

Proiect Identificarea Sistemelor

Partea 1. Modelarea unei funcții necunoscute

Balan Andreea, Bilțiu Mariana, Diniș Bianca-Ionela

Grupul 38

Grupa 30136

Specializarea Automatică și Informatică Aplicată, anul 3

Facultatea de Automatică și Calculatoare

Universitatea Tehnică din Cluj-Napoca

2023-2024

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

Bibliografie

Anexă

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

Bibliografie

Anexă

Introducere

- ▶ Ne este atribuit un set de date de tipul intrare-ieșire.
- ▶ Ieșirea sistemului e generată de o funcție (neliniară și statică), afectată de un zgomot (aditiv, Gaussian, de medie zero).
- ▶ Se solicită identificarea sistemului dat și reproducerea acestuia într-un mod care să asigure respingerea zgomotelor, urmată de identificarea celei mai bune aproximări a sistemului.

Obiectiv

- ▶ Dezvoltarea unui model matematic pentru analiza ieșirii sistemului, utilizând un aproximator polinomial.
- ▶ Concret, căutăm valorile optime pentru parametrii matricei θ pentru care $\hat{y} = \phi^T(k)\theta$, $k = \overline{1, n}$, unde n este gradul polinomului, să fie cât mai asemănător cu $y(k)$ din datele atribuite. Astfel, eroarea să fie cât mai mică pentru a putea simula modelul de bază.

Justificare utilizării metodei regresiei liniare

- ▶ Reprezentare matematică simplă și flexibilă pe datele de intrare, respectiv ieșire.
- ▶ Interpretare și înțelegerea relației intrare-ieșire.
- ▶ Ajustarea gradului polinomului în funcție de datele pe care le avem la dispoziție.
- ▶ Antrenarea eficientă a modelului pe baza metodei Regresiei liniare.

Scrierea sistemului sub formă matriceală

$$\begin{bmatrix} y(1,1) \\ y(1,2) \\ \vdots \\ y(N,N) \end{bmatrix} = \begin{bmatrix} \varphi_1(1,1) & \varphi_2(1,1) & \dots & \varphi_n(1,1) \\ \varphi_1(1,2) & \varphi_2(1,2) & \dots & \varphi_n(1,2) \\ \dots & \dots & \dots & \dots \\ \varphi_1(N,N) & \varphi_2(N,N) & \dots & \varphi_n(N,N) \end{bmatrix} \cdot \begin{bmatrix} \theta(1) \\ \theta(2) \\ \vdots \\ \theta(n) \end{bmatrix}$$

,unde $Y \in \mathbb{R}^{N^2}$, $\Phi \in \mathbb{R}^{N \times n}$ și θ reprezintă vectorul de parametri.

Matricea de regresori Φ

- ▶ Fiecare element din matrice reprezintă produsul dintre termenii polinomului ridicați la câte o putere i , respectiv j , unde $i, j \leq N$ și $i + j \leq N$, N reprezintă gradul polinomului.
- ▶ $\sum_{i=0}^N (\sum_{j=0}^N (x_{(1)}^i x_{(2)}^j))$, unde $i + j \leq N$

Matricea Y

- Matricea Y ne este dată din date sub formă de matrice de $N \times N$, iar noi o restructurăm sub formă de matrice coloană.

$$Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1N} \\ y_{21} & y_{22} & \dots & y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \dots & y_{NN} \end{bmatrix} \quad Y' = \begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \\ \vdots \\ y_{NN} \end{bmatrix}$$

Matricea cu parametrii θ

- ▶ $\theta = \Phi \backslash Y$
- ▶ Am folosit operația din MATLAB \backslash (backslash), care alege automat cel mai potrivit algoritm de rezolvare a ecuației.

