Proiect

Identificarea sistemelor

Băra Bogdan-Alin

Anul 3

Grupa 30131/1

Profesori indrumator

Prof. Dr. Ing. Lucian Bușoniu

Prof. Asist. Vicu Mihalis-Maer

Tema proiectului: Modelarea unei funcții necunoscute

Raport de activitate

- Introducere

Modelarea unei funcții necunoscute se realizează cu ajutorul unei aproximator polinomial, care va aproxima cu o eroare cat mai mica posibila curba a polinomului.

În problema noastră se dă un set de date de intrare[x(1),x(2)]/ieșire[y], atât pentru identificare cât și pentru validare, ieșirea fiind afectată de zgomot.

Va trebui sa găsim "m-ul"(gradul polinomului necunoscut) configurabil, astfel încât aproximatorul polinomial să fie cât mai precis, vectorul optim Ɵ cât mai apropiat de valorile de ieșire Y, astfel formând funcția necunoscută aproximată minimal.

```
load('proj_fit_28.mat')
x1=id.X{1};
x2=id.X{2};
y=id.Y;
```

```
x3=val.X{1};
x4=val.X{2};
y2=val.Y;
```

Fig. 1 Datele de intrare

- Aproximatorul

Aproximatorul este reprezentat de m-ul care este ajustabil și de către parametrii de intrare x1, x2, acesta fiind identificat prin intermediul funcției necunoscute care urmează sa fie aflată. De asemenea, aproximatorul este compus din regresori, prin intermediul cărora în final se vor încerca să se realizeze anumite predicții.

```
for m=1:7
PHI=[];
for i=1:N1
    for j=1:N2
        z1=[];
        for f=1:m
            for b=1:m
                z=(x1(i).^(f-1))*(x2(j).^(b-1));
                z1=[z1 z];
            end
        end
        PHI=[PHI;z1];
    end
end
```

Fig. 2 Structura aproximatorului

- Gasirea parametrilor Φ (Matricea Φ)

Am declarat matricea Φ ca și o matrice nulă. Apoi am parcurs lungimea celor doi vectori; id.X{1} și id.X{2} prin intermediul a două for-uri. Astfel, parametrii X{1} și X{2} ajung în combinație în toate variantele posibile. Ulterior am inițializat o nouă matrice, z1 ca și o matrice nulă, această realizând creearea fiecarei linii din matrice. Am folosit din nou două for-uri pentru realizarea tuturor combinațiilor de puteri posibile(f, respectiv b). Acest lucru a fost posibil prin intermediul variabilei “m”, care în cazul nostru este reprezentat de un vector cu șase poziții.

Luăm fiecare poziție posibilă, pe care ulterior o vom adăuga în variabila z. Pentru determinarea puteriilor, ne folosim de variabilele f si b. Ulterior poziția este adăugată în următoarea coloană din matricea z1. După se reia procesul pentru realizarea matricei Φ.

- Yhat

Pentru a determina matricea Y, vom transpune matricea linie y. Acum că știm atât matricea Φ cât și matricea Y, putem afla matricea Θ, care este realizată prin împărțirea celor două matrici Φ\ Y. Dupa aceasta, se află un nou Y1 (Yhat) prin înmulțirea matriciilor Φ* Θ.

```
Y=y(:);
Teta=PHI\Y;
Y1=PHI*Teta;
Y2=reshape(Y1,[41,41]);
```

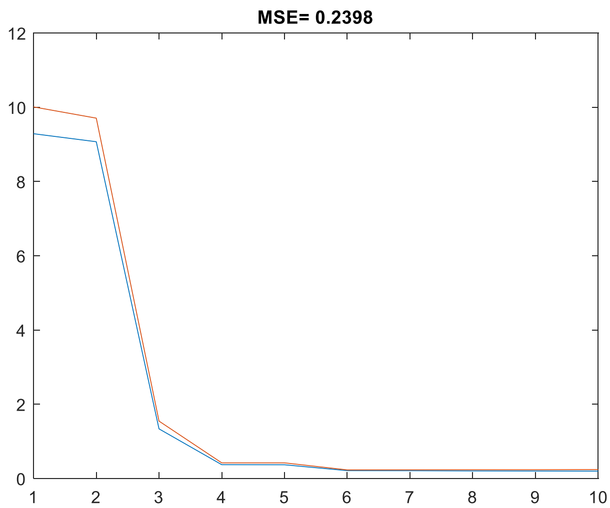
Fig. 3 Yhat (Y aproximat)

