BCICIV

# Format of the Data

Given are continuous signals of 59 EEG channels and, for the calibration data, markers that indicate the time points of cue presentation and the corresponding target classes.

Data are provided in Matlab format (\*.mat) containing variables:

* **cnt**: the continuous EEG signals, size [time x channels]. The array is stored in data type INT16. To convert it to uV values, use cnt= 0.1\*double(cnt); in Matlab.
* **mrk**: structure of target cue information with fields (the file of evaluation data does *not* contain this variable)
  + **pos**: vector of positions of the cue in the EEG signals given in unit *sample*, length #cues
  + **y**: vector of target classes (-1 for class one or 1 for class two), length #cues
* **nfo**: structure providing additional information with fields
  + **fs**: sampling rate,
  + **clab**: cell array of channel labels,
  + **classes**: cell array of the names of the motor imagery classes,
  + **xpos**: x-position of electrodes in a 2d-projection,
  + **ypos**: y-position of electrodes in a 2d-projection.

# Dimension of data

Input dimensions: (800\*4) x 59

Number of classes: 2

Number of electrodes: 59

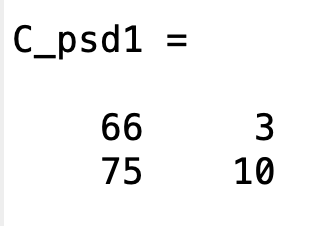
Number of data (for power spectral density): 516

Number of data (for discrete wavelet transform): 1612

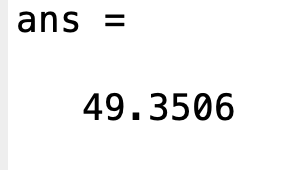
# Power Spectral Density

By using pwelch.

## Confusion Matrix



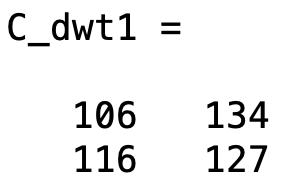
## Accuracy



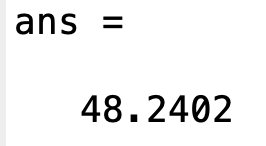
# Discrete Wavelet Transform

Using dwt with 'db4'

## Confusion Matrix



## Accuracy



| clear all; close all; Clc;  dataset = load('BCICIV\_calib\_ds1a.mat'); channels = 59;  x = double(dataset.cnt); x = x'; N = size(x,1)  x1 = x(:, 2091: 2890); x2 = x(:, 2891: 3690); x3 = x(:, 3691: 4490); x4 = x(:, 7692: 8491);  **for** i=1:N  x1(i,:)=bandpass(x1(i,:), [1 50], 500);  [psd1 freq]=pwelch(x1(i,:));   output1(:,i)=psd1(:,1);  [a1, b1] = dwt(x1(i,:),'db4');   cA1(i,:) = a1; % approximation coefficient  x2(i,:)=bandpass(x2(i,:), [1 50], 500);  [psd2 freq]=pwelch(x2(i,:));   output2(:,i)=psd2(:,1);  [a2, b2] = dwt(x2(i,:),'db4');   cA2(i,:) = a2; % approximation coefficient  x3(i,:)=bandpass(x3(i,:), [1 50], 500);  [psd3 freq]=pwelch(x3(i,:));   output3(:,i)=psd3(:,1);  [a3, b3] = dwt(x3(i,:),'db4');   cA3(i,:) = a3; % approximation coefficient  x4(i,:)=bandpass(x4(i,:), [1 50], 500);  [psd4 freq]=pwelch(x4(i,:));   output4(:,i)=psd4(:,1);  [a4, b4] = dwt(x4(i,:),'db4');   cA4(i,:) = a4; % approximation coefficient **end**;  y11 = ones(size(output1, 1),1); y12 = ones(size(output2, 1),1); y13 = 2\*ones(size(output3, 1),1); y14 = 2\*ones(size(output4, 1),1);  output1 = [output1 y11]; output2 = [output2 y12]; output3 = [output3 y13]; output4 = [output4 y14];  output1\_temp = vertcat(output1, output2, output3, output4); cv1 = cvpartition(size(output1\_temp,1),'HoldOut',0.3); idx1 = cv1.test; training = output1\_temp(~idx1,:); test = output1\_temp(idx1,:); Y1\_train = training(:,channels+1); Y1\_test = test(:, channels+1); training = training(:,1:channels); test = test(:, 1:channels);  y31 = ones(1, size(cA1, 2)); y32 = ones(1, size(cA2, 2)); y33 = 2\*ones(1, size(cA3, 2)); y34 = 2\*ones(1, size(cA4, 2));  cA1 = [cA1; y31]; cA2 = [cA2; y32]; cA3 = [cA3; y33]; cA4 = [cA4; y34];  cA1\_temp = vertcat(cA1', cA2', cA3', cA4'); cv3 = cvpartition(size(cA1\_temp,1),'HoldOut',0.3); idx3 = cv3.test; training2 = cA1\_temp(~idx3,:); test2 = cA1\_temp(idx3,:); Y3\_train = training2(:,channels+1); Y3\_test = test2(:,channels+1); training2 = training2(:,1:channels); test2 = test2(:, 1:channels);  result\_PSD1=classify(test,training,Y1\_train,'linear'); result\_dwt1=classify(test2,training2,Y3\_train,'linear');  %Confusion Matrix [C\_psd1,order] = confusionmat(Y1\_test,result\_PSD1) [C\_dwt1,order1]=confusionmat(Y3\_test,result\_dwt1)  %performance cp\_psd1=classperf(Y1\_test,result\_PSD1); cp2\_dwt1=classperf(Y3\_test,result\_dwt1);  %Correction rate for PSD, AR, DISCRETE WAVELET cp\_psd1.CorrectRate \* 100 cp2\_dwt1.CorrectRate \* 100 |
| --- |