CENG 235 ALGORİTMALARLA SAYISAL ÇÖZÜMLEME

Prof. Dr. Tufan TURACI
tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 14

14. Hafta Konular

- **Regresyon Analizi:**

  --- En Küçük Kareler Yöntemi

  --- Lineer Hale Dönüştürülebilen Modeller

Regresyon Analizi
En Küçük Kareler Yöntemi

Deneysel uygulamalar sonucu elde edilen veriler incelenir ve elde edilen verileri modelleyen bir fonksiyon bulma amaçlanır. Çoğu zaman bu veri tablosuna tam olarak uyan bir fonksiyon bulmak mümkün olmayabilir; veri tablosuna en iyi uyan fonksiyon belirlenmeye çalışılır. Bir veri tablosuna en iyi uyan fonksiyonu bulma sürecine ***regresyon analizi*** denir.

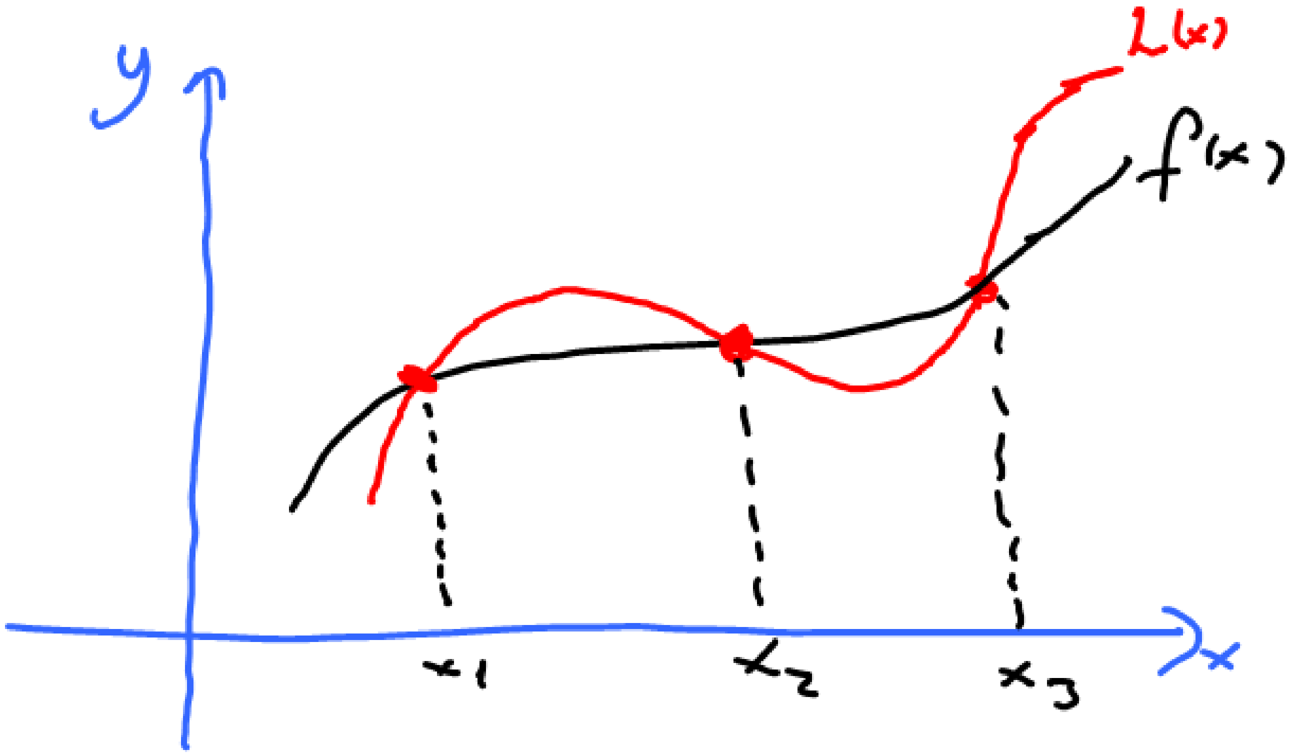
Regresyon analizi yaparken en çok kullanılan yöntemlerden biri ***en küçük kareler yöntemidir***. En küçük kareler yöntemi, tıp, finans, mühendislik, ziraat, biyoloji ve sosyoloji gibi çeşitli bilim dallarında çeşitli değişkenler arasındaki ilişkiler belirlenirken kullanılan en önemli yöntemlerden biridir.

x	x_1	x_2	- - - -	x_n
y	y_1	y_2	- - - - -	y_n

Bu noktalardan geçen en uygun $f(x)$?