- MSE(Mean Squared Error)

Pentru realizarea Mean Sqaured Error-ului, am aplicat un for unde am calculat “m” erori pe care ulterior le-am stocat într-un vector a3 de lungime m.

În graficul reprezentat mai jos(Fig. 4) în care am ales m = 10, putem observa că MSE-ul ajunge la o valoare minimă la valoarea de 6. De la respectiva valoare, MSE-ul crește ușor, de unde rezultă că valoarea optimă a MSE-ului este în punctul 6. De asemenea, putem observa ca MSE-ul realizat pentru identitate este deasupra liniei de validare.

a)



b)

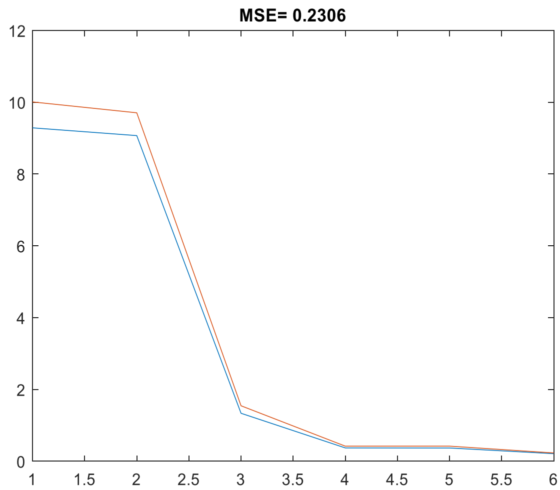


Fig. 4 Graficele pentru validare(albastru) și identificare(roșu) pentru

a) m=10 b) m=6

- Datele de validare

```
x3=val.X{1};
x4=val.X{2};
y2=val.Y;

N3=length(x3);
N4=length(x4);
PHI1=[];
for i=1:N3
    for j=1:N4
        z3=[];
        for f1=1:m
            for b1=1:m
                z2=(x3(i).^(f1-1))*(x4(j).^(b1-1));
                z3=[z3 z2];
            end
        end
        PHI1=[PHI1;z3];
end
end

Y3=PHI1*Teta;
Y4=reshape(Y3,[31,31]);
```

```
a2=0;
for  s1=1:(N3*N4)
a2=a2+(y2(s1)-Y4(s1)).^2;
end
a3(m)=(1/(N3*N4))*a2;
c1=round(a3(m),4);

end
```

Fig. 5 Validarea

Pentru datele de validare am ales valori la lungimi diferite, iar procesul de creeare a matricii și MSE-ului se aplică la fel ca la identificare. După ce am realizat noua matrice Φ pentru validare, am aflat o nouă matrice linie Y3{Yhat validare} prin intermediul Φ_validare și a aceluiaș Θ. Asemeni MSE-ului pentru id am calculat MSE-ul pentru validare.

- Grafice și ploturi

În imagine sunt 2 mesh-uri suprapuse, cu ajutorul lui hold on. Mesh-ul format din x3, x4 și y2, reprezintă datele de intrare și ieșire ale validarilor, iar al 2-lea mesh este format din datele de intrare ale validării și Y4, care reprezintă Yhat calculat la final pentru validare.

