Facultatea de Automatică și Calculatoare
Automatică și Informatică Aplicată

Proiect
Identificarea sistemelor 2021-2022

Studenți:
Dănescu Miruna Ioana
Iaromi Cristian Adrian
Mănase Mihai Sebastian

Coordonator:
Vicu Mihalis Maer

Grupa: 30133
Index: 5/5

Cuprins

Partea 1. Modelarea unei funcții necunoscute

1. Introducere

Partea de modelare a acestui proiect constă în programarea unui aproximator polinomial cu grad configurabil cu scopul de a găsi un grad *m* optim pentru aproximarea unei funcții necunoscute de două variabile.

Pornim de la un set de date de identificare folosit pentru găsirea modelului pe care îl vom testa pe un set de date de validare. Ambele structuri de date conțin o colecție de coordonate X pentru intrări și o colecție de ieșiri corespunzătoare Y. Identificarea celui mai bun *m* constă în testarea algoritmului de regresie pentru diferite grade.

2. Regresie liniară

Algoritmul de regresie constă în determinarea vectorului de parametri θ. În acest sens se construiește o matrice de regresori Φ, în care elementele sunt polinoame de grad *m*, cu 2 variabile.

Pornim de la ecuația Y = Φ * θ, în care:

- Y este un vector coloană de dimensiune $M_{n*n,1}(R)$, unde n = 41:

$Y = [y_0\ y_1\ y_2\ \dots\ y_n]^T$

- Φ este o matrice de dimensiune $M_{n*n,k}(R)$, care pentru un m =2, k =6 :

$$\Phi = \begin{pmatrix} 1 & x1(1) & x2(1) & x1(1)^2 & x2(1)^2 & x1(1)\cdot x2(1) \\ 1 & x1(1) & x2(2) & x1(1)^2 & x2(2)^2 & x1(1)\cdot x2(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x1(41) & x2(41) & x1(41)^2 & x2(41)^2 & x1(41)\cdot x2(41) \end{pmatrix}$$

- θ este un vector coloană de dimensiune $M_{k,1}(R)$:

$\theta = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6]^T$

Matricea Y, fiind cunoscută, se poate rezolva ecuția θ = Φ\Y, unde operatorul “\” din *Matlab* rezolvă ecuația în sensul celor mai mici pătrate (CMMP), reprezentând soluția problemei de regresie liniară. Astfel se poate aproxima o ieșire $\hat{y}$ cât mai apropiată de ieșirea sistemului inițial.

3. Alegerea gradului *m*

Pentru a identifica modelul potrivit este necesară o analiză a erorii medii pătratice (MSE) în funcție de diferite valori ale lui *m*, atât pe datele de identificare (Figura 1), cât și pe datele de validare (Figura 2).

