Table of Contents

```
clear; clc; close all;

GoalDatasetName = 'ALL_HC_GVS_OFF';
data_path = 'READY4DL_500.mat';

load(data_path, GoalDatasetName);
rng('default');
plot_path = 'Plots/'; mkdir(plot_path);

fs = 500;
dt = 1/fs;

GoalDataset = eval(GoalDatasetName);
[pt_no, epochs_no] = size(GoalDataset);
% r_vals = 50:50:100; nstacks = 50:50:150;
r_vals = 100; nstacks = 150;

reEpoch = 250;
nPl = 4;

for pt = 1 : 1 %patients

    data = cat(3,GoalDataset{pt,:});
    data = zscore(data);
    data = permute(data,[2,1,3]);
    data = reshape(data, [size(data,1), reEpoch,
 size(data,3)*size(data,2)/reEpoch]);

    nepoch = size(data,3);
    reconErrorTrain = NaN(length(nstacks), length(r_vals), nepoch);
    reconErrorTest = NaN(length(nstacks), length(r_vals), nepoch);

    %     for k = 1:nepoch
    for k = [30, 60, 90, 120]

        dataTrain = data(:,:,setdiff(1:nepoch, k));
        dataTrainMat{1} = reshape(data(:,:,1:k-1),size(data,1),[]);
        dataTrainMat{2} = reshape(data(:,:,k+1:end),size(data,1),[]);

        dataTest = data(:,:,k);

        for r = 1:length(r_vals)

            for n = 1:length(nstacks)
```

```
                fprintf('Starting CV = %d, nstacks = %d, r = %d ....
\n', k, nstacks(n),r_vals(r));
```

DMD Computations

```
               Xaug = []; Xaug1 = []; Xaug2 = [];
               for ii = 1:2
                   temp = genTimeShiftEmbedding(dataTrainMat{ii},
nstacks(n));
                   Xaug1 = cat(2,Xaug1,temp(:,1:end-1));
                   Xaug2 = cat(2,Xaug2,temp(:,2:end));
               end

               for ii = 1:size(dataTrain,3)
                   temp = genTimeShiftEmbedding(dataTrain(:,:,ii),
nstacks(n));
                   Xaug = cat(3,Xaug,temp);
               end

               Xaug_test = [];
               for ii = 1:size(dataTest,3)
                   Xaug_test =
cat(3,Xaug_test,genTimeShiftEmbedding(dataTest, nstacks(n)));
               end

               DmdStruct = runLowRankDMD(Xaug1, Xaug2, r_vals(r),
dt);

               tempErr = NaN(1,size(Xaug,3));
               for ii = 1:size(Xaug,3)
                   [Xdmd_train(:,:,ii), tempErr(:,ii)] =
predictDMD(DmdStruct, Xaug(:,:,ii), nstacks);
               end
               reconErrorTrain(n, r, k, :) = nanmean(tempErr);

               tempErr = NaN(1,size(Xaug_test,3));
               for ii = 1:size(Xaug_test,3)
                   [Xdmd_test(:,:,ii), tempErr(:,ii)] =
predictDMD(DmdStruct, Xaug_test(:,:,ii), nstacks);
               end
               reconErrorTest(n, r, k, :) = nanmean(tempErr,2);

               X = revTimeShiftEmbedding(Xaug, nstacks);
               X = revTimeShiftEmbedding(Xaug, nstacks);
               X = revTimeShiftEmbedding(Xaug, nstacks);

               plotTrainTest(reshape(Xaug(:,:,k-nPl:k-1),
size(Xaug,1), []), reshape(Xaug_test, size(Xaug,1), []),...
                   reshape(Xdmd_train(:,:,k-nPl:k-1),
size(Xdmd_train,1), []), reshape(Xdmd_test, size(Xdmd_test,1), []),
fs,...
                   pt,k,nstacks(n),r_vals(r),reconErrorTrain(n, r,
k),...
```

```
                    reconErrorTest(n, r, k),plot_path);

            end
        end
    end
    save(sprintf('Error_2020_07_21_Test_Epoch_0.5_Sub
%d.mat',pt),'reconErrorTest', 'reconErrorTrain');
end
```

```
Error using load
'READY4DL_500.mat' is not found in the current folder or on the MATLAB
 path, but exists in:
    /Users/abhijitc/Documents/Abhijit/NeuroData_Tutor/EEG_patient_id/
EEG_Patient_Lib/DMD_CrossValidation
    /Users/abhijitc/Documents/Abhijit/NeuroData_Tutor/EEG_patient_id/
googledrive-archive/RestData

Change the MATLAB current folder or add its folder to the MATLAB path.

Error in DY1_DMD_XAugment_CV (line 6)
load(data_path, GoalDatasetName);
```

Plot

```
fig = figure;
subplot(1,2,1);
heatmap(r_vals, nstacks, nanmedian(reconErrorTrain,3));
colormap hot;
xlabel('r - no. of dimensions');
ylabel('nstacks');
title('Training Error');
subplot(1,2,2);
heatmap(r_vals, nstacks, nanmedian(reconErrorTest,3));
colormap hot;
xlabel('r - no. of dimensions');
ylabel('nstacks');
title('Test Error');
```

