Table of Contents

Loading data sets

```
load dataset_BCIcomp1;
load labels_data_set_iii;
```

defining the parameters

```
fs=128;
n_trials=size(x_train,3); %140
n_channels=size(x_train,2); %3
n=size(x_train,1); %1152
kernel_size=256; %as mentioned in the essay
k=546; % best works with RBF kernel
```

filter design

```
order=3; type='bandpass';
[b_delta,a_delta]=butter(order,[0.01 4]/(fs/2),type);
[b_theta,a_theta]=butter(order,[4 8]/(fs/2),type);
[b_alpha,a_alpha]=butter(order,[8 12]/(fs/2),type);
[b_beta,a_beta]=butter(order,[12 30]/(fs/2),type);
[b_gamma,a_gamma]=butter(order,[30 63.99]/(fs/2),type);
```

extracting features for trainig

```
for m=1:n_trials
    temp1=[];%zeros(n_channels*n_features,1)
    for j=1:n_channels
        x=x_train((k:k+kernel_size-1),j,m);
        x_delta=filtfilt(b_delta,a_delta,x);
        x_theta=filtfilt(b_theta,a_theta,x);
        x_alpha=filtfilt(b_alpha,a_alpha,x);
        x_beta=filtfilt(b_beta,a_beta,x);
        x_gamma=filtfilt(b_gamma,a_gamma,x);
        %computing features for selected area
        channel_j_features=myFeatureExtarction(x_delta);
        temp1=cat(   1 , temp1 , channel_j_features   );
        channel_j_features=myFeatureExtarction(x_theta);
        temp1=cat(   1 , temp1 , channel_j_features   );
```

```
            channel_j_features=myFeatureExtarction(x_alpha);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_beta);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_gamma);
            temp1=cat(   1 , temp1 , channel_j_features   );
        end
        temp2(:,m)=temp1;
    end
    features_train(:,:)=temp2;

    for m=1:n_trials
        temp1=[];%zeros(n_channels*n_features,1)
        for j=1:n_channels
            x=x_test((k:k+kernel_size-1),j,m);
            x_delta=filtfilt(b_delta,a_delta,x);
            x_theta=filtfilt(b_theta,a_theta,x);
            x_alpha=filtfilt(b_alpha,a_alpha,x);
            x_beta=filtfilt(b_beta,a_beta,x);
            x_gamma=filtfilt(b_gamma,a_gamma,x);
    %         computing features for selected area
            channel_j_features=myFeatureExtarction(x_delta);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_theta);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_alpha);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_beta);
            temp1=cat(   1 , temp1 , channel_j_features   );
            channel_j_features=myFeatureExtarction(x_gamma);
            temp1=cat(   1 , temp1 , channel_j_features   );
        end
        temp2(:,m)=temp1;
    end
    features_test(:,:)=temp2;
```

applying PCA

```
n_PCA_features=34;
[~,newdata] = pca(features_train(:,:)');
newdata=newdata(:,1:n_PCA_features)';
PCA_features_train(:,:)=newdata;

[~,newdata] = pca(features_test(:,:)');
newdata=newdata(:,1:n_PCA_features)';
PCA_features_test(:,:)=newdata;
```

SVM-RBF

gaussian

```
sampleX_train=PCA_features_train;
```

```
sampley_train=y_train;
model=fitcsvm(sampleX_train',sampley_train,'Standardize',1,...
    'KernelFunction','RBF','KernelScale','auto');
%testing the model
sampleX_test=PCA_features_test;
op=predict(model,sampleX_test');
%computing the accuracy using leave one out method
sampley_test=y_test;
accuracy_normal_validation_with_PCA=sum(op==sampley_test)...
    /length(sampley_test) *100
%accuracy is 45% in this method
```

```
accuracy_normal_validation_with_PCA =

   45.7143
```

Published with MATLAB® R2020a