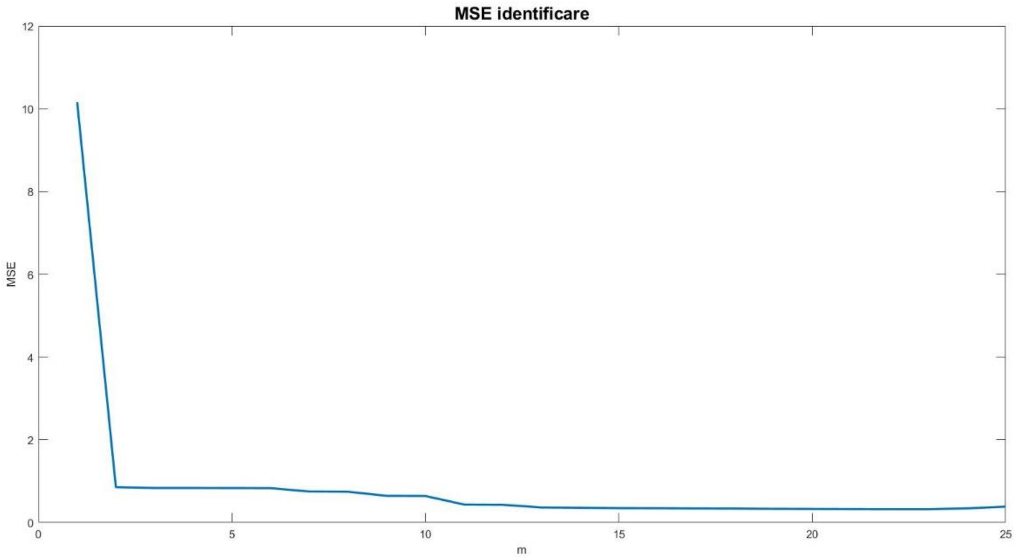

Figura 1

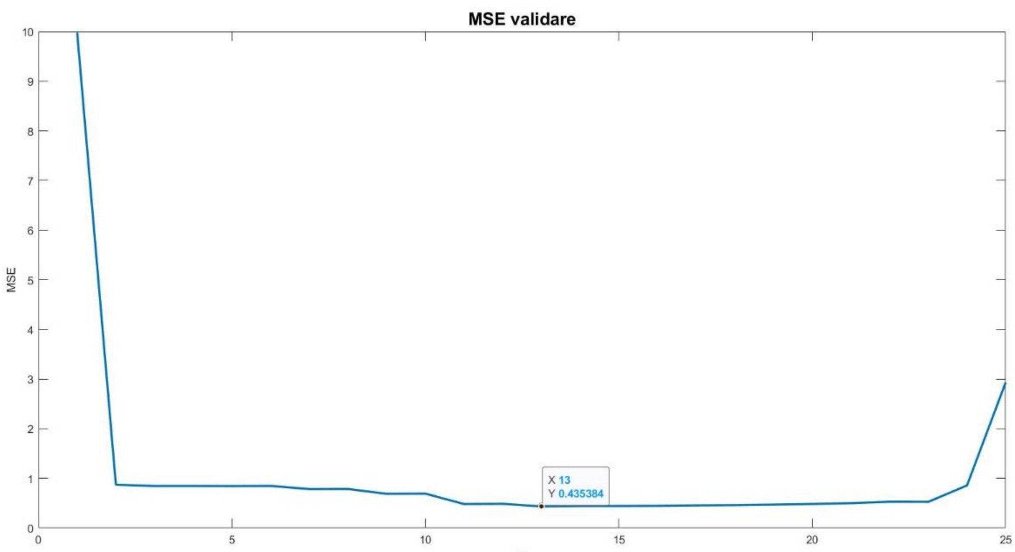

Figura 2

În prima figură se poate observa o scădere a erorii, odată cu creșterea gradului *m*. Problema în identificarea gradului optim apare la testarea modelului pe datele de validare, unde, de la *m* = 13 , eroarea medie pătratică începe să crească, așa cum se poate observa în Figura 2. Acest fenomen este cunoscut sub denumirea de *supraantrenare*.

