خروجی‌های تابع myCCA، بعنوان ویژگی استفاده کنید و این بار با استفاده از بردار ویژگی، مسئله طبقه بندی را برای مثال با کلاسیفایر SVM حل نمایید.

از اونجایی که ما یک مسئله ی سه کلاسه داریم و در واقع 160 تریال برای هر کدوم از کلاس ها داریم ،می تونیم نمایش و توزیع دیتا را ببینیم که بصورت زیر می باشد

```
%plot3(features(1,:),features(2,:),features(3,:), '0g')
plot3(features(1,label ==1),features(2,label == 1),features(3,label == 1),'0g');
hold on
plot3(features(1,label ==2),features(2,label == 2),features(3,label == 2),'0b');
hold on
plot3(features(1,label ==3),features(2,label == 3),features(3,label == 3),'0r')|
```



همین طور که از شکل بالا قابل مشاهده می باشد مثلا میتوان ببینیم که یکی از تریال های قرمز داخل تریال های کلاس ابی قرار گرفته است یا تریال کلاس ابی در کلاس سبز قرار گرفته ،بنابر این می توانیم بفهمیم کافیه با یک مدل یادگیری ماشین ساده این سه کلاس را از هم تفکیک کنیم و به دقت خوبی هم می توانیم دست پیدا کنیم چراکه تریال های خیلی کمی از هر کلاس در بقیه کلاس ها هستند که در ادامه می تونیم با استفاده از ماتریس ابهام تعداد این تریال هایی که به اشتباه در کلاس دیگر در نظر گرفته می شوند را بدست بیاوریم

ترکیب CCA با ML

مدل KNN

ویژگی هایی که با استفاده از روش CCA استخراج کردیم را به مدل KNN می دهیم

مدل knn را با 5 همسایگی فیت کردیم

```
% fit knn model
model= fitcknn(traindata',trainlabel,'NumNeighbors',5); % number of neighbors is 5

output= predict(model,testdata)';
```

صحت بدست آمده با استفاده از مدل KNN و با استفاده از 5 fold cross validation بصورت زیر می باشد :

Total Accuracy using KNN(k=5) : 97.7083%

Accuracy for class 1: 100%

Accuracy for class 2: 94.375%

Accuracy for class 3: 98.75%

ماتریس ابهام با استفاده از مدل KNN بصورت روبرو می باشد:

1. از این ماتریس می توانیم این نکته را بفهمیم که مدل KNN مون، 160 تا تریال مربوط به کلاس اول را به درستی کامل تشخیص داده

2. برای کلاس دوم، 151 تریال را درست در کلاس دوم تشخیص داده و 9 تا از تریال های آن را در کلاس اول به اشتباه در نظر گرفته

3. برای کلاس سوم نیز 158 تریال که مربوط به کلاس سوم هستند را به درستی برای کلاس سوم در نظر گرفته و تنها یک تریال را در کلاس اول و یک تریال را در کلاس دوم به اشتباه قرار داده است

Ct =

160	0	0
9	151	0
1	1	158

کلاس 1
کلاس 2
کلاس 3

سوال سوم برای ترکیب CCA با KNN

صحت بدست آمده با استفاده از KNN را میتوانیم بصورت زیر بدست بیاوریم :

```
ConfusionMatrix= confusionmat(testlabel,output);
Ctotal= Ctotal+ ConfusionMatrix;
accuracy(i)= sum(diag( ConfusionMatrix)) / sum( ConfusionMatrix(:))*100;
accuracy1(i)= ConfusionMatrix(1,1) / sum( ConfusionMatrix(1,:))*100;
accuracy2(i)= ConfusionMatrix(2,2) / sum( ConfusionMatrix(2,:))*100;
accuracy3(i)= ConfusionMatrix(3,3) / sum( ConfusionMatrix(3,:))*100;
```

تمام عناصر ماتریس را با هم جمع می‌کنیم

diag ماتریس ابهام را بدست می‌آوریم

160	0	0
9	151	0
1	1	158

صحت کل بدست آمده از مدل KNN(k=5) با kfold=5 برابر با 99.375% میباشد.

Total Accuracy using KNN(k=5): 99.375%

Accuracy for class 1: 100%

Accuracy for class 2: 98.75%

Accuracy for class 3: 99.375%

بنابراین صحت کل یا Total

Accuracy به این صورت می باشد که نسبت

جمع مقادیر قطر اصلی به تمام مولفه های

ماتریس ابهام را بدست می آوریم و در 100

ضرب میکنیم که برابر با 99.375% می باشد.