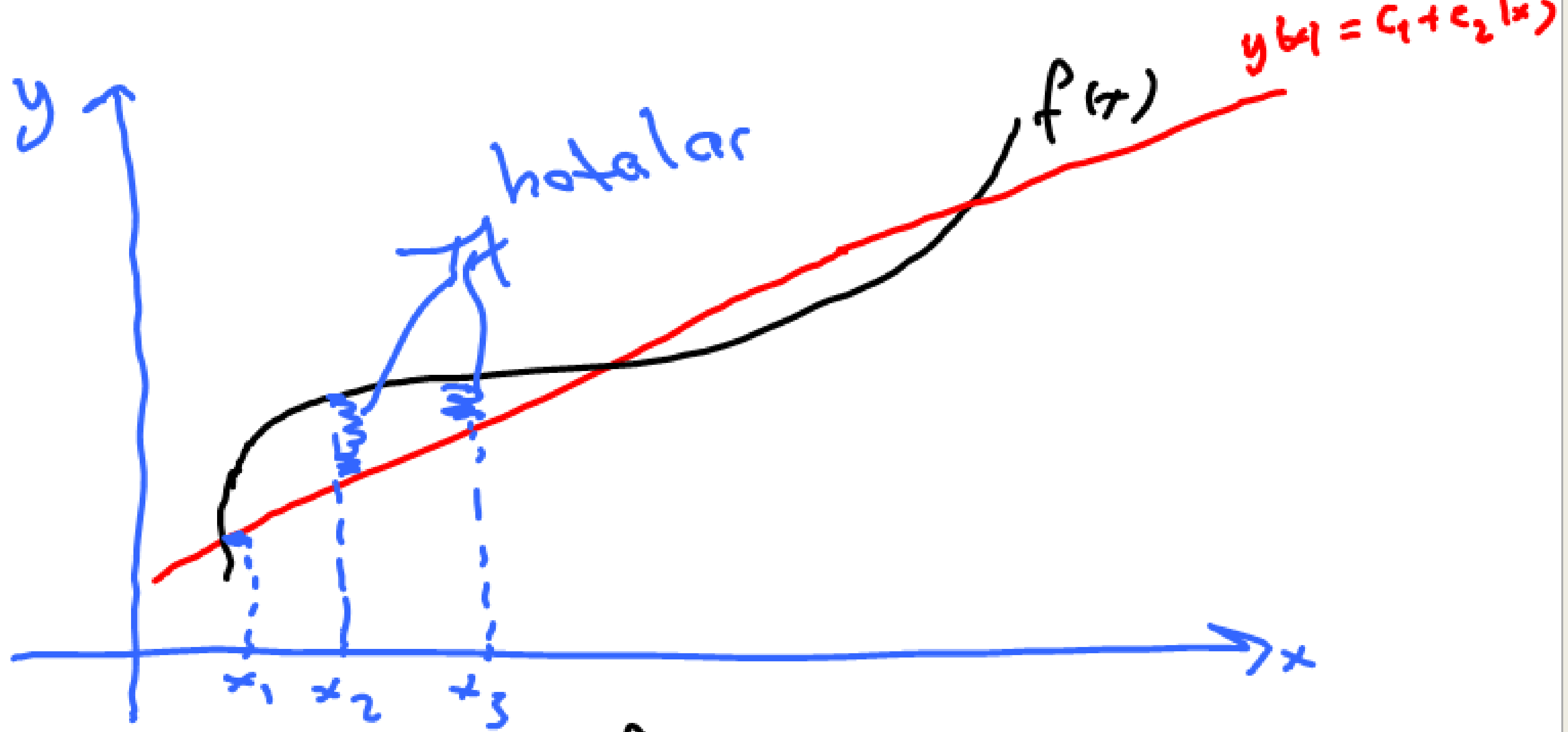
- $f(x)$ çok karmaşık bir fonksiyon ise daha basit bir fonksiyon bulunur.

