

به نام خدا



Blind Source Separation (BSS)

تکلیف شماره

3

محمد رضا آرانی

810100511

دانشگاه تهران

1402/01/15

جدول محتویات

بخش اول:	3
الف.....	5
ب.....	8
ج.....	9
د.....	11
بخش دوم:	14

بخش اول:

میخواهیم یک مسئله طبقه بندی دو کلاسه را با استفاده از رویکرد CSP بر روی داده های EEG انجام دهیم. کلاس اول مربوط به تصور حرکت پا و کلاس دوم مربوط به تصور انجام یک سری عملیات تفریق ذهنی است. سه ماتریس سه بعدی و یک بردار در فایل hw3-1.mat به شرح زیر در اختیار شما قرار داده شده است:

داده های آموزش مربوط به کلاس اول در یک ماتریس سه بعدی با نام TrainData_class1 به ابعاد $256 \times 60 \times 30$ (تعداد کانالها 30، تعداد نمونه های زمانی 256 و تعداد آزمایشها 60) قرار داده شده است. برای همه ی آزمایش ها، میانگین هر کانال را صفر کنید و در نهایت داده های جدید را در TrainData_class1 ذخیره کنید.

داده های آموزش مربوط به کلاس دوم در یک ماتریس سه بعدی با نام TrainData_class2 به ابعاد $256 \times 60 \times 30$ (تعداد کانالها 30، تعداد نمونه های زمانی 256 و تعداد آزمایشها 60) قرار داده شده است. برای همه ی آزمایش ها، میانگین هر کانال را صفر کنید و در نهایت داده های جدید را در TrainData_class2 ذخیره کنید.

داده های آزمون در یک ماتریس سه بعدی با نام TestData به ابعاد $256 \times 40 \times 30$ (تعداد کانالها 30، تعداد نمونه های زمانی 256 و تعداد آزمایشها 40) قرار داده شده است. برای همه ی آزمایش ها، میانگین هر کانال را صفر کنید و در نهایت داده های جدید را در TestData ذخیره کنید.

کلاس داده های آزمون در یک بردار با نام TestLabel به ابعاد 1×40 قرار داده شده است.

در ابتدا اقدام به صفر کردن مقدار میانگین داده ها برای هر آزمایش در هر کانال می کنیم:

```
1. TrainData_class1 = TrainData_class1 - repmat( mean(TrainData_class1, 2), 1, Num_of_Cols) ;
2. TrainData_class2 = TrainData_class2 - repmat( mean(TrainData_class2, 2), 1, Num_of_Cols) ;
```

در این راستا، ابتدا میانگین داده ها را حساب کرده که یک ماتریس $60 \times 1 \times 30$ می شود و سپس برای تطبیق با ماتریس اولیه مقدار میانگین را به ازای تعداد داده های هر کانال در هر آزمایش کنار هم چیده و سپس از ماتریس اولیه کم می کنیم. به این ترتیب از همه ی داده ها مقدار میانگین کم شده و با محاسبه ی دوباره ی میانگین با مقادیر نزدیک به 0 به دلیل دقت محاسباتی مواجه می شویم.

- برای محاسبه‌ی فیلترهای مکانی مورد نیاز برای این سوال، از مفهوم *CSP* یا *Common Spatial Pattern* استفاده می‌کنیم که از آمارگان مرتبه دوم داده‌ها برای طراحی یک کلاس‌بند خطی استفاده می‌کند.
- مسئله پیدا کردن یک نگاشت خطی است که داده‌ها به فضایی ببرد که در آن فضا داده‌های دو کلاس نسبت به هم بیشینه پراکندگی رو داشته باشند. برای این کار تابع هدف زیر را در نظر می‌گیریم:

$$f(w) = \frac{\sigma_{class-1}^2}{\sigma_{class-2}^2} = \frac{w^T (X_{class-1} X_{class-1}^T) w}{w^T (X_{class-2} X_{class-2}^T) w} = \frac{w^T R_{class-1} w}{w^T R_{class-2} w}$$