برای کلاس اول نیز همین طور که از ماتریس ابهام بالا نیز قابل مشاهده می باشد برابر با 100% می باشد .

برای کلاس دوم نیز 98.75% و برای کلاس سوم 99.375% میباشد.

در این مسئله در واقع یک مسئله ی سه کلاسه می باشد نیاز داریم SVM را کمی تغییر بدیم بنابراین می توانیم از تکنیک یکی در مقابل همه یا تکنیک دوم که یکی در برابر یکی One vs One استفاده کنیم. که در این تمرین از تکنیک دوم استفاده کردیم و بصورت خلاصه تابعی که تعریف کردیم برای این کار را توضیح میدهیم

```
function [model] = multisvmtrain0v0(Xtrain,Ytrain,kernel)
labelClass= unique(Ytrain);
y1=labelClass(1);
y2=labelClass(2);
y3=labelClass(3);

%% class 1 vs class 2
data1= Xtrain(:,Ytrain==labelClass(1));
data2= Xtrain(:,Ytrain==labelClass(2));

traindata=[data1,data2]; %concat

trainlabel= [y1*ones(1,size(data1,2)),y2*ones(1,size(data2,2))];

model.svm1 = fitcsvm(traindata',trainlabel,'KernelFunction',kernel,'Standardize',1);
```

یک تابع با نام multisvmtrain0v0 تعریف کردیم و کد مربوط به کلاس یک در برابر کلاس دو را نیز آوردیم (ما بقی نیز به همین صورت می باشد) و از اونجایی که با یک مسئله ی 3 کلاسه روبرو هستیم تابع را برای مسئله ی سه کلاسه مون تعریف کردیم به این صورت که هر کدام از لیبل های کلاس ها را در y مربوط به آن کلاس قرار دادیم و وقتی دوتا دوتا داریم کلاس ها را بررسی میکنیم در نتیجه مسئله مون به نحوی به یک مسئله ی دو کلاسه تبدیل میشود .

```
function [output] = multiclassSVM0v0(model,Xtest)
userlabel= model.userlabel;

%%
svm1=model.svm1;
output1 = predict(svm1,Xtest)';

svm2=model.svm2;
output2 = predict(svm2,Xtest)';

svm3=model.svm3;
output3 = predict(svm3,Xtest)';
```

از این تابع روبرو نیز بعد از تابع اول استفاده میکنیم تا مدل SVM را روی دیتای تست مون فیت کنیم

```
k = 5 ;%cross validation when set k_fold is 5
```

همچنان ارزیابی مون را با kfold cross validation برابر با 5 انجام می دهیم و در ادامه از دوتا function ای که در صفحه ی قبل توضیح دادیم ،استفاده میکنیم تا بتوانیم مدل SVM one VS one را روی این دیتاست پیاده سازی کنیم

```
SVMMModel =multisvmtrain0v0(traindata,trainlabel,'rbf');  
output=multiclassSVM0v0(SVMMModel,testdata);
```

کرنل را همانطور که از کد بالا نیز قابل مشاهده می باشد برابر با rbf قرار میدهم و یک object بنام SVMMModel می سازیم با استفاده از function ای که تعریف کردیم بعدش برای پیش بینی خروجی نیز از function دومی که تعریف کردیم استفاده میکنیم و output را به ما میدهد.

سوال سوم برای ترکیب CCA با SVM

```
Ctotal =
```

```
160      0      0  
    7    153      0  
    1      1    158
```

نتایج بدست آمده با استفاده از مدل SVM one VS one

صحت کل بدست آمده از مدل SVM با kfold=5

برابر با 98.125% میباشد.

```
Total Acc using SVM is: 98.125%
```

```
Acc for class 1 is: 100%
```

```
Acc for class 2 is: 95.625%
```

```
Acc for class 3 is: 98.75%
```

$$\frac{153}{160} \times 100 = 95.625\%$$

$$\frac{158}{160} \times 100 = 98.75\%$$

1. در SVM نیز مانند مدل KNN که بررسی کردیم، کلاس یک را کاملاً به درستی تشخیص میدهد در نتیجه صحت برای کلاس 1 برابر با 100% می باشد.

2. برای کلاس دوم نیز 153 تریال را از 160 تریال به درستی تشخیص میدهد و صحت کلاس دوم نیز برابر با 95.625% میباشد.

3. برای کلاس سوم نیز 158 تریال را به خوبی تشخیص داده و دقت 98.75% را بدست آوردیم .