* İnterpolasyonda $y_i = f(x_i)$ olmak zorunda fakat en küçük kareler yönteminde de bu noktalardan geçmek zorunluluğu yoktur.



$f(x) \rightarrow$ gerçek fonksiyon

$L(x) \rightarrow$ interpolasyon polinomu



$f(x) \rightarrow$ gerçek fonksiyon

$y(x) \Rightarrow$ en küçük kareler yöntemi ile elde edilen yaklaşık fonksiyon

Eğri uydurma $y = F(x, c_0, c_1, \ldots, c_n)$

şeklindedir. (c_i'ler parametrelerdir.)

- parametrelere göre lineer bir bağıntı ile bağımlı değişkene bağlı modellerin söz konusu olduğu durumlarda en uygun eğrinin araştırılması problemine lineer regresyon denir.

$$y = c_0 \cdot \varphi_0(x) + c_1 \cdot \varphi_1(x) + \cdots + c_n \cdot \varphi_n(x)$$

$$\varphi_i(x) = x^i \quad (\text{polinom})$$

$$\varphi_i(x) = e^{ix}$$

$$\varphi_i(x) = \sin(ix)$$

$$\varphi_i(x) = \cos(ix)$$

$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ eğrik noktalar

$$f(x_k) = y_k + e_k \quad , k = 1, 2, \ldots, N$$

$$e_k = f(x_k) - y_k \quad \text{(Hata)}$$

Maksimum hata $E_\infty(f) = \max_{1 \le k \le N} |f(x_k) - y_k|$

Ortalama hata $E_1(f) = \frac{1}{N} \sum_{k=1}^{N} |f(x_k) - y_k|$

Ortalama kare hatanın karekökü:

$$E_2(f) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (f(x_k) - y_k)^2}$$

En iyi sonucu veren hata formülü.

En uygun eğri $y = a + bx$ ise:

$$y_i = a + b \cdot x_i + e_i \quad , \quad i = 1, 2, \ldots, N$$

$$e_i = y_i - a - bx_i$$

$$\sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (y_i - a - bx_i)^2 = E(a,b)$$

bu değeri minimum yapan a ve b değerleri nedir?

1

$$\frac{\partial E}{\partial a} = 0$$

2

$$\frac{\partial E}{\partial b} = 0$$

(Ekstremum nokta koşulu)

$$1)\quad 2\cdot\sum_{i=1}^{N}(y_i - a - bx_i)(-1) = 0$$

$$a\cdot\sum_{i=1}^{N}1 + b\cdot\sum_{i=1}^{N}x_i = \sum_{i=1}^{N}y_i$$

$$2)\quad 2\cdot\sum_{i=1}^{N}(y_i - a - bx_i)\cdot(-x_i) = 0$$

$$a\cdot\sum_{i=1}^{N}x_i + b\cdot\sum_{i=1}^{N}x_i^2 = \sum_{i=1}^{N}x_i\cdot y_i$$

Bu denklemler aşağıdaki şekilde ifade edilirse:

$$\begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i \cdot y_i \end{bmatrix}$$

↳ a ve b ?

2. dereceden bir polinom uydurmak için:

$$y = a_0 + a_1 x + a_2 x^2 \Rightarrow \begin{matrix} a_0 = ? \\ a_1 = ? \\ a_2 = ? \end{matrix}$$

Matrislerle gösterimi:

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{bmatrix}$$

Çalışma Sorusu olarak bırakılmıştır.

k. dereceden en uygun polinom!

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

$$a_0, a_1, a_2, \ldots, a_k \; ?$$

$$\begin{bmatrix} N & \sum x_i & \cdots & \sum x_i^k \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{k+1} \\ \vdots & \vdots & & \vdots \\ \sum x_i^k & \sum x_i^{k+1} & \cdots & \sum x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x_i^k \end{bmatrix}$$

Örnek:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_i	-1	0	1	2	3	4	5	6
y_i	10	9	7	5	4	3	0	-1
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8

noktalarına $y = a + b.x$ tipinde öyle bir doğru bulunuz ki hataların kareleri toplamı minimum olsun

(Bu verilere uyan en küçük kareler anlamındaki doğru nedir?)