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \vdots \\ \theta_n \end{bmatrix}$$

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

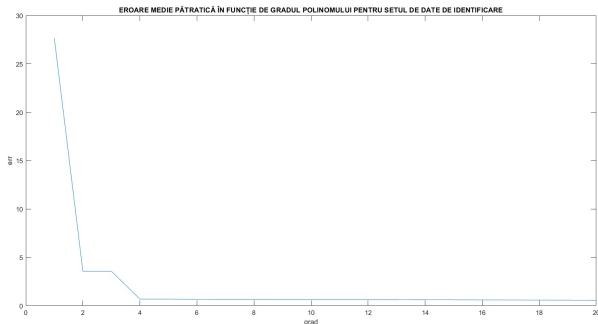
Bibliografie

Anexă

Antrenare model

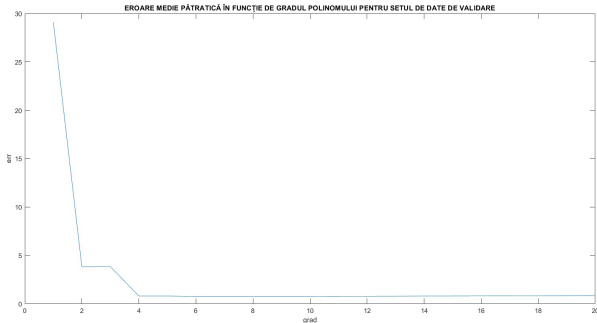
- ▶ Am antrenat modelul până la gradul 20 și am obținut un set de erori atât pentru partea de identificare, cât și pentru partea de validare.
- ▶ Analizând aceste date, am identificat în ce măsură variația în variabila dependentă poate fi explicată de variația variabilei independente.

Eroarea medie pătratică în funcție de gradul polinomului pentru setul de date de identificare



Grad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
err	27.6	3.567	3.56	0.69	0.69	0.64	0.64	0.64	0.64	0.64	0.63	0.62	0.62	0.61	0.6	0.6	0.59	0.59	0.58	0.57

Eroarea medie pătratică în funcție de gradul polinomului pentru setul de date de validare



Grad	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
err	29.08	3.82	3.82	0.79	0.8	0.761	0.766	0.766	0.765	0.765	0.77	0.78	0.78	0.79	0.8	0.81	0.81	0.82	0.82	0.84

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

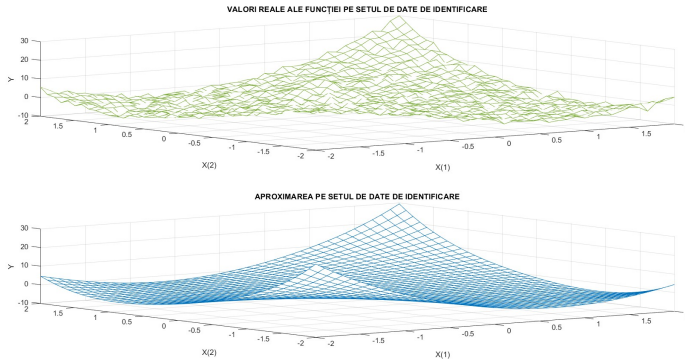
Simulare

Concluzie

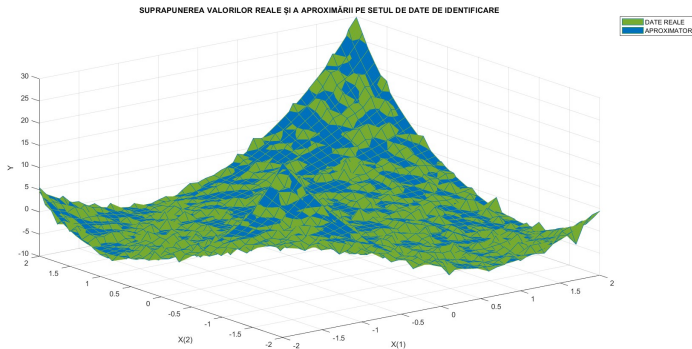
Bibliografie

Anexă

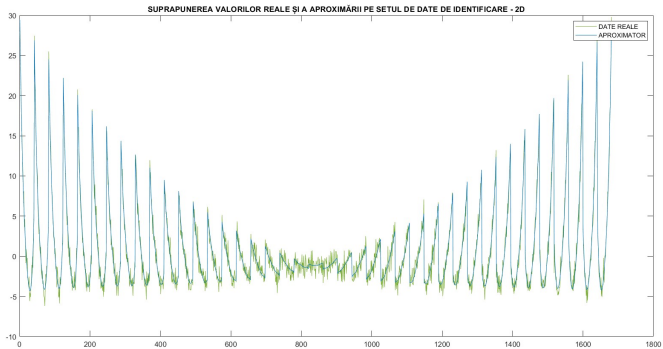
Comparație între setul de date inițial și aproximare pe partea de identificare