که در آن، برای هر کلاس داده یک ماتریس کواریانس تشکیل می‌دهیم و به دلیل تعدد آزمایش‌ها، آن‌ها را به ازای تمام آزمایش‌ها جمع کرده و میانگن‌گیری می‌کنیم:

```

1. 1. for i=1:Num_of_Pages
2. 2. R_class_1{i,1} = TrainData_class1(:,i)*TrainData_class1(:,i)';
3. 3. R_class_2{i,1} = TrainData_class2(:,i)*TrainData_class2(:,i)';
4. 4.
5. 5. R_class_1_Sum = R_class_1{i,1} + R_class_1_Sum;
6. 6. R_class_2_Sum = R_class_2{i,1} + R_class_2_Sum;
7. 7.
8. 8. end
9. 9.
10. 10. % Calculate the Mean of every Page:
11. 11. R_class_1_Mean = R_class_1_Sum/Num_of_Pages;
12. 12. R_class_2_Mean = R_class_2_Sum/Num_of_Pages;
13.

```

الف)

الف) با استفاده از داده های آموزش، فیلترهای مکانی CSP (W_{csp}) را به دست آورده و بر روی داده های آموزش اعمال کنید. حتما ستون های W_{csp} را نرمالیزه کنید. سیگنال فیلتر شده متناظر با فیلتر اول ($W_{csp}(:,1)$) و آخر ($W_{csp}(:,30)$) را برای آزمایش ۴۹م کلاس اول و همچنین برای آزمایش ۴۹م کلاس دوم رسم کنید (مشابه شکلی که در کلاس کشیده شد). همان طور که مشاهده می کنید برای هر آزمایش، خروجی یک فیلتر پراکندگی زیادی دارد و خروجی فیلتر دیگر پراکندگی کمی دارد. به صورت کمی هم پراکندگی ها (واریانس ها) را (فقط برای این دو آزمایش!) گزارش کنید.

پس از به دست آمدن ماتریس کواریانس مورد نظر، باید ابتدا GEVD را اعمال کرده تا بردارهای ویژه و مقادیر ویژه به دست بیاید و سپس اقدام به نرمالیزه کردن ستون های ماتریس W به دست آمده بکنیم:

```
1. % Use GEVD (Generalized Eigen-Value Decomposition) over R_class_1_Mean on
2. % top and R_class_2_Mean in denominator:
3.
4. [W,Lambda] = eig(R_class_1_Mean,R_class_2_Mean) ;
5.
6. % Maximizing the Rayleigh Ratio requires the biggest Lambda and its
7. % corresponding Eigen Vector to be chosen!
8.
9. % Normalizing W Columns:
10. Norm_Matrix_of_Cols_W = W'*W;
11. for j=1:Num_of_Rows
12.     W(:,j) = W(:,j)/(Norm_Matrix_of_Cols_W(j,j));
13. end
14.
```

در واقع مقادیر روی قطر $W^T W$ همان میزان نرم-2 ستون های ماتریس W است.