$$a.\sum_{i=1}^{N=8} 1 + b.\sum_{i=1}^{N=8} x_i = \sum_{i=1}^{N=8} y_i$$

$$a.\sum_{i=1}^{N=8} x_i + b.\sum_{i=1}^{N=8} x_i^2 = \sum_{i=1}^{N=8} x_i . y_i$$

x	y	x_i^2	$y_i \cdot x_i$
-1	10	1	-10
0	9	0	0
1	7	1	7
2	5	4	10
3	4	9	12
4	3	16	12
5	0	25	0
6	-1	36	-6
$\sum$ 20	37	92	25

$$a\cdot\overbrace{\sum_{i=1}^{N=8} 1}^{8} + b\cdot\overbrace{\sum_{i=1}^{N=8} x_i}^{20} = \overbrace{\sum_{i=1}^{N=8} y_i}^{37}$$

$$a\cdot\underbrace{\sum_{i=1}^{N=8} x_i}_{20} + b\cdot\underbrace{\sum_{i=1}^{N=8} x_i^2}_{92} = \underbrace{\sum_{i=1}^{N=8} x_i\cdot y_i}_{25}$$

Böylece:

$$8.a + 20.b = 37$$

$$20.a + 92.b = 25$$

elde edilir.

$$-5 / \quad 8.a + 20.b = 37$$

$$2 / \quad 20a + 92.b = 25$$

$$-\cancel{40a} - 100b = -185$$

$$\cancel{40a} + 184b = 50$$

$$84b = -135$$

$$\boxed{b = -1.607143}$$

1. denklemde yerine yazalım:

$$8 \cdot a + 20 \cdot (-1.607143) = 37$$

$$\boxed{a = 8.642858}$$

Denklem: $y = 8.642858 - (1.607143) \cdot x$

(Doğru denklemi)

C Kodu:

```c
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <locale.h>
float determinant(float M[2][2]);
void matrisyaz(float N[2][2]);

int main()
{   setlocale(LC_ALL,"Turkish");
    int m,n,i,j,t,s,topx=0,topy=0,topx2=0,topyx=0,detA;
    printf("Kaç adet nokta olduğunu giriniz--->");
       scanf("%d",&n);
    float X[n],Y[n],A[2][2],B[2],tempA[2][2],c[2];

    for(i=0;i<n;i++)
    { printf("%d. noktadaki x değerini giriniz: ",i+1);
         scanf("%f",&X[i]);
      printf("%d. noktadaki y değerini giriniz: ",i+1);
         scanf("%f",&Y[i]);
      }
```

```
for(i=0;i<n;i++)
 {topx=topx+X[i];
   topy=topy+Y[i];
   topx2=topx2+(X[i]*X[i]);
   topyx=topyx+(X[i]*Y[i]);
  }
  A[0][0]=n; A[0][1]=topx;
  A[1][0]=topx; A[1][1]=topx2;
  B[0]=topy; B[1]=topyx;

  printf("Lineer Denklem Sistemi Cramer Yöntemiyle Çözülmüştür...\n");
  printf("Katsayılar matrisi:");
  matrisyaz(A);
  printf("B sonuç matrisi:\n");
  for(i=0;i<2;i++)
  printf("%f\n",B[i]);
  printf("\n");
```

```
detA=determinant(A);
for(i=0;i<2;i++)
{
   for(t=0;t<2;t++)
    {
      for(s=0;s<2;s++)
       { tempA[t][s]=A[t][s];}
      }

    for(j=0;j<2;j++)
   {tempA[j][i]= B[j]; }

  c[i]=determinant(tempA)/detA;
  printf("%d. sütun elemanları ile B matrisinin elemanları yer değiştirdi:",i+1);
  matrisyaz(tempA);
 }
```

```
printf("Sonuclar\n");
for(i=0;i<2;i++)
    {printf("c[%d]=%f\n",i+1,c[i]);}

 printf("\nElde Edilen Birinci Derecen Lineer Regresyon Denklemi (Doğru Denklemi):\n");
 printf("y=%f + (%f).x\n",c[0],c[1]);
 getch();
 return 0;
}
```

```
float determinant(float M[2][2])
{   float det;
    det=(M[0][0]*M[1][1])-(M[0][1]*M[1][0]);
    return det;
}
```

```
void matrisyaz(float N[2][2])
 {
      int i,j;
       for(i=0;i<2;i++)
       {
           printf("\n");
           for(j=0;j<2;j++)
             printf("%f\t",N[i][j]);
       }
       printf("\n\n");
  }
```