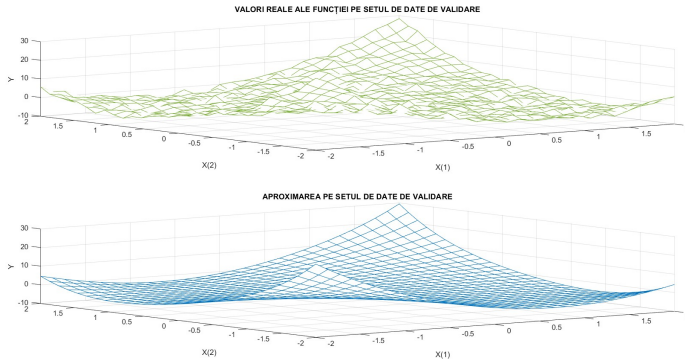
Suprapunere - 3D



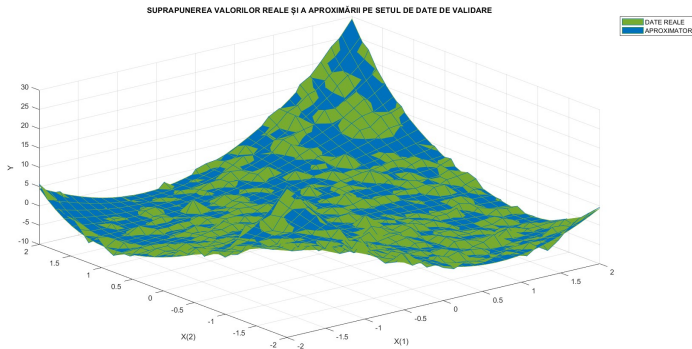
Suprapunere - 2D



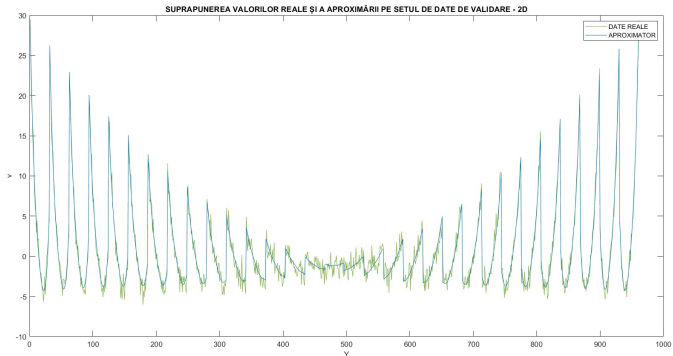
Comparație între setul de date inițial și aproximare pe partea de validare



Suprapunere - 3D



Suprapunere - 2D



Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

Bibliografie

Anexă

Concluzie

- ▶ Un instrument esențial în înțelegerea relațiilor dintre variabile și în realizarea unor predicții utile.
- ▶ Numărul mare de eșantioane din datele de identificare a folosit la generarea unei matrici de regresori care face predicția mai exactă.
- ▶ Numărul de eșantioane nu a depășit un anumit maxim pentru a nu duce la supraantrenare.
- ▶ Gradul a fost ales astfel încât totul să decurgă corect din punct de vedere matematic (operațiile cu matrici să fie realizate corect).

