Project Assignment System Identification 2023-2024

Part 1. Fitting an unknown function

- Chis Dalina

Index: 9

Problem Statement

- The primary objective of the project is to create an appropriate approximation for an unknown function, using the data set provided by the teacher. In pursuit of this goal, I constructed a polynomial approximator for the function given by the data (index 9) and established a model fitting using linear regression. A lot of advanced models use linear regression as a base. It offers a simple model, characterized by its ease of understanding.

Generation of Regressors and tuning process

In order to find the polynomial of a certain degree m, I created a function: generatePoly, which is my key feature, that automates the process. Let's consider the regressed variable Y, defined as:

$$Y = \begin{bmatrix} y[1,1] \\ y[1,2] \\ \cdots \\ y[L1,L2] \end{bmatrix}$$

,where L1 represents the length of x1(input parameter given in the data set) and L2 represents the length of x2(input parameter given in the data set).

The regressor ϕ will be determined using the function I mentioned before, starting from the formula:

$$\boldsymbol{\phi} = [\boldsymbol{\phi}(\mathbf{1}, \mathbf{1}), \boldsymbol{\phi}(\mathbf{1}, \mathbf{2}), \ldots, \boldsymbol{\phi}(\boldsymbol{L}\mathbf{1}, \boldsymbol{L}\mathbf{2})]T$$

,where the elements are the coefficients determined by my function.

The parameter vector θ:

$$\theta = \begin{bmatrix} \theta 1 \\ \theta 2 \\ \cdots \end{bmatrix}$$

The determination of θ using the formula:

$$Y = \phi * \theta$$

I extract θ using left matrix division:

$$\theta = \phi \backslash Y$$

Having the regressor ϕ and θ, I can find the approximation:

$$\widehat{Y} = \phi \cdot \theta$$

After I calculated $\widehat{Y}$, I can calculate the MSE as a function of m with the formula:

$$\frac{1}{N} \sum_{i=1}^{N} \left(y_i - \widehat{y_i} \right)^2$$

, where N represents the number of data points, yi is the output parameter for the identification data and the other one is the approximation computed for the identification data.

The **evolution of the errors** is presented below (Figure 1). With blue is represented the evolution of MSE for the identification data set and with red for validation data set .

