تمرین شماره 3

علیرضا حسینی

شماره دانشجویی : ۸۱۰۱۰۱۱۴۲

جداسازی کور منابع

دکتر اخوان

بهار 1402

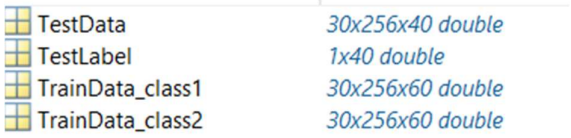
فهرست مطالب

فهرست اشکال

۱-۱- بخش اول

1-1-1- لود کردن دیتاست

ابتدا به کمک دستور ()load. فایل دیتاست را میخوانیم . دیتاست به صورت زیر میباشد.



شکل (۱-۱) محتویات فایل Hw3-1.mat

با توجه به صورت سوال ، داده ها را در متغیر های مد نظر ریخته و میانگین آن ها را صفر میکنیم.

```
%% load data:
dataset = load('hw3-1.mat');
TrainData_class1 = dataset.TrainData_class1;
TrainData_class2 = dataset.TrainData_class2;
TestData = dataset.TestData;
TestLabel = dataset.TestLabel;
%% Zero mean
TrainData_class1 = TrainData_class1 - mean(TrainData_class1,2);
TrainData_class2 = TrainData_class2 - mean(TrainData_class2,2);
TestData = TestData - mean(TestData,2);
```

شکل (۱-۲) کد متلب لود دیتا در هر کدام از متغیر ها و صفر کردن میانگین آن ها

1-1-2- الف

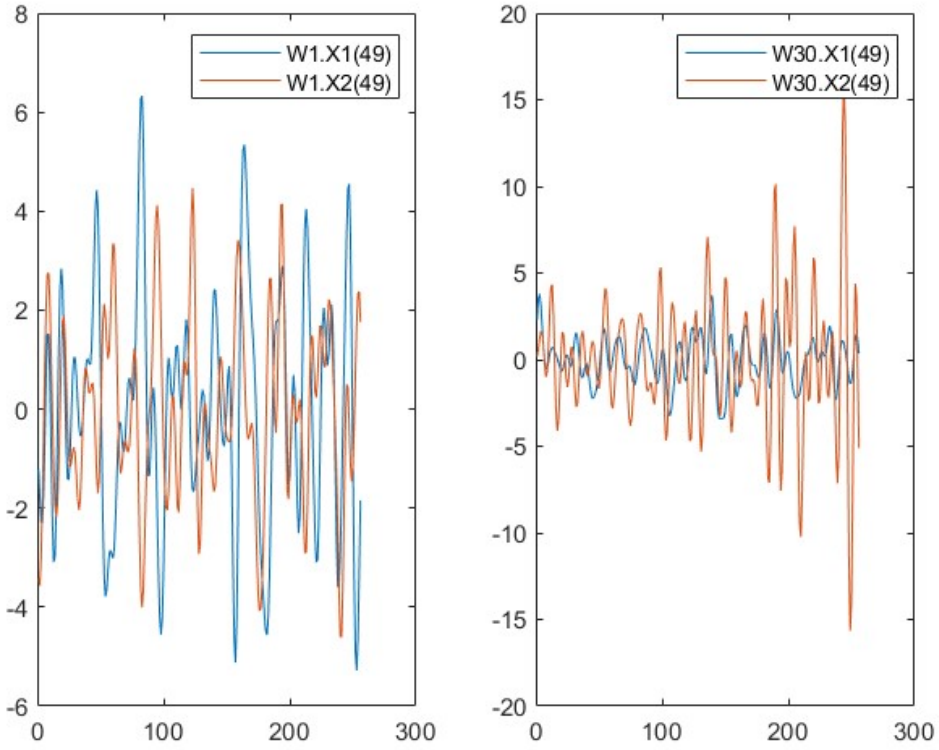
جهت به دست آوردن فیلتر های مکانی CSP مراحل به صورت زیر میباشد:

ابتدا باید Rx هر کلاس را به دست بیاوریم و EIG این 2 را محاسبه کنیم مقادیر ویژه ها را سورت کرده و بردار ویژه های متناظر جواب مساله هست.