Ekran Çıktısı:

```
Kaç adet nokta olduğunu giriniz--->8
1. noktadaki x değerini giriniz: -1
1. noktadaki y değerini giriniz: 10
2. noktadaki x değerini giriniz: 0
2. noktadaki y değerini giriniz: 9
3. noktadaki x değerini giriniz: 1
3. noktadaki y değerini giriniz: 7
4. noktadaki x değerini giriniz: 2
4. noktadaki y değerini giriniz: 5
5. noktadaki x değerini giriniz: 3
5. noktadaki y değerini giriniz: 4
6. noktadaki x değerini giriniz: 4
6. noktadaki y değerini giriniz: 3
7. noktadaki x değerini giriniz: 5
7. noktadaki y değerini giriniz: 0
8. noktadaki x değerini giriniz: 6
8. noktadaki y değerini giriniz: -1
```

```
Lineer Denklem Sistemi Cramer Yöntemiyle Çözülmüştür...
Katsayılar matrisi:
8,000000        20,000000
20,000000       92,000000

B sonuç matrisi:
37,000000
25,000000

1. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
37,000000       20,000000
25,000000       92,000000

2. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
8,000000        37,000000
20,000000       25,000000

Sonuclar
c[1]=8,642858
c[2]=-1,607143

Elde Edilen Birinci Derecen Lineer Regresyon Denklemi (Doğru Denklemi):
y=8,642858 + (-1,607143).x
```

Örnek! Yapılan bir gözlem sonucunda elde edilen (x_i, y_i) değerleri aşağıda verilmiştir.

x_i	0	2	3	4
y_i	1	2	3	5

1-) Bu gözlemlere uygun en küçük kareler anlamında doğru? $(y = c_1 + c_2 . x)$

2-) Bu gözlemlere uygun en küçük kareler anlamında parabol? $(y = c_1 + c_2 . x + c_3 . x^2)$

$$E = \sum_{i=1}^{4} (c_1 + c_2 x_i - y_i)^2$$

$$\frac{\partial E}{\partial c_1} = 2 \cdot \sum_{i=1}^{4} (c_1 + c_2 x_i - y_i) = 0$$

$$c_1 \cdot \sum_{i=1}^{4} 1 + c_2 \sum_{i=1}^{4} x_i = \sum_{i=1}^{4} y_i$$

$$\frac{\partial E}{\partial c_2} = 2 \cdot \sum_{i=1}^{4} x_i \cdot (c_1 + c_2 x_i - y_i) = 0$$

$$c_1 \cdot \sum_{i=1}^{4} x_i + c_2 \cdot \sum_{i=1}^{4} x_i^2 = \sum_{i=1}^{4} y_i \cdot x_i$$

$$\sum_{i=1}^{4} x_i = 9 \qquad \sum_{i=1}^{4} y_i = 11 \qquad \sum_{i=1}^{4} x_i^2 = 29 \qquad \sum_{i=1}^{4} y_i x_i = 33$$

Böylece:

$$4c_1 + 9c_2 = 11$$
$$9c_1 + 29c_2 = 33$$

$$c_1 = \frac{22}{35} \qquad c_2 = \frac{33}{35}$$

$$c_1 = 0.628571 \qquad c_2 = 0.942857$$

$$y = 0.628571 + (0.942857) \cdot x$$

Program Çalıştırıldıktan Sonra Ekran Çıktısı:

```
Kaç adet nokta olduğunu giriniz--->4
1. noktadaki x değerini giriniz: 0
1. noktadaki y değerini giriniz: 1
2. noktadaki x değerini giriniz: 2
2. noktadaki y değerini giriniz: 2
3. noktadaki x değerini giriniz: 3
3. noktadaki y değerini giriniz: 3
4. noktadaki x değerini giriniz: 4
4. noktadaki y değerini giriniz: 5
```

```
Lineer Denklem Sistemi Cramer Yöntemiyle Çözülmüştür...
Katsayılar matrisi:
4,000000        9,000000
9,000000        29,000000

B sonuç matrisi:
11,000000
33,000000

1. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
11,000000       9,000000
33,000000       29,000000

2. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
4,000000        11,000000
9,000000        33,000000

Sonuclar
c[1]=0,628571
c[2]=0,942857

Elde Edilen Birinci Derecen Lineer Regresyon Denklemi (Doğru Denklemi):
y=0,628571 + (0,942857).x
```