Functions

```
function DmdStruct = runLowRankDMD(X,Y, r, dt)

[U, S, V] = svd(X, 'econ');
Ur = U(:, 1:r);
Sr = S(1:r, 1:r);
Vr = V(:, 1:r);

Atilde = Ur'*Y*Vr/Sr;
[W, D] = eig(Atilde);
Phi = Y*Vr/Sr*W;

DmdStruct.r = r;
DmdStruct.dt = dt;
DmdStruct.A = Atilde;
DmdStruct.Phi = Phi;
DmdStruct.Lambda = diag(D);
DmdStruct.Freq = imag(log(diag(D))/dt)/(2*pi);
DmdStruct.omega = log(diag(D))/dt;

end

function Xaug = genTimeShiftEmbedding(X, nstacks)

% construct the augmented, shift-stacked data matrices
Xaug = [];
for st = 1:nstacks
    Xaug = [Xaug; X(:, st:end-nstacks+st)];
end

end

function X = revTimeShiftEmbedding(Xaug, nstacks)

ch = size(Xaug,1)/nstacks;
cols = size(Xaug, 2);
T = cols + nstacks - 1;

X_big = NaN(ch,T,cols);

for col = 1:cols
    X_big(:,col:col+nstacks-1,col) = reshape(Xaug(:,col),ch,nstacks);
end

X = nanmean(X_big, 3);

end

function [Xdmd, reconError] = predictDMD(DmdStruct, X, nstacks)

Phi = DmdStruct.Phi;
omega = DmdStruct.omega;
r = DmdStruct.r;
dt = DmdStruct.dt;

% Compute DMD mode amplitudes b
x1 = X(:,1);
b = Phi\x1;

% DMD reconstruction
mm1 = size(X, 2); % mm1 = m - 1
time_dynamics = zeros(r, mm1);
t = (0:mm1-1)*dt; % time vector

for iter = 1:mm1
    time_dynamics(:,iter) = (b.*exp(omega*t(iter)));
end
```

```
Xdmd = Phi * time_dynamics;

reconError(1) = immse(X, Xdmd);

X = revTimeShiftEmbedding(X, nstacks);
Xdmd = revTimeShiftEmbedding(real(Xdmd), nstacks);

reconError(2) = immse(X, Xdmd);
reconError(3) = immse(X(:,nstacks+1:end), Xdmd(:,nstacks+1:end));

end

function [] = plotSVD(X, n)

[u1,s1,v1]=svd(X,'econ');

subplot(3,1,1), plot(diag(s1)/(sum(diag(s1))),'ro','Linewidth',3);
 title("Eigen values");
subplot(3,1,2), plot(v1(:,1:n),'Linewidth',2); title("V basis");
subplot(3,1,3), plot(u1(:,1:n),'Linewidth',2); title("U basis");

end


function [] = plotCompare(X, Xdmd, chs, rows, cols, fs)

for ch = chs
    subplot(rows, cols, ch);
    t = (1:size(X,2))/fs;
    plot(t, X(ch,:)); hold on;
    plot(t, Xdmd(ch,:));
    if ch == 1
        legend("X", "Xdmd");
    end
end

end

function [] = plotTrainTest(X, X_test, Xdmd, Xdmd_test, fs,
 pt,ep,nstacks,r,reconErrorTrain,reconErrorTest,plot_path)
% Reconstruction error

X = revTimeShiftEmbedding(X, nstacks);
X_test = revTimeShiftEmbedding(X_test, nstacks);
Xdmd = revTimeShiftEmbedding(real(Xdmd), nstacks);
Xdmd_test = revTimeShiftEmbedding(real(Xdmd_test), nstacks);

fig = figure;
set(fig, 'Position', [0 0 720 720]);
plotCompare(X, real(Xdmd), 1:27, 9, 3, fs)
suptitle(sprintf('Sub No: %d, Epoch No: %d, nstacks = %d, r = %d,
 Training Error = %.2f',pt,ep,nstacks,r,reconErrorTrain));
saveas(fig, fullfile(plot_path,sprintf('ReconErrorTrain_CV_Sub%d_Epoch
%d_nstacks%d_r%d.png',pt,ep,nstacks,r)));
```

```
fig = figure;
set(fig, 'Position', [0 0 720 720]);
plotCompare(X_test, real(Xdmd_test), 1:27, 9, 3, fs)
suptitle(sprintf('Sub No: %d, Epoch No: %d, nstacks = %d, r = %d, Test
 Error = %.2f',pt,ep,nstacks,r,reconErrorTest));
saveas(fig, fullfile(plot_path,sprintf('ReconErrorTest_CV_Sub%d_Epoch
%d_nstacks%d_r%d.png',pt,ep,nstacks,r)));

end
```

Published with MATLAB® R2019b