در نهایت نیز Wscp را نرمال کرده و میتوان برای هر آزمایش در هر کلاس W^T را در آن ضرب کرده و خروجی را مشاهده کنیم.

W csp گویی یک ترکیب خطی از کانال ها را با واریانس های متفاوت برمیگرداند.

اگر پلات های مد نظر در صورت سوال را رسم کنیم متوجه میشویم که خروجی هر 2 فیلتر در 49 امین آزمایش به صورت زیر میباشد و مشاهده میشود خروجی یک فیلتر پراکندگی بیشتری دارد و خروجی یک فیلتر دیگر پراکندگی کمتری.



شکل (۳-۱) سیگنال فیلترشده متناظر با فیلتر اول و اخر برای ازمایش 49ام هر 2 کلاس

اگر بخواهیم به صورت کمی var ها را گزارش دهیم میتوان از دستور ()var استفاده کرد.

به ترتیب برای هر کلاس و فیلتر مقادیر واریانس به صورت زیر میباشد.

$Var(W_1^T * Train\_class1_{49}) = 5.3904$

$Var(W_1^T * Train\_class2_{49}) = 3.3305$

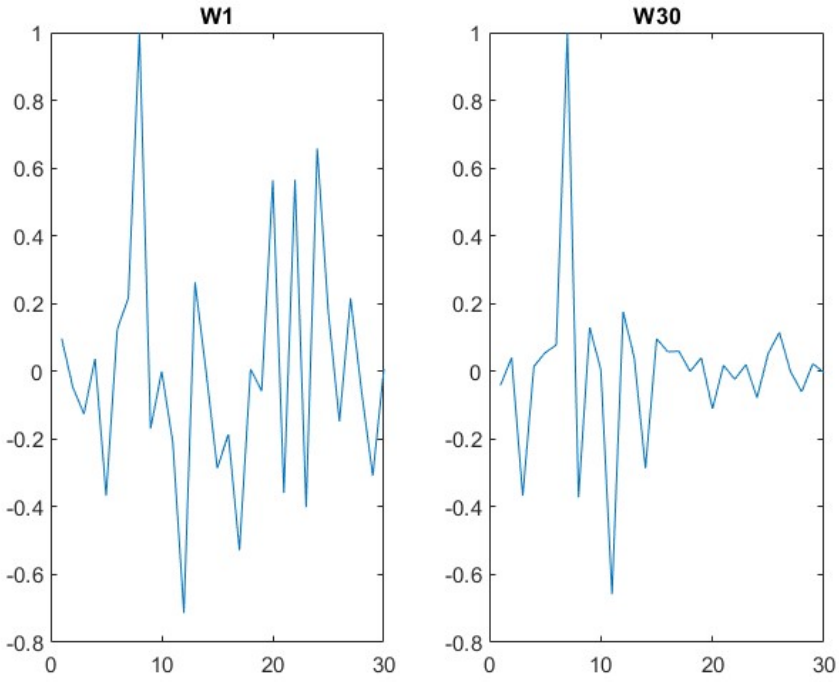
$Var(W_{30}^T * Train\_class1_{49}) = 1.9227$