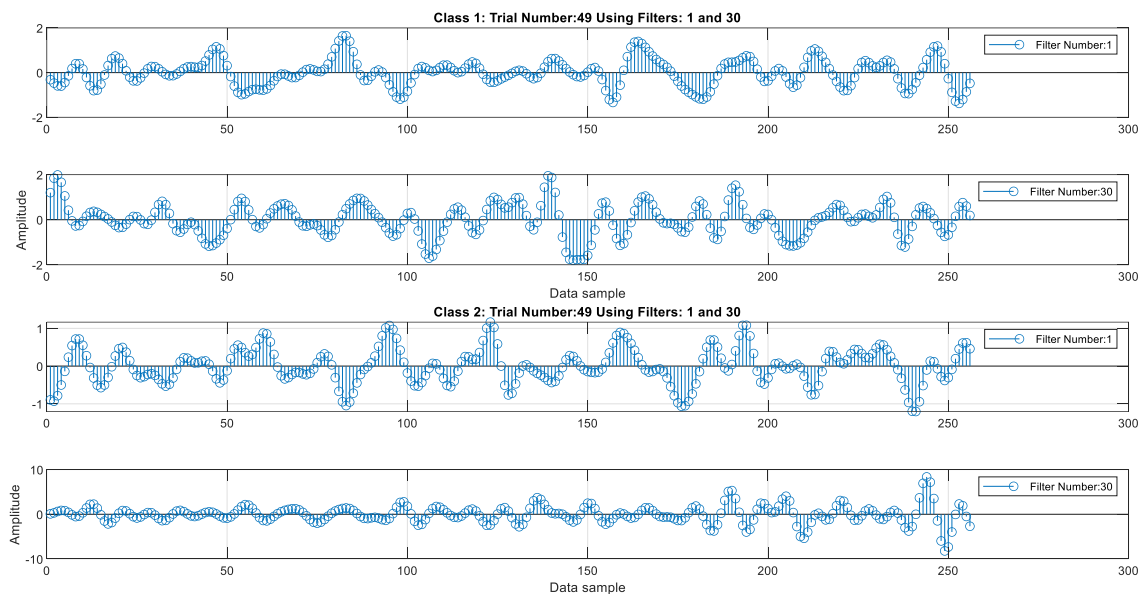
پس از نرمالایز کردن ستون های ماتریس W وقت اعمال تبدیل بر روی داده های داده شده است:

```
1. Z_Class1 = tensorprod(W',TrainData_class1,2,1);
2. Z_Class2 = tensorprod(W',TrainData_class2,2,1);
3.
4.
5. %% Plot Filtered Data:
6. Page_Num = 49;
7. Filter_Num = [1,30];
8.
```

```

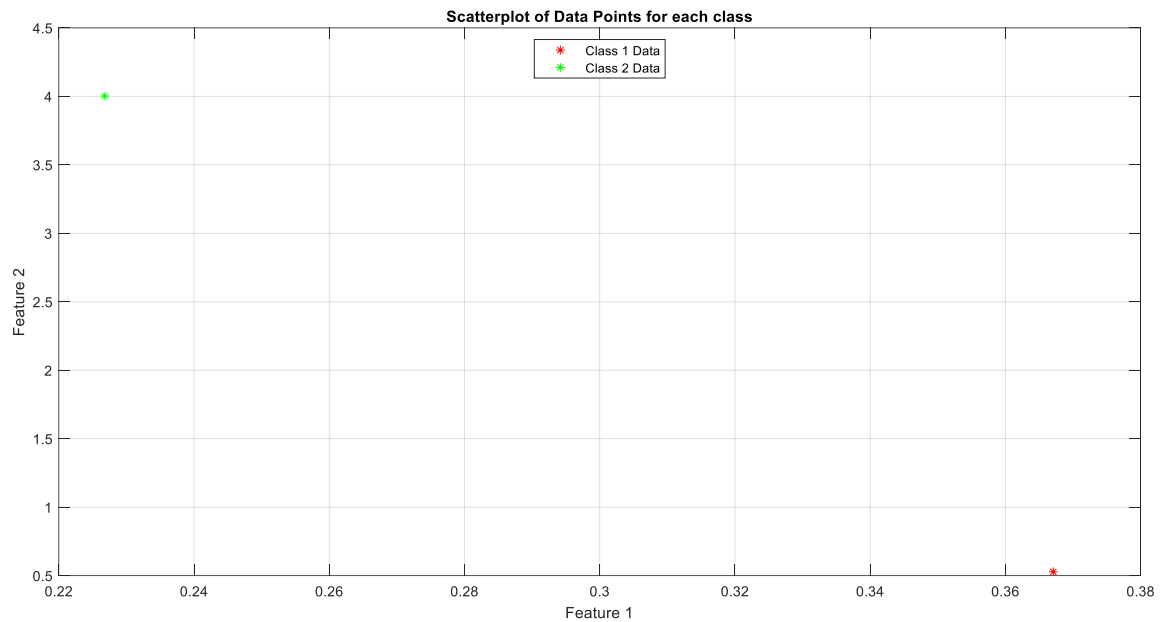
9.
10. Sample_Data_1_Filtered_Class1 = Z_Class1(Filter_Num(1),:,Page_Num);
11. Sample_Data_2_Filtered_Class1 = Z_Class1(Filter_Num(2),:,Page_Num);
12.
13. Sample_Data_1_Filtered_Class2 = Z_Class2(Filter_Num(1),:,Page_Num);
14. Sample_Data_2_Filtered_Class2 = Z_Class2(Filter_Num(2),:,Page_Num);
15.
    
```