4. Identificare model pentru grad *m* = 13

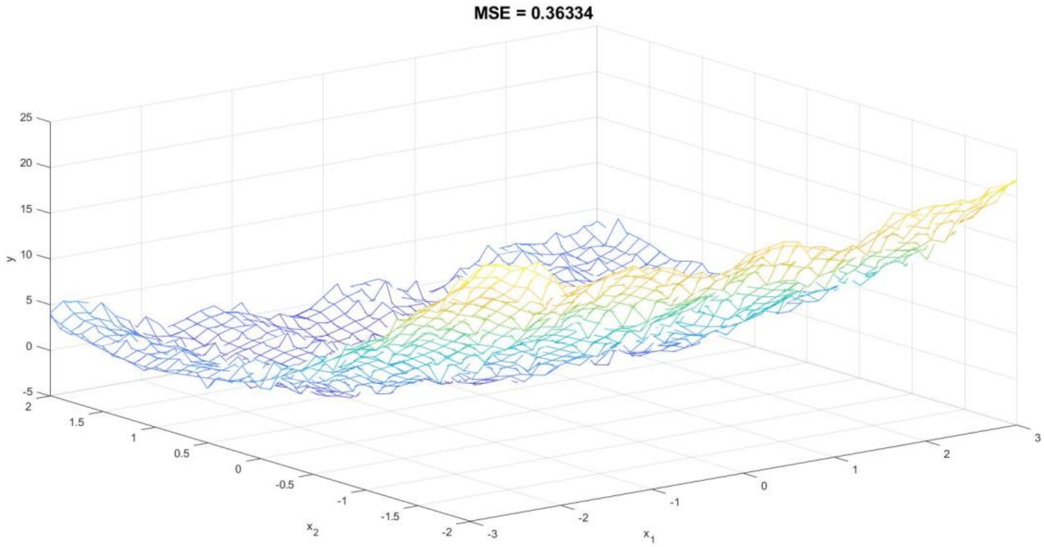

Figura 3

Pentru partea de identificare a modelului, un prim pas, este construirea unei matrice de regresori Φ cu ajutorul căreia se determină vectorul de parametri θ. La pasul următor se calculează o aproximare a ieșirii. În Figura 3 sunt suprapuse ieșirea inițială cu ieșirea aproximată. Modelul reprezentat în imagine este o bună aproximare a sistemului inițial, fapt susținut și de valoarea mică rezultată a lui MSE de 0.36.

5. Validare model pentru grad *m* = 13

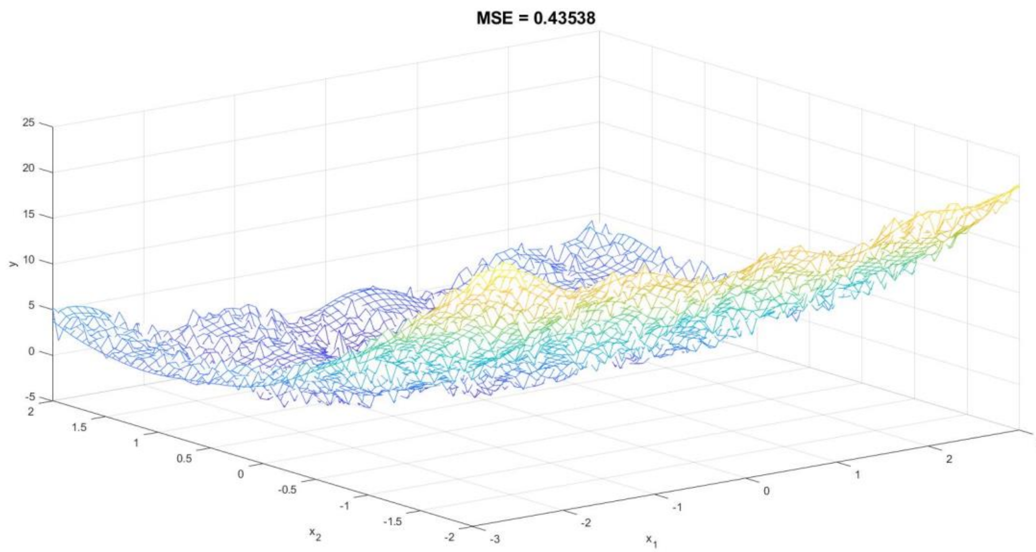

Figura 4

Pentru validarea modelului s-a folosit vectorul de parametri θ determinat anterior și s-a construit o nouă matrice de regresori Φ. Precum în cazul identificării, se calculează o aproximare a ieșirii y care este apoi suprapusă peste ieșirea inițială (Figura 4).

6. Concluzii

În urma testării algoritmului de regresie pentru diferite valori ale lui *m* se poate observa din figurile 1 și 2, o subantrenare pentru un *m* mic și o supraantrenare pentru un *m* mare. Gradul optim corespunzător datelor noastre este m = 13, pentru care am reprezentat graficele de identificare și validare din figurile 3 și 4.