$Var(W_{30}^T * Train\_class2_{49}) = 14.5696$

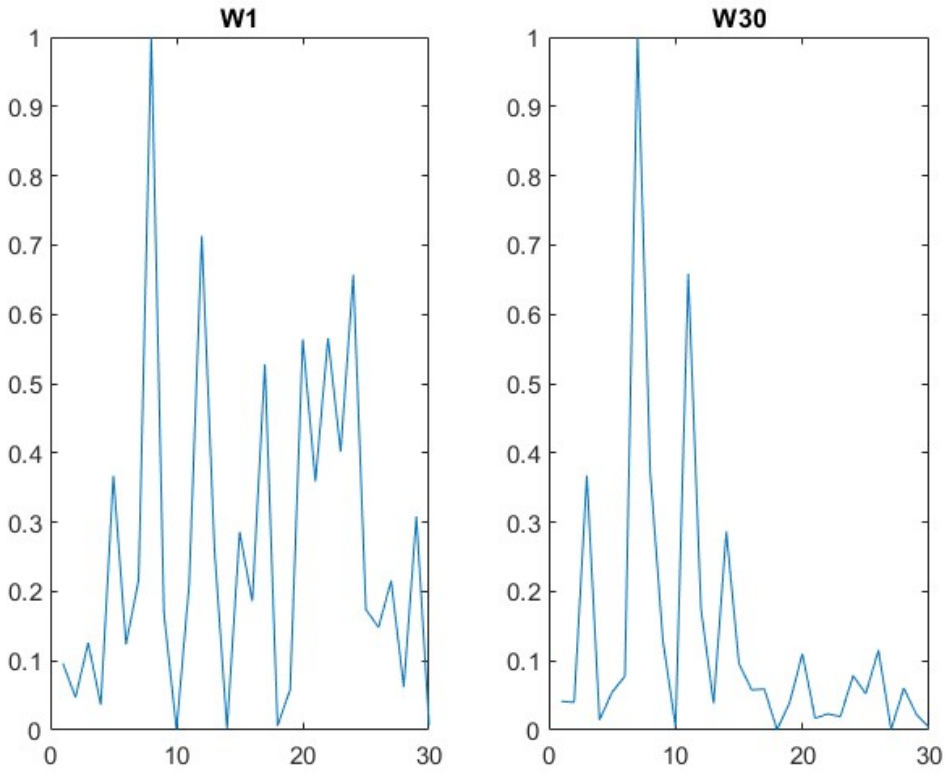
فیلتر اول واریانس روی کلاس 1 را بیشتر میکند و فیلتر دوم واریانس (پراکندگی) را روی کلاس دوم خیلی خیلی زیاد میکند.

3-1-1- ب

اگر قدر مطلق فیلتر های مکانی اول و آخر را رسم کنیم مشاهده میشود که مثلا کانال 8 در این مساله طبقه بندی بسیار مهم میباشد.



شکل (۴-۱) فیلتر های مکانی اول و اخر



شکل (۵-۱) قدر مطلق فیلتر های مکانی اول و آخر

ج -1-1-4

در ادامه ابتدا فیلتر های مکانی مهم را طبق صورت سوال جدا کرده و در ادامه الگوریتم LDA را اجرا میکنیم.

برای به دست آوردن W_{LDA} باید میانگین و کوواریانس های فضای ویژگی (W^{T}_{CSP} * Data) را به دست آورده و در نهایت عبارات زیر را محاسبه کنیم. (mo همان میانگین است)

*=(mo1 – mo2) (mo1-mo2)^{T}

**=Covariance 1 + Covariance 2

در نهایت پس از محاسبه EIG های * و ** ، بردار ویژه متناظر با بزرگترین مقدار ویژه جواب مساله

میباشد.

مقدار WLDA و مرز تصمیم گیری به صورت زیر میباشد.

```
>> WLDA(:,14)

ans =

    0.0015
    0.1543
   -0.0013
    0.1284
   -0.0014
    0.0991
    0.1239
    0.0065
   -0.0092
   -0.0560
   -0.0052
   -0.0975
    0.0166
    0.0194

>> C

C =

    2.3240
```