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

Bibliografie

Anexă

Bibliografie

- ▶ BUȘONIU, L. Identificarea Sistemelor 2023-2024

Cuprins

Introducere

Algoritm

Rezultatul reglajelor

Simulare

Concluzie

Bibliografie

Anexă

```
clear all;
load('proj_fit_38.mat');

% VALORILE PE CARE LE FOLOSIM (DATE DE IDENTITATE ȘI VALIDARE)
idx1 = id.X{1,1};
idx2 = id.X{2,1};
valx1 = val.X{1,1};
valx2 = val.X{2,1};

val_config = 20; %GRADUL PÂNĂ LA CARE VREM SĂ CALCUĂM CEA MAI BUNĂ APROXIMARE (CONFIGURABIL)

%SCHIMBĂM FORMA MATRICELOR DE IDENTITATE ȘI VALIDARE ÎN MATRICE COLOANĂ
id_formatat = [];
for i = 1 : length(id.Y(1,:))
    id_formatat = [id_formatat; id.Y(:,i)];
end

val_formatat = [];
for i = 1 : length(val.Y(1,:))
    val_formatat = [val_formatat; val.Y(:,i)];
end

% VECTORI PENTRU STOCAREA ERORILOR DE PE IDENTITATE ȘI VALIDARE A FIECĂRUI GRAD
err_vect_id = zeros(1, val_config);
err_vect_val = zeros(1, val_config);
```

```
%ÎNCEPEM PARCURGEREA PENTRU FIECARE GRAD PÂNĂ LA GRADUL MAXIM DAT DE NOI
for p = 1 : val_config
    phi_id = []; %INIȚIALIZĂM MATRICEA phi_id ȘI ÎNCEPEM CALCULUL EI
    for k = 1:length(idx1) %FOR-uri PENTRU PARCURGEREA ELEM. DIN X1, RESPECTIV X2 (ID)
        for r = 1:length(idx2)
            phi_linie = []; %COMPLETĂM MATRICEA ID CU CÂTE O LINIE
            for i = 0 : p
                for j = 0 : p
                    if i+j <= p
                        phi_linie = [phi_linie, (idx1(k)^i)*(idx2(r)^j)]; %CALCUL ELEM. DE PE FIECARE LINIE
                    end
                end
            end
            phi_id = [phi_id; phi_linie]; %PUNEM ELEM. ÎN MATRICEA phi_id
        end
    end
    teta = phi_id\id_formatat; %TETA
    y_id = phi_id*teta; %Y IDENTITATE
```

```
phi_val = []; %INIȚIALIZĂM MATRICEA phi_val ȘI ÎNCEPEM CALCULUL EI
for k = 1:length(valx1)
    for r = 1:length(valx2)
        phi_linie = [];
        for i = 0 : p
            for j = 0 : p
                if i+j <= p
                    phi_linie = [phi_linie, (valx1(k)^i)*(valx2(r)^j)];
                end
            end
        end
        phi_val = [phi_val; phi_linie];
    end
end
y_val = phi_val*teta; %Y VALIDARE

%CALCULĂM ERORILE ȘI LE STOCĂM ÎN VECTORII CREAȚI PENTRU ERORILE RESPECTIVE
err_id = mean((id_formatat - y_id).^2);
err_val = mean((val_formatat - y_val).^2);
err_vect_id(p) = err_id;
err_vect_val(p) = err_val;
end
```

```
[err_min, m] = min(err_vect_val);      %CAUT EROAREA MINIMĂ ȘI GRADUL POLINOMULUI LA CARE SE OBTINE  
  
%CALCULEZ DATELE PENTRU err_min PRIN ACEEAȘI METODĂ DIN FOR-ul PRINCIPAL  
phi_id = [];  
for k = 1:length(idx1)  
    for r = 1:length(idx2)  
        phi_linie = [];  
        for i = 0 : m  
            for j = 0 : m  
                if i+j <= m  
                    phi_linie = [phi_linie, (idx1(k)^i)*(idx2(r)^j)];  
                end  
            end  
        end  
        phi_id = [phi_id; phi_linie];  
    end  
end  
teta = phi_id\id_formatat;  
y_id = phi_id*teta;
```



```
phi_val = [];  
for k = 1:length(valx1)  
    for r = 1:length(valx2)  
        phi_linie = [];  
        for i = 0 : m  
            for j = 0 : m  
                if i+j <= m  
                    phi_linie = [phi_linie, (valx1(k)^i)*(valx2(r)^j)];  