2-) $E = \sum_{i=1}^{4} [c_1 + c_2 x_i + c_3 x_i^2 - y_i]^2 = 0$

$$\frac{\partial E}{\partial c_1} = \sum_{i=1}^{4} 2[c_1 + c_2 x_i + c_3 x_i^2 - y_i].(1) = 0$$

$$c_1 . \sum_{i=1}^{4} 1 + c_2 . \sum_{i=1}^{4} x_i + c_3 . \sum_{i=1}^{4} x_i^2 = \sum_{i=1}^{4} y_i$$

$$\frac{\partial E}{\partial c_2} = \sum_{i=1}^{4} 2.[c_1 + c_2 x_i + c_3 x_i^2 - y_i].(x_i) = 0$$

$$c_1 . \sum_{i=1}^{4} x_i + c_2 . \sum_{i=1}^{4} x_i^2 + c_3 . \sum_{i=1}^{4} x_i^3 = \sum_{i=1}^{4} y_i . x_i$$

$$\frac{\partial E}{\partial c_3} = \sum_{i=1}^{4} 2.\left[c_1 + c_2 x_i + c_3 x_i^2 - y_i\right].(x_i^2) = 0$$

$$c_1.\sum_{i=1}^{4} x_i^2 + c_2.\sum_{i=1}^{4} x_i^3 + c_3.\sum_{i=1}^{4} x_i^4 = \sum_{i=1}^{4} y_i.x_i^2$$

$$\sum_{i=1}^{4} x_i = 9 \qquad \sum_{i=1}^{4} y_i = 11 \qquad \sum_{i=1}^{4} x_i^2 = 29 \qquad \sum_{i=1}^{4} y_i.x_i = 33$$

$$\sum_{i=1}^{4} x_i^3 = 99 \qquad \sum_{i=1}^{4} x_i^4 = 353 \qquad \sum_{i=1}^{4} y_i.x_i^2 = 115$$

Böylece:

$$4c_1 + 9c_2 + 29c_3 = 11$$
$$9c_1 + 29c_2 + 99c_3 = 33$$
$$29c_1 + 99c_2 + 353c_3 = 115$$

$$c_1 = \frac{56}{55} \qquad c_2 = \frac{-6}{55} \qquad c_3 = \frac{15}{55}$$

$$c_1 = 1.018182 \qquad c_2 = -0.109091 \qquad c_3 = 0.272727$$

$$\boxed{y = 1.018182 - 0.109091 \cdot x + 0.272727 \cdot x^2}$$

parabol

C Kodu:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <locale.h>
float determinant(float M[3][3]);
void matrisyaz(float N[3][3]);

int main()
{   setlocale(LC_ALL,"Turkish");
    int m,n,i,j,t,s,topx=0,topy=0,topx2=0,topx3=0,topx4=0,topyx=0,topyx2=0,detA;
    printf("Kaç adet nokta olduğunu giriniz--->");
      scanf("%d",&n);
    float X[n],Y[n],A[3][3],B[3],tempA[3][3],c[3];

    for(i=0;i<n;i++)
    { printf("%d. noktadaki x değerini giriniz: ",i+1);
        scanf("%f",&X[i]);
      printf("%d. noktadaki y değerini giriniz: ",i+1);
        scanf("%f",&Y[i]);
    }
```

```
for(i=0;i<n;i++)
    {topx=topx+X[i];
    topy=topy+Y[i];
    topx2=topx2+(X[i]*X[i]);
    topx3=topx3+(X[i]*X[i]*X[i]);
    topx4=topx4+(X[i]*X[i]*X[i]*X[i]);
    topyx=topyx+(X[i]*Y[i]);
    topyx2=topyx2+(X[i]*X[i]*Y[i]);
    }
    A[0][0]=n; A[0][1]=topx;  A[0][2]=topx2;
   A[1][0]=topx; A[1][1]=topx2;  A[1][2]=topx3;
   A[2][0]=topx2; A[2][1]=topx3;  A[2][2]=topx4;
   B[0]=topy; B[1]=topyx;   B[2]=topyx2;
   printf("Lineer Denklem Sistemi Cramer Yöntemiyle Çözülmüştür...\n");
   printf("Katsayılar matrisi:");
   matrisyaz(A);
   printf("B sonuç matrisi:\n");
   for(i=0;i<3;i++)
   printf("%f\n",B[i]);
   printf("\n");
```