شکل (۶-۱) مقدار WLDA و ثابت اسکالر مرز C

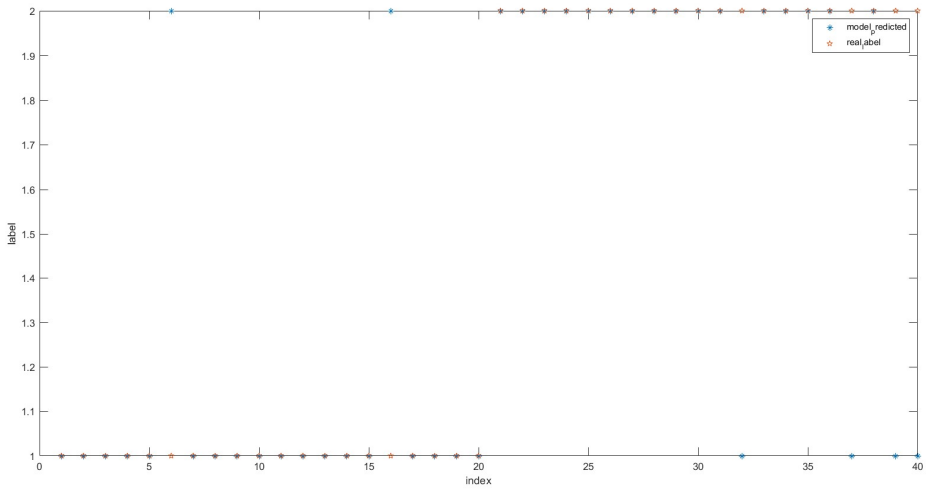
5-1-1- د

برای تصمیم گیری داده ها را در WLDA ضرب میکنیم اگر از C مقدارش بزرگتر بود کلاس 1 آن را لیبل میزنیم و در غیر این صورت آن را کلاس 2 لیبل میزنیم.

ه -1-1-6

برای سرعت محاسبات بیشتر در همان loop بخش د بر روی داده های تست بررسی میکنیم که آیا label زده شده با label اصلی مطابقت دارد یا خیر (در ضمن تمامی پیشبینی ها را هم ذخیره میکنیم) از یک کانتر برای شمارش موارد درست استفاده میشود که بتوانیم در نهایت دقت مدل را روی داده های تست هم ارزیابی کنیم.

دقت مدل روی داده های تست 85 درصد میباشد (34 برچسب از 40 تا درست بوده) در شکل زیر (ستاره های قرمز تو پر با آبی داده هایی هستن که درست برچسب گذاری شده اند)



شکل (۷-۱) عملکرد مدل بر روی داده های تست

با توجه به شکل فوق مدل روی داده های ۳۲ و ۳۷ و ۲۹ و ۶ و ۱۶ تصمیم گیری درستی نداشته است.

۲-۱- بخش دوم – CCA

1-2-1- توضیحات و روش حل

تجزیه و تحلیل (CCA) یک روش آماری چند متغیره است که برای یافتن رابطه خطی بین دو مجموعه از متغیرها استفاده می شود. در مورد داده های SSVEP، دو مجموعه متغیر داریم: داده های ثبت شده و فرکانس های تحریک استفاده شده در آزمایش. CCA می تواند به ما کمک کند تا همبستگی بین دو مجموعه از متغیرها را پیدا کنیم و فرکانس هایی که باعث داده های ثبت شده شده است را شناسایی کنیم.

برای پیاده سازی CCA، ابتدا باید ماتریس های قالب (template یا همان ماتریس X) برای هر فرکانس ایجاد کنیم. ماتریس الگو برای هر فرکانس با گرفتن مقادیر سینوس و کسینوس آن فرکانس و هارمونیک های آن تا 40 هرتز و چیدن آنها به صورت افقی ایجاد می شود. ماتریس حاصل دارای ابعاد = 1250*2*6*40 600000 است که 1250 تعداد نمونه های زمانی، 6 تعداد کانال ها و 2 تا 40 تا تعداد اجزای سینوس و کسینوس است.

هنگامی که ماتریس های الگو را ایجاد کردیم، می توانیم از CCA برای یافتن همبستگی بین داده های ثبت شده و ماتریس های الگو استفاده کنیم. سپس می توانیم از ضرایب همبستگی برای تخمین فرکانس تحریک هر داده استفاده کنیم.

برای پیاده سازی CCA از ابتدا در متلب، باید مراحل زیر را انجام دهیم:

ماتریس های قالب را برای هر فرکانس ایجاد کنید.

برای هر داده ثبت شده، ضرایب همبستگی بین داده ها و هر ماتریس الگو را با استفاده از فرمول محاسبه کنید:

r = corrcoef.(داده ها، ماتریس_الگو)

ضریب همبستگی عنصر مورب ماتریس حاصل است.