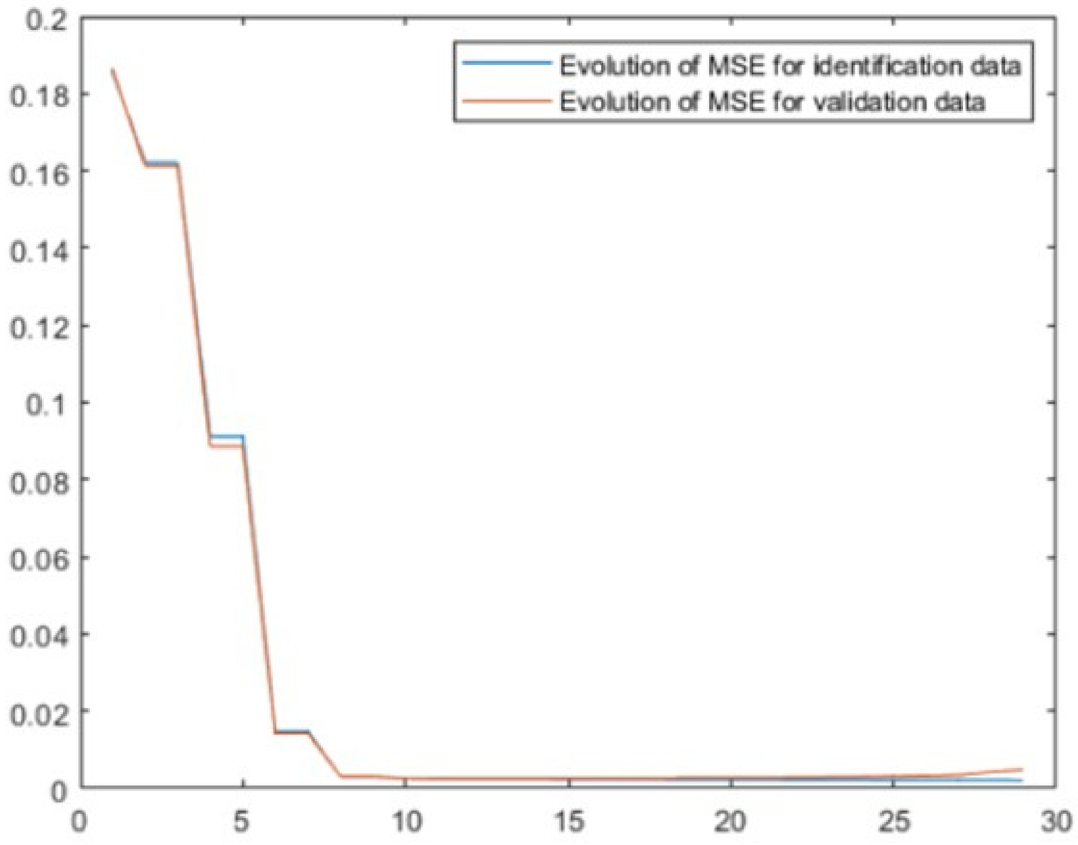
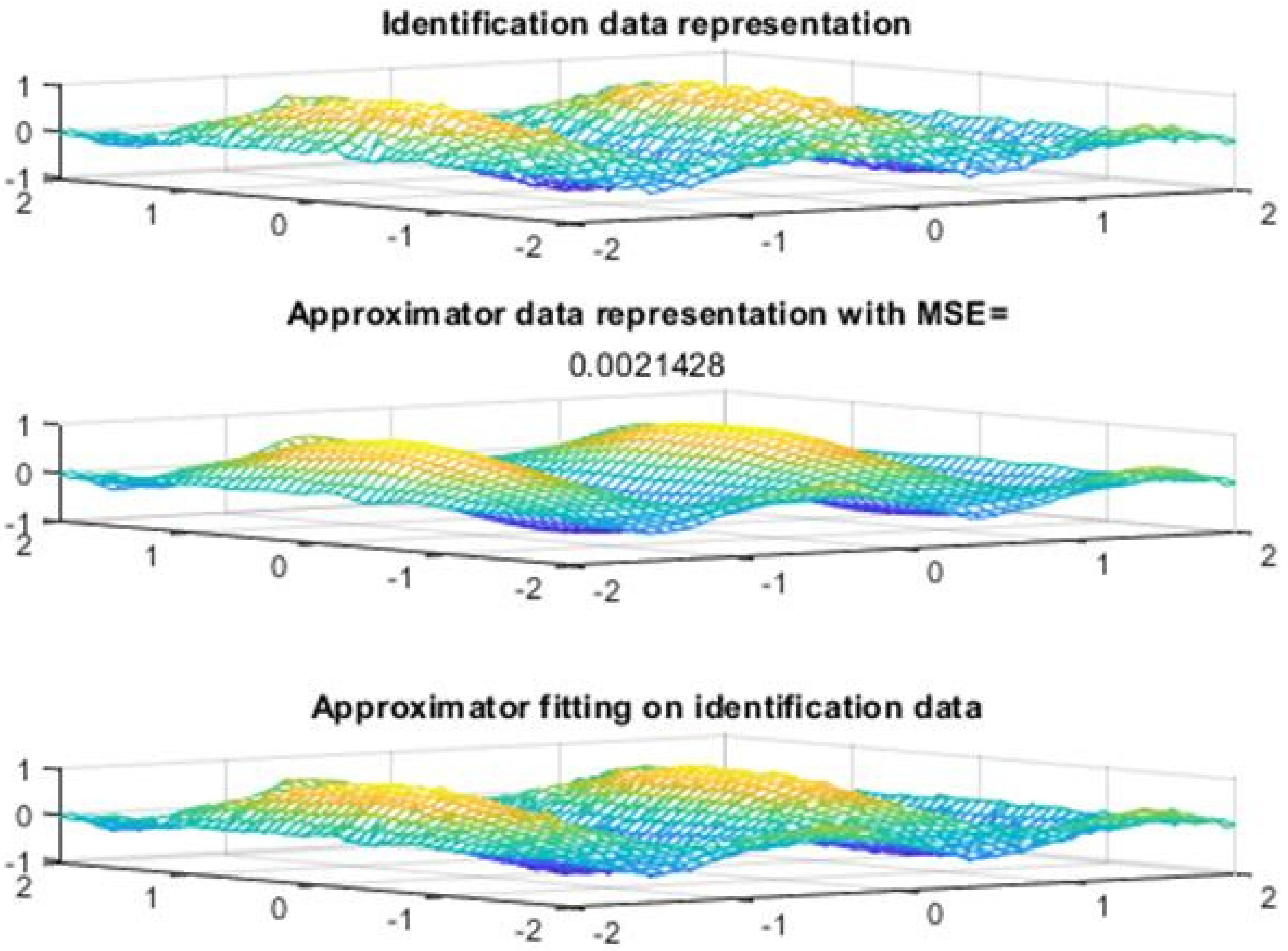