خروجی مورد نظر به صورت زیر در آمده است:



شکل 1

برای گزارش پراکندگی برای این دو داده داریم:



شکل 2

محور پایین به عنوان ویژگی اول و محور عمودی به عنوان ویژگی دوم انتخاب شده اند که در واقع هر کدام واریانس داده‌ها پس از اعمال فیلتر مکانی اند. محور پایین حاصل از فیلتر مکانی 1 و محور عمودی حاصل از فیلتر مکانی 30 ام است. مقادیر به صورت عددی نیز عبارتند از:

```

1. Var_Sample_Data_1_Filtered_Class1 =
2.
3.     0.3671
4.
5.
6. Var_Sample_Data_2_Filtered_Class1 =
7.
8.     0.5282
9.
10.
11. Var_Sample_Data_1_Filtered_Class2 =
12.
13.     0.2268
14.
15.
16. Var_Sample_Data_2_Filtered_Class2 =
17.
18.     4.0024
19.

```

(ب)

(ب) فیلترهای مکانی اول و آخر به دست آمده در قسمت الف را رسم کنید. برای هر کلاس، کدام کانال (یا کانال ها) تاثیر بیشتری در خروجی فیلتر دارند؟ در واقع در اینجا متوجه می شویم اثر هر کدام از تصوراتی که فرد در کلاس ۱ و ۲ داشته، در کجای مغز بیشتر است.

پس از رسم ستون اول و آخر وزن فیلتر مورد نظر، در این نمودار مقادیر بیشتر نشان از اهمیت آن سنسور در اندازه گیری دارد:

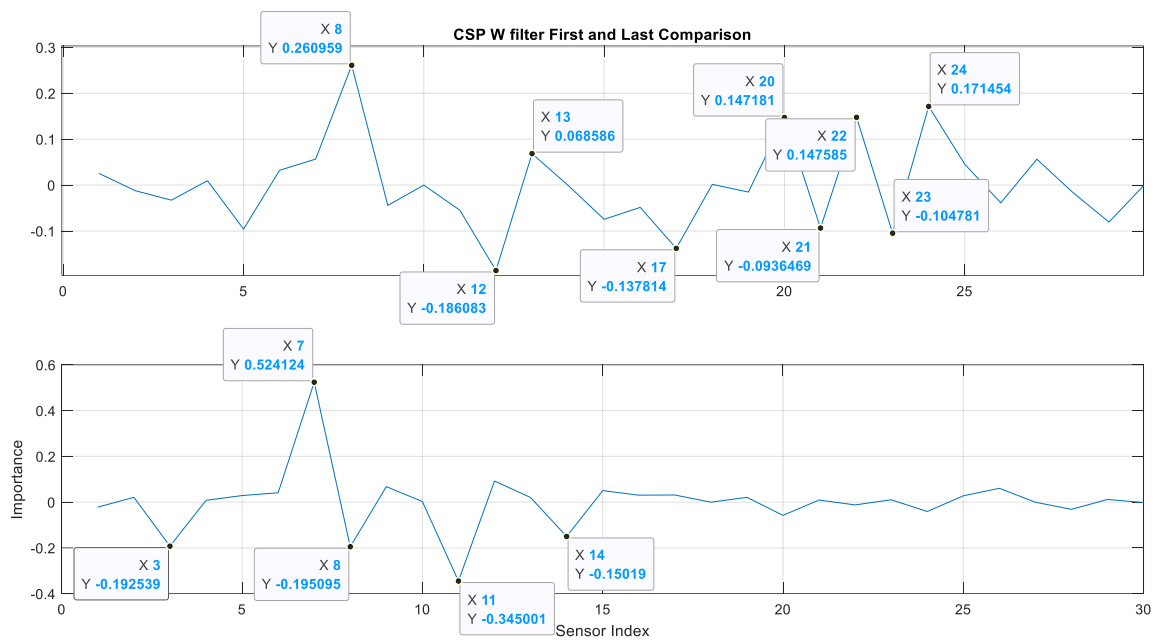
برای فیلتر اول سنسورهای:

[8 ,12,13,17,20,21,22,23,24]

و برای فیلتر دوم سنسورهای:

[3,7,8,11,14]

دارای بیشترین تاثیر هستند.