فرکانس با بالاترین ضریب همبستگی را به عنوان فرکانس تحریک تخمینی داده ها انتخاب کنید.

فرکانس تخمینی تحریک را با مقدار واقعی مقایسه کنید و دقت را محاسبه کنید.

در متلب نیز می توانیم از تابع داخلی "cannocorr" برای انجام CCA استفاده کنیم. تابع دو ماتریس را به عنوان ورودی می گیرد و همبستگی ها و ضرایب متعارف را برمی گرداند. می‌توانیم از ضرایب متعارف برای تخمین فرکانس تحریک هر داده و مقایسه آن با مقدار واقعی برای محاسبه دقت استفاده کنیم. با این حال، اگر بخواهیم CCA را از ابتدا پیاده سازی کنیم، باید مراحل ذکر شده در بالا را دنبال کنیم.

2-2-1- ایجاد template

کد زیر داده ها را از فایل 'hw3-2.mat' بارگیری می کند که شامل داده های ضبط شده، فرکانس های تحریک استفاده شده در آزمایش و برچسب های هر داده است.

سپس، کد ماتریس های الگو را برای هر فرکانس با گرفتن مقادیر سینوس و کسینوس آن فرکانس و هارمونیک های آن تا 40 هرتز تولید می کند. الگوها با استفاده از فرکانس نمونه برداری 250 هرتز و مدت زمان هر آزمایش که 5 ثانیه یا 1250 نمونه است تولید می شوند.

در نهایت، کد یک آرایه سلولی به نام template ایجاد می کند که ماتریس های الگو را برای هر فرکانس ذخیره می کند. طول آرایه سلولی به اندازه تعداد فرکانس های مورد استفاده در آزمایش است.

```
%% Load data from .mat file
data_file = load('hw3-2.mat');

% Extract variables from the loaded data
excitation_freqs = data_file.freq; % Vector of excitation frequencies
data_labels = data_file.label; % Vector of labels for each data
recorded_data = data_file.data; % Matrix of recorded data
```

شکل (۸-۱) کد متلب لود hw3-2 و تجزیه محتویات آن در متغیر های مختلف

```
% Generate template matrices for each frequency
sampling_freq = 250; % Hz
time_samples = size(recorded_data,2); % Number of time samples
time_vector = 0:(1/sampling_freq):(time_samples-1)*(1/sampling_freq); % Time vector
num_freqs = length(excitation_freqs); % Number of frequencies used in experiment
template_matrices = cell(1,num_freqs); % Cell array to store template matrices

for i = 1:num_freqs
    % Generate sine and cosine values for the current frequency and its harmonics up to 40 Hz
    template = [sin(2*pi*excitation_freqs(i)*time_vector); cos(2*pi*excitation_freqs(i)*time_vector)];

    % Add sine and cosine values for the remaining harmonics
    for k = 2:7
        if (excitation_freqs(i)*k > 40)
            break;
        end
        template = [template; sin(2*pi*k*excitation_freqs(i)*time_vector); cos(2*pi*k*excitation_freqs(i)*time_vector)];
    end

    % Store the template matrix for the current frequency in the cell array
    template_matrices{i} = template;
end
```

شکل (۹-۱) کد متلب تولید template (توضیحات کامل کد کامنت شده است)

3-2-1- آنالیز CCA

در کد زیر ، ابتدا تعداد آزمایش ها از اندازه ماتریس داده ها به دست می آید و یک بردار صفر برای ذخیره برچسب های برآورد شده ایجاد می شود.

سپس برای هر آزمایش، داده های ثبت شده استخراج می شود و ماتریس های کوواریانس Ry، Rx، Rxy و Ryx بر اساس داده های ثبت شده Y و ماتریس X الگو برای هر فرکانس محاسبه می شوند.