Figure 1

Tuning Results

- In Figure 2 I have 3 subplots containing the identification data representation, having the best approximator data representation for m =30 and MSE=0.0021451 and the approximator fitting on the identification data.



Identification data representation
Approximator data representation with MSE=
0.0021428
Approximator fitting on identification data

Figure 2

In Figure 3 I have 3 subplots containing the Validation data representation, the best approximator data representation for m =11 and MSE=0.0025808 and the approximator fitting on the validation data.

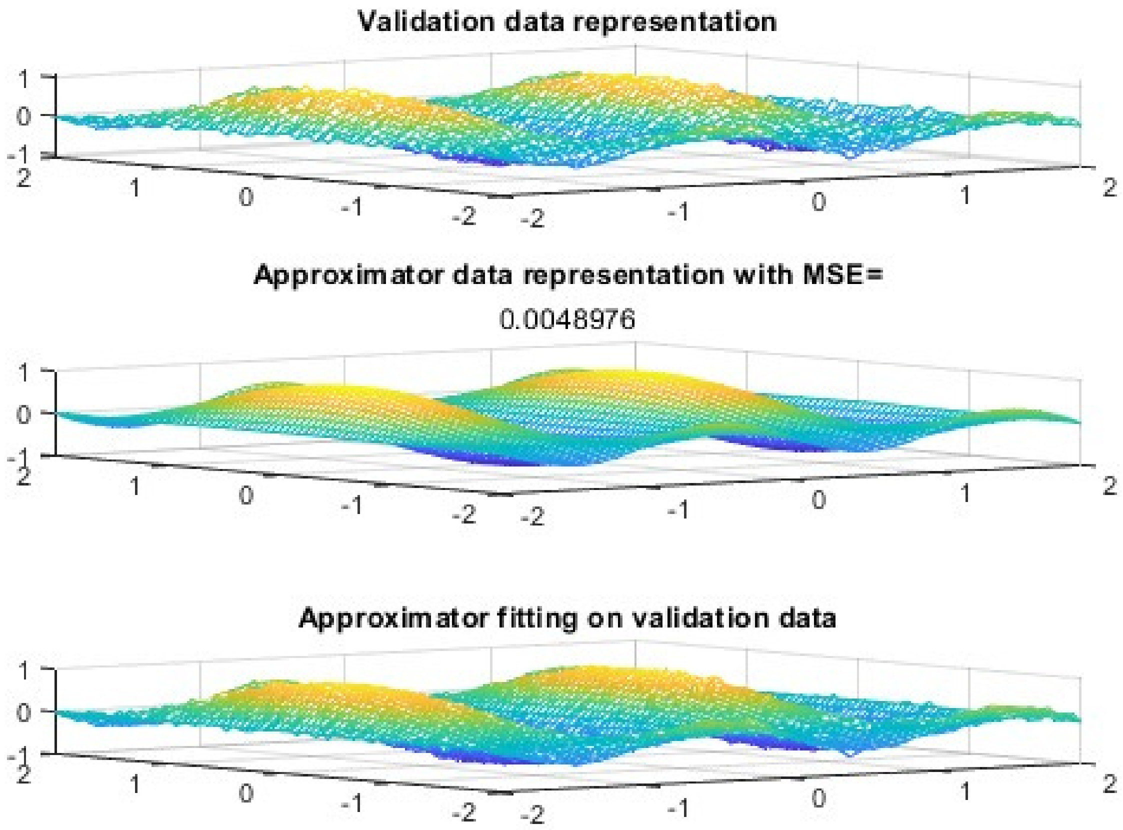


Figure 3

Code

```
load("proj_fit_09.mat")
% We declare our variables
x1=id.X{1,1};
x2=id.X{2,1};
y=id.Y;

x1v=val.X{1,1};
x2v=val.X{2,1};
yval=val.Y;

MSEmin=200;
MSEidMin=300;
mseID_vector=[];
mse_vector=[];
yHatID_min=[];
yHat_min=[];
```

```
% We start a for loop to determine the most convenient degree of the
% polynomial
for  m=2:30
PHI = [];% We need to calculate the regressor for id data
line_nr= 1;

        for i=1:length(x1)
            for j=1:length(x2)
                PHI_line = generatePoly(i,j,m,x1,x2);% We created a function to
generate the polynomial
                for index=1:length(PHI_line)
                    PHI(line_nr,index) = PHI_line(index);
                end
                line_nr = line_nr + 1;
            end
        end
% Calculation of theta
y_resh=reshape(y,length(x1)*length(x2),1);
THETA=PHI\y_resh;

yhatID=PHI*THETA;% Calculation of the aproximation of Y for identification
```

```
y_r = reshape(y,length(y)*length(y), 1);
MSEid=1/length(yhatID)*sum((y_r-yhatID).^2);% Calculation of the MSE of the
identification data
mseID_vector(m-1)=MSEid;

 % Finding the smallest MSE that will give the most convenient degree for
 % the identification data
    if MSEidMin>MSEid
        MSEidMin=MSEid;
        yHatID_min=yhatID;
        degreeID_min=m;
    end