شکل 3

ج

ج) واریانس داده های فیلتر شده را با در نظر گرفتن ۱۴ فیلتر مکانی مهمتر (متناظر با فیلترهای شماره ۱ تا ۷ و فیلترهای شماره ۲۴ تا ۳۰) استخراج کنید. در واقع هر داده ی آموزش یک ماتریس 256×30 بوده است که در اینجا به یک بردار با ۱۴ درایه تبدیل (و یا فشرده) می شود. حال با استفاده از طبقه بند LDA بهترین تبدیل خطی (WLDA) و مرز متناظر (C) که داده های دو کلاس را از هم جدا می کند، پیدا کنید. توجه داشته باشید که WLDA یک بردار با ۱۴ درایه است و C یک اسکالر است. مقدار WLDA و C به دست آمده را حتما گزارش کنید.

ابتدا داده ها با استفاده از این 14 فیلتر به فضای تصویر می بریم که به ما در واقع یک ماتریس $60 \times 256 \times 14$ می دهد که با گرفتن واریانس از هر کانال از این 14 کانال نهایی، به یک ساختار $60 \times 1 \times 14$ می رسیم که به ازای هر داده ی آموزش، یک بردار 14 تایی شامل واریانس متناظر با 14 فیلتر مورد نظر خواهیم داشت. پس در یک فضای 14 بعدی باید داده ها را از یکدیگر متمایز کنیم.

- همانطور که در 2 بعد می‌توانستیم با یک خط برای مثال داده‌ها را جدا کنیم، که برای معادله‌ی یک خط نیاز به 2 وزن یکی برای بایاس و دیگری برای شیب خط هست، در 14 بعد هم نیاز به 14 وزن برای به دست آوردن معادله‌ی Hyperplane جداکننده‌ی داده‌ها خواهد بود.
- متأسفانه نمی‌توان داده‌ها را در 14 بعد رسم کرد ولی برای مثال ما اینکار را با استفاده از 2 تا از مهمترین فیلترها در 2 بعد انجام می‌دهیم تا نمود عملی داشته باشد. سپس از 2 بعد به 14 بعد تعمیم می‌دهیم.
- مانند قبل میانگین‌گیری را انجام می‌دهیم و در این مورد بردار میانگین ما دارای 14 المان خواهد بود که در واقع نمایانگر میانگین در هر بعد است.
-

مقادیر قابل گزارش برای W و C به صورت زیر اند:

```

1. C =
2.
3. -8.2331e-04
4.
5. >> W_LDA_Chosen'
6.
7. ans =
8.
9. Columns 1 through 11
10.
11. 0.0258 -0.0233 -0.0458 0.0004 -0.0683 0.0603 -0.0044 0.0361 0.0474 -0.0152
12. 0.0150
13. Columns 12 through 14
14.
15. -0.0506 0.0059 -0.0574
16.

```



د) حال به سراغ داده های آزمون می رویم. در ابتدا فرض کنید برچسب (کلاس) داده های آزمون را ندارید. با استفاده از ۱۴ فیلتر مکانی مهمتر که از داده های آموزش به دست آمده بودند و همچنین WLDA و C که آنها هم از داده های آموزش استخراج شده بودند، درباره ی برچسب داده های آزمون تصمیم گیری کنید.

ابتدا دقت را روی داده های آموزش اندازه گیری کردیم:

```

1. Cntr1 = 0;
2. Cntr2 = Cntr1;
3.
4. New_Data_Projected_LDA_Class_1 = tensorprod( W_LDA_Chosen, Z_Class1_Chosen , 1,1 );
5. New_Data_Projected_LDA_Class_2 = tensorprod( W_LDA_Chosen, Z_Class2_Chosen , 1,1 );
6.
7. for i=1:Num_of_Pages
8.     Data_Mean_Check_Class_1 = mean(New_Data_Projected_LDA_Class_1(1,:,i)) ;
9.     Data_Mean_Check_Class_2 = mean(New_Data_Projected_LDA_Class_2(1,:,i)) ;
10.    if( C < Data_Mean_Check_Class_1 ) % Belongs to Class 1
11.        Cntr1 = Cntr1+1;
12.    end
13.    if( C > Data_Mean_Check_Class_2 ) % Belongs to Class 1
14.        Cntr2 = Cntr2+1;
15.    end
16. end
17.
18. Accuracy_over_Train_Data_Class_1 = Cntr1/Num_of_Pages;
19. Accuracy_over_Train_Data_Class_2 = Cntr2/Num_of_Pages;
20.