Bibliografie

1. Matlab2020
2. https://busoniu.net/teaching/sysid2021/index_ro.html

Anexă script

```
%% Identificare pentru m = 13
clear; clc; close all;
load('proj_fit_05.mat');
x1 = id.X{1};
x2 = id.X{2};
y = id.Y;
mesh(x1, x2, y);

m = 13;
nr = 0;

% Determinare numar de parametri
for a = 0:m
    for b = 0:m
        if(a+b <= m)
            nr = nr + 1;
        end
    end
end

% Construire matrice de regresori (phi) pentru datele de identificare
row = 1;
col = 1;
phi = zeros(id.dims(1)*id.dims(2), nr); % initializarea matricei phi
for i = 1:id.dims(1)
    for j = 1:id.dims(2) % dimensiunile matricei phi
        for a = 0:m % a reprezinta exponentul coordonatei x1
            for b = 0:m  % b reprezinta exponentul coordonatei x2
                if(a+b <= m) % suma exponetilor este maxim m
                    % fiecare element al matricei phi il calculam ca fiind
                    % produsul dintre cele doua coordonate ale intrarii la
                    % diferite puteri, acestia fiind regresorii
                    phi(row, col) =  x1(i)^a*x2(j)^b;
                    col = col + 1;
                end
            end
        end
        col = 1;
        row = row+1;
    end
end
% Transformam matricea Y intr-o matrice coloana pentru a putea aplica regresia
```

```
yid = y(:);
theta = phi\yid; % aplicam regresia propriu-zisa
y_hatcol = phi*theta; % aproximam iesirea cu ajutorul matricii de regresori
                      % si a vectorului de parametri calculati anterior

% Redimensionam y_hatcol pentru a readuce matricea la forma initiala
y_hatId = zeros(id.dims(1), id.dims(2)); % initializarea matricii de aproximare
for i = 1:id.dims(1)
   % reconstruirea matricii din matrice coloana in matrice de
   % id.dims(1) x id.dims(2)
   y_hatId(:,i) = y_hatcol((i-1)*id.dims(1)+1:i*id.dims(1));
end

% Calculam MSE pe datele de identificare pentru m = 13
MSE = sum((y_hatcol - yid).^2)/length(yid);

% Suprapunem modelul gasit pe datele de identificare cu sistemul real
hold on
mesh(x1, x2, y_hatId);
title(['MSE = ', num2str(MSE)],'FontSize',16);
xlabel('x_1'); ylabel('x_2'); zlabel('y');
hold off
%% Validare pentru m = 13
clc; close all;
x1 = val.X{1};
x2 = val.X{2};
y = val.Y;
mesh(x1, x2, y);

% Construire matrice de regresori (phi) pentru datele de validare
row = 1;
col = 1;
phi = zeros(val.dims(1)*val.dims(2), nr); % initializarea matricei phi

for i = 1:val.dims(1)
   for j = 1:val.dims(2) % dimensiunile matricei phi
      for a = 0:m % a reprezinta exponentul coordonatei x1
         for b = 0:m % b reprezinta exponentul coordonatei x2
            if(a+b <= m) % suma exponetilor este maxim m
               % fiecare element al matricei phi il calculam ca fiind
               % produsul dintre cele doua coordonate ale intrarii la
               % diferite puteri, acestia fiind regresorii
               phi(row, col) =  x1(i)^a*x2(j)^b;
               col = col + 1;
            end
```

```
            end
        end
        col = 1;
        row = row+1;
    end
end

% Transformam matricea Y intr-o matrice coloana pentru a putea aplica regresia
yval = y(:);
y_hatVal = zeros(val.dims(1), val.dims(2)); %initializarea matricii de aproximare

% Folosim matricea de parametri (theta) determinata anterior pe datele de
  % identificare
y_hatcol = phi*theta;  % aproximam iesirea cu ajutorul matricii de regresori
                       % calculata pe datele de validare si a vectorului de
                       % parametri calculat pe datele de identificare

% Redimensionam y_hatcol pentru a readuce matricea la forma initiala
for i = 1:val.dims(1)
    % reconstruirea matricii din matrice coloana in matrice de
    % val.dims(1) x val.dims(2)
    y_hatVal(:,i) = y_hatcol((i-1)*val.dims(1)+1:i*val.dims(1));
end
% Calculam MSE pe datele de validare pentru m = 13
MSE = sum((y_hatcol - yval).^2)/length(yval);

% Suprapunem modelul gasit pe datele de validare cu sistemul real
hold on
mesh(x1, x2, y_hatVal);
title(['MSE = ', num2str(MSE)],'FontSize',16)
xlabel('x_1'); ylabel('x_2'); zlabel('y');
hold off
%% MSE in functie de m
clc; clear; close all;
load('proj_fit_05.mat');

for m = 1:25
 % Determinare numar de parametri