% We calculate the regressor for the validation data
PHIval = [];
linenr_val = 1;
        for i=1:length(x1v)
            for j=1:length(x2v)
                PHI_line = generatePoly(i,j,m,x1v,x2v);
                for index=1:length(PHI_line)
                    PHIval(linenr_val,index) = PHI_line(index);
                end
                linenr_val = linenr_val + 1;
            end
        end

yhat=PHIval*THETA;% Calculation of the aproximation of Y for validation
```

```
yval_r = reshape(yval, length(yval)*length(yval), 1);
MSE=1/length(yhat)*sum((yval_r-yhat).^2);% Calculation of the MSE of the
validation data
mse_vector(m-1)=MSE;

 % Finding the smallest MSE that will give the most convenient degree for
 % the validation data

     if MSEmin>MSE
         MSEmin=MSE;
         yHat_min=yhat;
         degree_min=m;
     end

 end

subplot(311)
mesh(x1v,x2v,yval);
title('Validation data representation')
subplot(312)
yHat_r=reshape(yHat_min,71,71);
mesh(x1v,x2v,yHat_r);
title('Approximator data representation with MSE=',MSEmin)
```

```
subplot(313)
mesh(x1v,x2v,yval);
hold on;
newZLim=[-1,1];
zlim(newZLim);
mesh(x1v,x2v,yHat_r);
title('Approximator fitting on validation data')
hold off;
figure;
subplot(311)
mesh(x1,x2,y);
title('Identification data representation')
subplot(312)
yHatID_r=reshape(yHatID_min,51,51);
mesh(x1,x2,yHatID_r);
title('Approximator data representation with MSE=',MSEidMin)
subplot(313)
mesh(x1,x2,y);
hold on
mesh(x1,x2,yHatID_r);
hold off;
title('Approximator fitting on identification data')
figure;
plot(mseID_vector)
hold on;
plot(mse_vector)
hold off;
legend('Evolution of MSE for identification data','Evolution of MSE for validation
data')
```

Function:

```
function poly = generatePoly(i,j,m,x1,x2)
    poly = [];
    for x1_power = 0:m
        for x2_power = 0:m
            if (x1_power + x2_power) <= m
             poly = [poly, x1(i)^x1_power * x2(j)^x2_power];
            end
        end
    end
 end
```