```

که در این مورد به دقت 100 درصد برای کلاس 1 و دقت 58 درصد برای کلاس 2 رسیدیم.

```

1. Accuracy_over_Train_Data_Class_1 =
2.
3.     1
4.
5. Accuracy_over_Train_Data_Class_2 =
6.
7.     0.5833
8.

```

حال داده های آموزش را ابتدا unbiased کرده و سپس با ضرب در W به دست آمده از روی CSP به فضای تصویر می بریم. سپس 14 تای مهم تر را انتخاب می کنیم و با

نگاشت خطی توسط W_{LDA} به روی یک خط منتقل می‌کنیم. اینجا باید تصمیم‌گیری در مورد لیبل را انجام دهیم.

```

1. % Normalizing W Columns:
2. Norm_Matrix_of_Cols_W_LDA = W_LDA'*W_LDA;
3. for j=1:Chosen_Num_Rows*2
4.     W_LDA(:,j) = W_LDA(:,j)/(Norm_Matrix_of_Cols_W(j,j));
5. end
6.
7. % Choosing the Maximum Discriminator!
8. W_LDA_Chosen = W_LDA(:,1);
9.
10. New_Mean_Class_1 = W_LDA_Chosen'*MIU_Class_1_Mean_Trial;
11. New_Mean_Class_2 = W_LDA_Chosen'*MIU_Class_2_Mean_Trial;
12.
13. C = mean([New_Mean_Class_1 , New_Mean_Class_2]) ; % The Threshold
14.

```

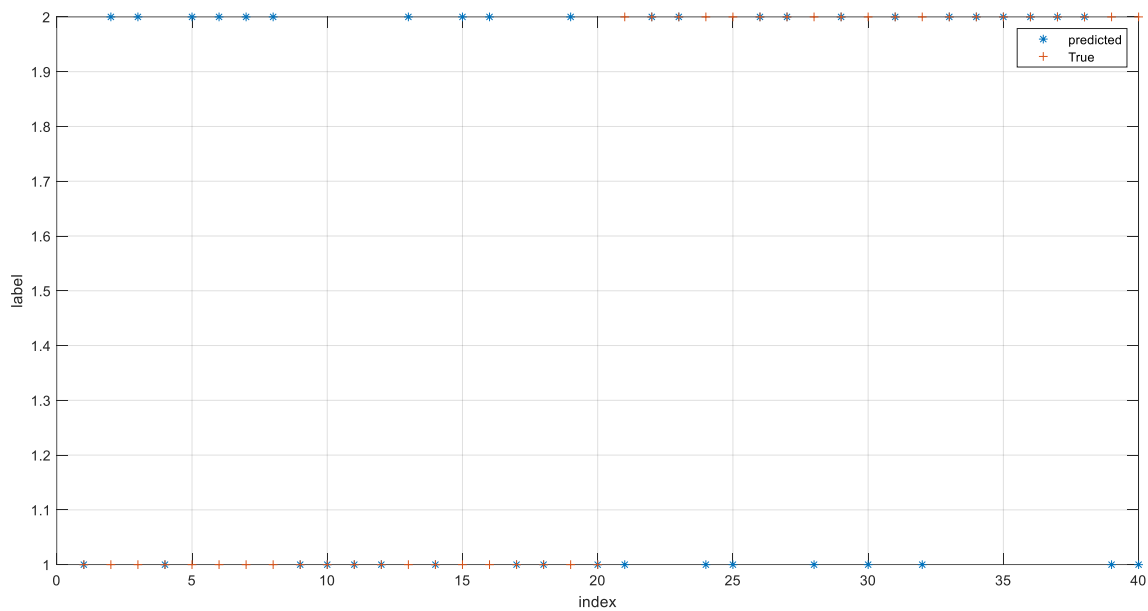
در بالا فرآیند انتخاب مرز و W آورده شده است. در زیر برای داده‌های آزمون داریم:

```

1. %% Now Test Data:
2.
3. Z_Test = tensorprod(W',TestData,2,1);
4. Num_of_Pages_Test = length( TestData(1,1,:));
5.
6. Z_Test_Chosen = zeros(Chosen_Num_Rows*2,Num_of_Cols,Num_of_Pages_Test);
7. Z_Test_Chosen(1:7, :, :) = Z_Test(1:Chosen_Num_Rows, :, :) ;
8. Z_Test_Chosen(8:14, :, :) = Z_Test(Num_of_Rows-Chosen_Num_Rows+1:Num_of_Rows, :, :);
9.
10.
11. Cntr1 = 0;
12. Cntr2 = Cntr1;
13.
14. New_Data_Projected_LDA_Test = tensorprod( W_LDA_Chosen, Z_Test_Chosen , 1,1 );
15.
16. My_Labels = zeros(1,Num_of_Pages_Test);
17. for i=1:Num_of_Pages_Test
18.     Data_Mean_Check_Test = mean(New_Data_Projected_LDA_Test(1,:,i)) ;
19.
20.     if( C < Data_Mean_Check_Test ) % Belongs to Class 1
21.         My_Labels(i) = 1;
22.     else
23.         My_Labels(i) = 2;
24.     end
25.
26. end
27.
28.
29. %%
30. Error = (TestLabel - My_Labels);
31. Accuracy_over_Test_Data = sum(Error==0)/length(Error);
32.