    nr = 0;
    for a = 0:m
        for b = 0:m
            if(a+b <= m)
                nr = nr + 1;
            end
        end
```

```
    end

  % Construire matrice de regresori (phi) pentru datele de identificare
    row = 1;
    col = 1;
    phi = zeros(id.dims(1)*id.dims(2), nr); % initializarea matricei phi
    x1 = id.X{1};
    x2 = id.X{2};
    y = id.Y;
    for i = 1:id.dims(1)
        for j = 1:id.dims(2) % dimensiunile matricei phi
            for a = 0:m % a reprezinta exponentul coordonatei x1
                for b = 0:m % b reprezinta exponentul coordonatei x2
                    if(a+b <= m) % suma exponetilor este maxim m
                        % fiecare element al matricei phi il calculam ca fiind
                        % produsul dintre cele doua coordonate ale intrarii la
                        % diferite puteri, acestia fiind regresorii
                        phi(row, col) =  x1(i)^a*x2(j)^b;
                        col = col + 1;
                    end
                end
            end
            col = 1;
            row = row+1;
        end
    end
  % Transformam matricea Y intr-o matrice coloana pentru a putea aplica regresia
    yid = y(:);
    theta = phi\yid;  % aplicam regresia propriu-zisa
    y_hatcol = phi*theta; % aproximam iesirea cu ajutorul matricii de regresori
                          % si a vectorului de parametri calculati anterior

    y_hatId = zeros(id.dims(1), id.dims(2)); % initializarea matricii de aproximare
    for i = 1:id.dims(1)
        % reconstruirea matricii din matrice coloana in matrice de
        % id.dims(1) x id.dims(2)
        y_hatId(:,i) = y_hatcol((i-1)*id.dims(1)+1:i*id.dims(1));
    end

    % Calculam MSE pe datele de identificare pentru m variabil
    MSEid(m) = sum((y_hatcol - yid).^2)/length(yid);

  % Construire matrice de regresori (phi) pentru datele de validare
    row = 1;
    col = 1;
```

```
    phi = zeros(val.dims(1)*val.dims(2), nr); % initializarea matricei phi
    x1 = val.X{1};
    x2 = val.X{2};
    y = val.Y;

    for i = 1:val.dims(1)
        for j = 1:val.dims(2) % dimensiunile matricei phi
            for a = 0:m % a reprezinta exponentul coordonatei x1
                for b = 0:m % b reprezinta exponentul coordonatei x2
                    if(a+b <= m) % suma exponetilor este maxim m
                        % fiecare element al matricei phi il calculam ca fiind
                        % produsul dintre cele doua coordonate ale intrarii la
                        % diferite puteri, acestia fiind regresorii
                        phi(row, col) =  x1(i)^a*x2(j)^b;
                        col = col + 1;
                    end
                end
            end
            col = 1;
            row = row+1;
        end
    end
 % Transformam matricea Y intr-o matrice coloana pentru a putea aplica regresia
    yval = y(:);
    y_hatcol = phi*theta; % aproximam iesirea cu ajutorul matricii de regresori
                          % calculata pe datele de validare si a vectorului de
                          % parametri calculat pe datele de identificare
    y_hatVal = zeros(val.dims(1), val.dims(2)); %initializarea matricii de aproximare

% Redimensionam matricea y_hatcol pentru a o readuce la forma initiala
    for i = 1:val.dims(1)
        % reconstruirea matricii din matrice coloana in matrice de
        % val.dims(1) x val.dims(2)
        y_hatVal(:,i) = y_hatcol((i-1)*val.dims(1)+1:i*val.dims(1));
    end

 % Calculam MSE pe datele de validare pentru m din intervalul [1,25]
    MSEval(m) = sum((y_hatcol - yval).^2)/length(yval);
end

% Afisam graficul lui MSE din intervalul [1,25] atat pentru datele de
% identificare, cat si pentru datele de validare
figure, plot(MSEid,'LineWidth',2);
xlabel('m'); ylabel('MSE');
title('MSE identificare','FontSize',16);
```

```
figure, plot(MSEval,'LineWidth',2);
title('MSE validare','FontSize',16);
xlabel('m'); ylabel('MSE');
MSEmin = MSEval(1);
mmin = 1;
for i = 2:length(MSEval) % calculul gradului minim
   if(MSEval(i) < MSEmin)
      MSEmin = MSEval(i);
      mmin = i;
   end
end

hold on
plot(mmin, MSEmin, 'r*','LineWidth',2); % evidentierea gradului minim
hold off
```