                end  
            end  
        end  
        phi_val = [phi_val; phi_linie];  
    end  
end  
y_val = phi_val*teta;  
  
%SCHIMBĂM FORMA MATRICELOR DE IDENTITATE ȘI VALIDARE ÎN MATRICE COLOANĂ  
y_val_mat = reshape(y_val, length(valx1), length(valx2));  
y_id_mat = reshape(y_id, length(idx1), length(idx2));
```

```
%AFIȘĂM DATELE DE INTRARE ȘI DE IEȘIRE PENTRU SETUL DE DATE DE IDENTIFICARE
subplot(2, 1, 1);
mesh(idx1, idx2, id.Y, 'EdgeColor', '#77AC30');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("VALORI REALE ALE FUNCȚIEI PE SETUL DE DATE DE IDENTIFICARE");
subplot(2, 1, 2);
mesh(idx1, idx2, y_id_mat, 'EdgeColor', '#0072BD');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("APROXIMAREA PE SETUL DE DATE DE IDENTIFICARE");

%FACEM SUPRAPUNEREA LOR PENTRU A OBSERVA DACĂ E CORECTĂ APROXIMAREA
figure;
mesh(idx1, idx2, id.Y, 'FaceColor', '#77AC30', 'EdgeColor', '#0072BD');
hold on;
mesh(idx1, idx2, y_id_mat, 'FaceColor', '#0072BD', 'EdgeColor', '#77AC30');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("SUPRAPUNEREA VALORILOR REALE ȘI A APROXIMĂRII PE SETUL DE DATE DE IDENTIFICARE");
legend('DATE REALE', 'APROXIMATOR');
```

```
figure;
%%AFIȘĂM DATELE DE INTRARE ȘI DE IEȘIRE PENTRU SETUL DE DATE DE VALIDARE
subplot(2, 1, 1);
mesh(valx1, valx2, val.Y, 'EdgeColor', '#77AC30');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("VALORI REALE ALE FUNCȚIEI PE SETUL DE DATE DE VALIDARE");
subplot(2, 1, 2);
mesh(valx1, valx2, y_val_mat, 'EdgeColor', '#0072BD');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("APROXIMAREA PE SETUL DE DATE DE VALIDARE");

figure;
mesh(valx1, valx2, val.Y, 'FaceColor', '#77AC30', 'EdgeColor', '#0072BD');
hold on;
mesh(valx1, valx2, y_val_mat, 'FaceColor', '#0072BD', 'EdgeColor', '#77AC30');
xlabel('X(1)');
ylabel('X(2)');
zlabel('Y');
title("SUPRAPUNEREA VALORILOR REALE ȘI A APROXIMĂRII PE SETUL DE DATE DE VALIDARE");
legend('DATE REALE', 'APROXIMATOR');
```

```
% AFIȘĂM ÎN PLAN 2D PENTRU O OBSERVAȚIE MAI BUNĂ
%VALIDARE
plot(val_formatat, 'Color', '#77AC30'); %MATRICEA COLOANĂ PENTRU DATE REALE
hold on;
plot(y_val, 'Color', '#0072BD'); %MATRICEA COLOANĂ PENTRU APROXIMATOR
title('SUPRAPUNEREA VALORILOR REALE ȘI A APROXIMĂRII PE SETUL DE DATE DE VALIDARE - 2D');
legend('DATE REALE', 'APROXIMATOR');
xlabel('Y');
ylabel('v');
figure;

%IDENTIFICARE
plot(id_formatat, 'Color', '#77AC30'); %MATRICEA COLOANĂ PENTRU DATE REALE
hold on;
plot(y_id, 'Color', '#0072BD'); %MATRICEA COLOANĂ PENTRU APROXIMATOR
title('SUPRAPUNEREA VALORILOR REALE ȘI A APROXIMĂRII PE SETUL DE DATE DE IDENTIFICARE - 2D');
legend('DATE REALE', 'APROXIMATOR');
```

```
t=1:val_config; % VECTORUL t CARE CONȚINE FIECARE GRAD CALCULAT ÎN FUNCȚIE DE vav_config
figure;
%EROAREA MEDIE PĂTRATICĂ PE SETUL DE DATE DE IDENTIFICARE
plot(t, err_vect_id);
xlabel('grad');
ylabel('err');
title('EROARE MEDIE PĂTRATICĂ ÎN FUNCȚIE DE GRADUL POLINOMULUI PENTRU SETUL DE DATE DE IDENTIFICARE');
figure;

%EROAREA MEDIE PĂTRATICĂ PE SETUL DE DATE DE VALIDARE
plot(t, err_vect_val);
xlabel('grad');
ylabel('err');
title('EROARE MEDIE PĂTRATICĂ ÎN FUNCȚIE DE GRADUL POLINOMULUI PENTRU SETUL DE DATE DE VALIDARE');
```