```

پس از مقایسه داریم:



شکل 4

تعداد داده‌های درست تخمین زده شده برابر 22 داده از 40 داده می‌باشد.

به دلیل پایین بودن دقت این Classifier، می‌توان از واریانس به جای میانگین در صورت کسر استفاده کرد و به این ترتیب میزان دقت این کلاس‌بند از 55 درصد به 85 درصد افزایش می‌یابد.

بخش دوم:

در این تمرین می خواهیم روش CCA را برای شناسایی فرکانس تحریک داده های SSVEP استفاده کنیم (steady state visually evoked potentials). یک ماتریس سه بعدی و دو بردار در فایل hw3-2.mat به شرح زیر در اختیار شما قرار داده شده است:

- فرکانس های تحریک مورد استفاده در آزمایش در بردار freq با ابعاد 1×5 قرار دارند.
- داده های ضبط شده در ماتریس data با ابعاد $15 \times 1250 \times 6$ (تعداد کانالها ۶، تعداد نمونه های زمانی ۱۲۵۰ و تعداد آزمایشها ۱۵) قرار دارند. فرکانس نمونه برداری ۲۵۰ هرتز بوده است و یا به عبارت دیگر طول زمانی هر آزمایش ۵ ثانیه می باشد.
- برچسب واقعی داده ها در بردار label با ابعاد 1×15 قرار دارند. در واقع با استفاده از این بردار متوجه می شویم هر داده با تحریک چه فرکانسی ضبط شده است.