در مرحله بعد، ماتریس های همبستگی متعارف SIGMA1 و SIGMA2 با استفاده از ماتریس های کوواریانس محاسبه شده و بردارهای ویژه و مقادیر ویژه آنها به دست می آید. اولین ضرایب همبستگی متعارف و متغیرهای متعارف مربوطه در بردارهای ro و c (برای SIGMA1) و d (برای SIGMA2) ذخیره می‌شوند.

سپس از شاخص حداکثر ضریب همبستگی متعارف در ro برای تخمین فرکانس تحریک آزمایش استفاده می شود و فرکانس تخمینی در بردار label_estimation ذخیره می شود.

```
%% CCA
num_trials = size(recorded_data, 3);
estimated_labels = zeros(size(data_labels));
for trial = 1:num_trials
    trial_data = recorded_data(:,:,trial);
    Ryy = trial_data*trial_data'; % Calculate cross-covariance matrix of recorded data
    correlation_coeffs = zeros(1,num_freqs);
    for i = 1:num_freqs
        template_matrix = template_matrices{i};
        Rxx = template_matrix*template_matrix'; % Calculate autocovariance matrix of template
        Rxy = template_matrix*trial_data'; % Calculate cross-covariance matrix between template and recorded data
        Ryx = Rxy'; % Calculate cross-covariance matrix between recorded data and template
        % Apply CCA to calculate correlation coefficient
        Sigma_1 = (Rxx^-0.5)*Rxy*(Ryy^-1)*Ryx*(Rxx^-0.5);
        [V, Lambda] = eig(Sigma_1);
        [lambda_vals, indices] = sort(diag(Lambda),'descend');
        V = V(:,indices);
        c = V(:,1);
        correlation_coeffs(i) = lambda_vals(1);

        Sigma_2 = (Ryy^-0.5)*Ryx*(Rxx^-1)*Rxy*(Ryy^-0.5);
        [V, Lambda] = eig(Sigma_2);
        [lambda_vals, indices] = sort(diag(Lambda),'descend');
        V = V(:,indices);
        d = V(:,1);
        correlation_coeffs(i) = lambda_vals(1);
    end
    [~, max_index] = max(correlation_coeffs); % Choose the frequency with the highest correlation coefficient
    estimated_labels(trial) = excitation_freqs(max_index);
end
```

شکل (۱۰-۱) کد متلب CCA – توضیحات به عنوان کامنت در کد آمده است

در نهایت، دقت تخمین با مقایسه برچسب های برآورد شده با برچسب های واقعی ارزیابی می شود.

```
%% Calculate accuracy of estimated labels
num_correct = sum(estimated_labels == data_labels);
accuracy = num_correct/length(data_labels);
fprintf('Accuracy of estimated labels: %0.2f\n', accuracy);
```

شکل (۱۱-۱) کد متلب محاسبه دقت

```
Accuracy of estimated labels: 1.00
```

شکل (۱۲-۱) خروجی دقت

با توجه به شکل فوق دقت تخمین 100 درصد میباشد شکل زیر لیبل ها و estimated ها را نمایش میدهد.

data_labels

1x15 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	6.5000	6.5000	6.5000	7.3500	7.3500	7.3500	8.3000	8.3000	8.3000	9.6000	9.6000	9.6000	11.6100	11.6100	11.6100

estimated_labels

1x15 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	6.5000	6.5000	6.5000	7.3500	7.3500	7.3500	8.3000	8.3000	8.3000	9.6000	9.6000	9.6000	11.6100	11.6100	11.6100

شکل (۱۳-۱) Data label و estimated label ها

از آنجایی که دقت الگوریتم CCA 100٪ است، به این معنی است که الگوریتم قادر به شناسایی دقیق فرکانس تحریک برای همه آزمایش‌ها بوده است. این نشان می دهد که روش CCA یک تکنیک قوی و دقیق برای شناسایی فرکانس تحریک یک سیگنال در محیط های نویزدار است.

در نتیجه، الگوریتم CCA با موفقیت برای شناسایی فرکانس تحریک سیگنال های ضبط شده با دقت 100٪ استفاده شد. این روش را می توان به سایر برنامه هایی که نیاز به شناسایی فرکانس در محیط های پر سر و صدا دارند تعمیم داد.