```
detA=determinant(A);
   for(i=0;i<3;i++)
   {
       for(t=0;t<3;t++)
        {
          for(s=0;s<3;s++)
           {  tempA[t][s]=A[t][s];}
         }

        for(j=0;j<3;j++)
           {tempA[j][i]= B[j];}

         c[i]=determinant(tempA)/detA;
         printf("%d. sütun elemanları ile B matrisinin elemanları yer değiştirdi:",i+1);
        matrisyaz(tempA);
   }
```

```
printf("Sonuclar\n");
   for(i=0;i<3;i++)
       printf("c[%d]=%f\n",i+1,c[i]);

  printf("\nElde Edilen İkinci Derecen Lineer Regresyon Denklemi (Parabol Denklemi):\n");
  printf("y=%f + (%f).x + (%f).x^2\n",c[0],c[1],c[2]);

   getch();
   return 0;
}
```

Ekran Çıktısı:

```
float determinant(float M[3][3])
{ float det,m1,m2,m3;
   m1=M[0][0]*(M[1][1]*M[2][2]-M[1][2]*M[2][1]);
   m2=M[0][1]*(M[1][0]*M[2][2]-M[1][2]*M[2][0]);
   m3=M[0][2]*(M[1][0]*M[2][1]-M[1][1]*M[2][0]);
   det=m1+(-1)*m2+m3;
   return det;
}

void matrisyaz(float N[3][3])
 {
      int i,j;
       for(i=0;i<3;i++)
       {
           printf("\n");
           for(j=0;j<3;j++)
             printf("%f\t",N[i][j]);
       }
       printf("\n\n");
  }
```

```
Kaç adet nokta olduğunu giriniz--->4
1. noktadaki x değerini giriniz: 0
1. noktadaki y değerini giriniz: 1
2. noktadaki x değerini giriniz: 2
2. noktadaki y değerini giriniz: 2
3. noktadaki x değerini giriniz: 3
3. noktadaki y değerini giriniz: 3
4. noktadaki x değerini giriniz: 4
4. noktadaki y değerini giriniz: 5
Lineer Denklem Sistemi Cramer Yöntemiyle Çözülmüştür...
Katsayılar matrisi:
4,000000        9,000000        29,000000
9,000000        29,000000       99,000000
29,000000       99,000000       353,000000

B sonuç matrisi:
11,000000
33,000000
115,000000
```

```
1. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
11,000000       9,000000        29,000000
33,000000       29,000000       99,000000
115,000000      99,000000       353,000000

2. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
4,000000        11,000000       29,000000
9,000000        33,000000       99,000000
29,000000       115,000000      353,000000

3. sütun elemanları ile B matrisinin elemanları yer değiştirdi:
4,000000        9,000000        11,000000
9,000000        29,000000       33,000000
29,000000       99,000000       115,000000