در این قسمت، سعی بر آن است که با استفاده از تحلیل CCA که مخفف Canonical Correlation Analysis است، فرکانس های تحریک داده های ضبط شده را پیدا کنیم. برای اینکار، ابتدا ماتریس های مورد نظر را برای فرکانس های داده شده و هارمونیک های آنها تا حد داده شده ی ۴۰ هرتز تولید می کنیم:

```

1. 1. %% Generate X for given frequencies:
2. 2.
3. 3. X_F = cell(length(freq),1);
4. 4.
5. 5. for i=1:length(X_F)
6. 6.     % The frequency itself:
7. 7.     Temp = [sin(2*pi*freq(i)*t); cos(2*pi*freq(i)*t)];
8. 8.     % Consider Harmonics till 40Hz:
9. 9.     for j=2:10
10. 10.        if(j*freq(i)<40)
11. 11.            Temp = [Temp ; sin(2*pi*j*freq(i)*t); cos(2*pi*j*freq(i)*t)];
12. 12.        end
13. 13.    end
14. 14.    X_F{i,1} = Temp;
15. 15. end
16. 16.

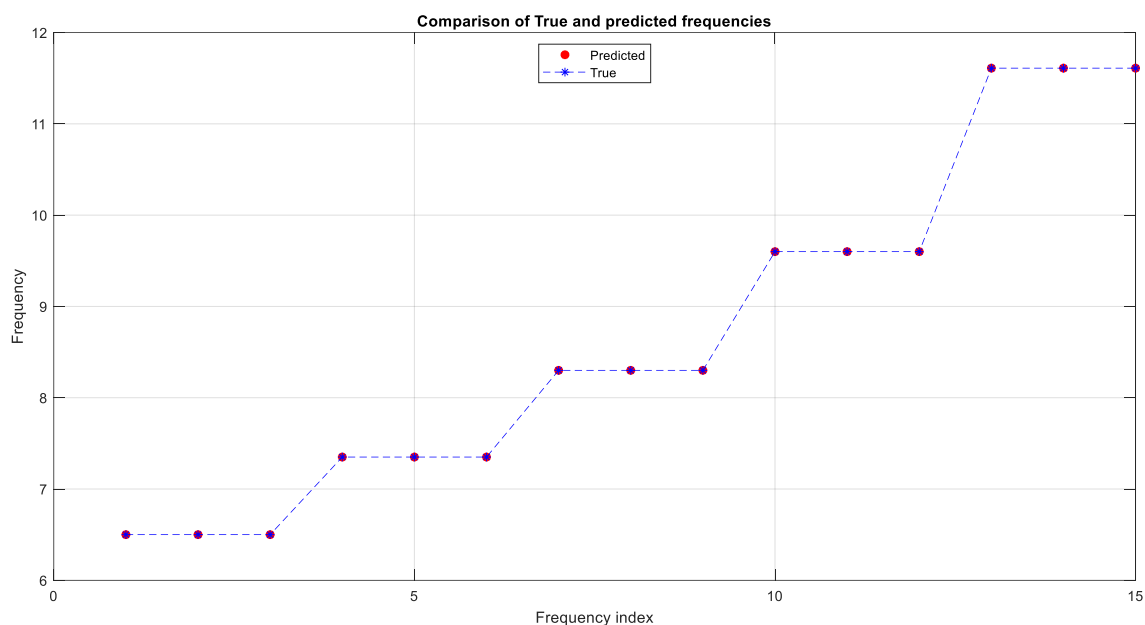
```

سپس برای تعیین فرکانس تحریک داده‌ها، کسر نسبت رایی را به عنوان **Cost Function** تشکیل داده که در صورت تطابق باید مقدار آن بیشینه شده که یعنی زاویه بین دو بردار تشکیل شده توسط ترکیب خطی این دو ماتریس کمینه شده و روی هم بافتند.

برای پیدا کردن مقدار این کسر پس از حل مسئله، باید از روش **GEVD** استفاده کنیم که در واقع مقادیر ویژه همان مقدار کسر و بردارهای ویژه در اینجا متناسب با ضرایب ترکیب خطی در فضای بردارند. هر چه مقدار ویژه برای یک فرکانس بالاتر باشد یعنی به فرکانس تحریم نزدیک تر است پس از بین مقادیر ویژه پیدا شده برای هر آزمایش، بیشترین را به عنوان لیبل در نظر می‌گیریم:

```
1. [Val , pos]= max(Ray_Ratio_for_Each_Freq,[],1);
2. Freq_Labels = freq(pos);
3.
4. %% CComparison:
5.
6. figure();
7. plot(Freq_Labels,"r*","LineWidth",2)
8. hold on
9. plot(label,"b--*")
10. grid on
11. title("Comparison of True and predicted frequencies");
12. legend("Predicted","True",'Location','north');
13. xlabel("Frequency index")
14. ylabel("Frequency")
15.
```

در نهایت برای مقایسه بین فرکانس‌های پیش‌بینی شده و فرکانس‌های واقعی داریم:



شکل 5

که تمامی نمونه‌ها به درستی پیش‌بینی شده اند و دقت این روش برابر 100 درصد برای این داده است.

پایان