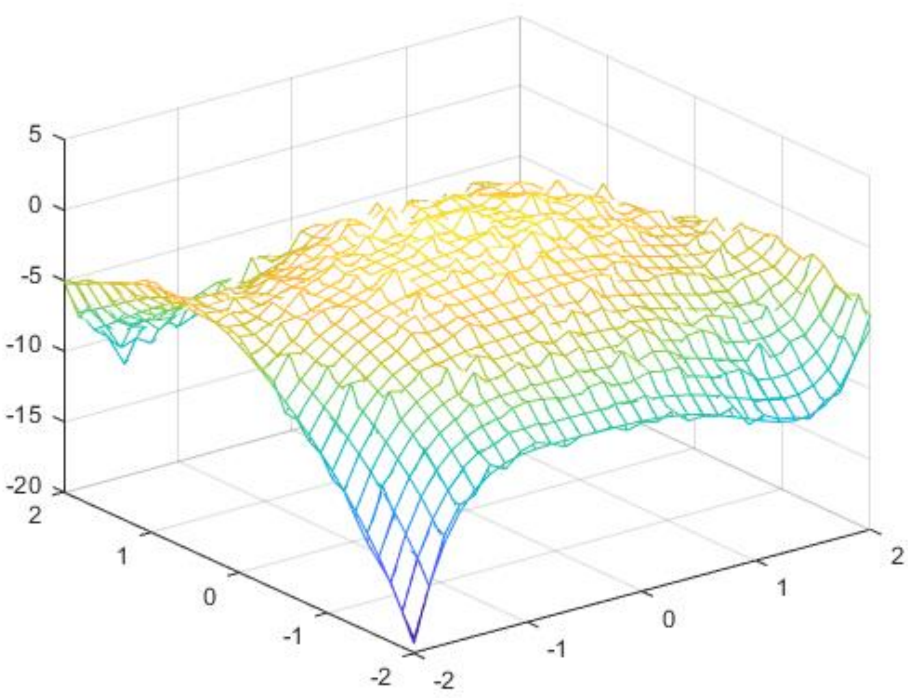


Fig.6 Mesh-ul pentru valoarea m=6

- Discuție finală și concluzia

În concluzie prin folosirea aproximatorului și comparând valorile de intrare cu cele de ieșire, am putut ajunge la concluzia că valoarea cea mai optimă este la m=6, acest lucru reieșind din graficul de MSE. Respectivul aproximator este format din regresori și este implementat pentru o funcție necunoscută. Rezultatele acestei implementări sunt reprezentate de 2 grafice, suprapuse în figura 6, unde se poate observă ca funcția de ieșire este afectată de zgomot, iar funcția Yhat nu este afectată.

Anexă

```
load('proj_fit_28.mat')
x1=id.X{1};
x2=id.X{2};
y=id.Y;
for m=1:6

N1=length(x1);
N2=length(x2);

PHI=[];
for i=1:N1
    for j=1:N2
        z1=[];
        for f=1:m
            for b=1:m
                z=(x1(i).^(f-1))*(x2(j).^(b-1));
                z1=[z1 z];
            end
        end
        PHI=[PHI;z1];
    end
end

Y=y(:);
Teta=PHI\Y;
Y1=PHI*Teta;
Y2=reshape(Y1,[41,41]);


a=0;
for  s=1:(N1*N2)
a=a+(y(s)-Y2(s)).^2;
end
a1(m)=(1/(N1*N2))*a;
c=round(a1(m),4);


x3=val.X{1};
x4=val.X{2};
y2=val.Y;

N3=length(x3);
N4=length(x4);
PHI1=[];
for i=1:N3
    for j=1:N4
        z3=[];
        for f1=1:m
            for b1=1:m
```

```
                z2=(x3(i).^(f1-1))*(x4(j).^(b1-1));
                z3=[z3 z2];
            end
        end
        PHI1=[PHI1;z3];
end
end

Y3=PHI1*Teta;
Y4=reshape(Y3,[31,31]);

a2=0;
for  s1=1:(N3*N4)
a2=a2+(y2(s1)-Y4(s1)).^2;
end
a3(m)=(1/(N3*N4))*a2;
c1=round(a3(m),4);

end


figure(2);
mesh(x3,x4,y2);
hold on;
mesh(x3,x4,Y4);


%%
figure(1);
plot(a1);
title("MSE= " + c);
hold on;
plot(a3);
title("MSE= " + c1);
```