Sonuclar
c[1]=1,018182
c[2]=-0,109091
c[3]=0,272727

Elde Edilen İkinci Derecen Lineer Regresyon Denklemi (Parabol Denklemi):
y=1,018182 + (-0,109091).x + (0,272727).x^2
```

Yöntemin Genel Halinin C kodu Çalışma Sorusu Olarak Bırakılmıştır…

Çalışma Sorusu:

$(-3,3)$, $(0,1)$, $(2,1)$, $(4,3)$ noktalarına

$y = c_1 + c_2 . x + c_3 . x^2$ tipinde öyle bir

eğri tipi bulunuz ki hataların kareleri

toplamı minimum olsun.

$c_1 = 0.850519$

$c_2 = -0.192495$

$c_3 = 0.178462$

Lineer Hale Dönüştürülebilen Modeller

$y = a.e^{bx}$, a ve b parametre olmak üzere lineer olmayan bir denklemdir.

(x_i, y_i) , $i = 1, ---, N$ (Veri Seti)

$y_i = a.e^{bx_i} + \varepsilon_i$, $\varepsilon_i \rightarrow$ hatadır

$\sum(\varepsilon_i)^2 = \sum(a.e^{bx_i} - y_i)^2 = E(a,b)$

$$\frac{\partial E}{\partial a} = 0 \quad , \quad \frac{\partial E}{\partial b} = 0$$

$$a \cdot \sum x_i \, e^{2b x_i} - \sum x_i y_i \cdot e^{b x_i} = 0$$

$$a \cdot \sum e^{2b \cdot x_i} - \sum y_i \cdot e^{b x_i} = 0$$

(lineer olmayan denklem sistemi)

Basit iterasyon veya Newton-Raphson yöntemleri ile bu denklem sistemi çözülebilir.

- Fakat, şimdi bu sistemi Lineer hale dönüştüreceğiz.

$$y = a.e^{bx} \Rightarrow \ln y = \ln a + bx$$

$\ln y = Y$

$\ln a = A$

$$\boxed{Y = A + bx}$$

→ Lineer olmayan modelin Lineer hali.

$$y = a \cdot x^b \Rightarrow \log y = \log a + b \cdot \log x$$

Lineer değil

$\log y = Y$

$\log a = A$

$\log x = X$

$$\boxed{Y = A + b \cdot X}$$

Lineer hali

$$y = a \cdot \frac{x}{b+x} \Rightarrow \frac{1}{y} = \frac{b+x}{ax} = \frac{b}{a} \cdot \frac{1}{x} + \frac{1}{a}$$

Lineer değil

(b paydada olduğundan)

$\frac{1}{y} = Y$

$\frac{1}{x} = X$

$\frac{b}{a} = A$

$\frac{1}{a} = B$

$$\boxed{Y = A \cdot X + B}$$

Lineer hali

Örnek!

x_i	0	1	2	3	4
y_i	1.5	2.5	3.5	5	7.5

verilerini kullanarak $y = c \cdot e^{ax}$ olacak şekilde en uygun eğriyi bulunuz.

$$y = c \cdot e^{ax}$$

$$\underbrace{\ln y}_{} = \underbrace{\ln c}_{} + a.x$$

$$Y = B + a.x \quad \text{(Lineer hali)}$$

$$\begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} B \\ a \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum Y_i \cdot x_i \end{bmatrix}$$

x_i	y_i	$Y_i = \ln(y_i)$	x_i^2	$x_i \cdot Y_i$
0	1.5	0.40547	0	0
1	2.5	0.91629	1	0.91629
2	3.5	1.25276	4	2.50553
3	5	1.60944	9	4.82831
4	7.5	2.01490	16	8.05961
+ 10	+ 20	+ 6.19886	+ 30	+ 16.30974

$$\begin{bmatrix} N & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} B \\ a \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum Y_i \cdot x_i \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 \\ 10 & 30 \end{bmatrix} \begin{bmatrix} B \\ a \end{bmatrix} = \begin{bmatrix} 6.19886 \\ 16.30974 \end{bmatrix}$$

$a = ?$ $\quad B = ?$

$$5\beta + 10a = 6.19886$$

$$10\beta + 30a = 16.30974$$

$$a = 0.391202$$

$$\beta = 0.457367$$

$$\ln c = \beta \Rightarrow c = e^{\beta} = 1.579908$$

$$y = c \cdot e^{ax} \Rightarrow \boxed{y = 1.579908 \cdot e^{(0.391202) \cdot x}}$$

Kaynaklar

- Numerical Analysis, Richard L. Burden, Brooks/Cole Cengage Learning, Boston., 2009.
- Numerical Methods for Mathematics, Science, and Engineering, 2nd Edition, John H. Mathews, Prentice Hall International Edition, 1992.
- Nümerik Analiz, (Numerical Analysis, D. Kincaid, W. Cheney, 3rd ed.(2002)), Nuri Özalp, Elif Demirci, Gazi Kitabevi Yayınları, 2012.
- Sayısal Analiz ve Mühendislik Uygulamaları, İrfan Karagöz, Nobel Yayıncılık, 2011.
- Sayısal Çözümleme, Recep Tapramaz, Literatür yayıncılık, 2002.
- Bilgisayar Uygulamalı Sayısal Analiz Yöntemleri, Eyüp Sabri Türker, Engin Can, II. Baskı, Değişim Yayınları.