プログラミング演習 第3課題1

滝本亘（学籍番号 1029-33-1175）

2022年6月13日

問題1(転置行列)

(1)

作成したプログラムをコード1に示す。

コード 1: 転置行列

```c
#include <stdio.h>
#define N 3

int main(void)
{
    //行列の定義
    double a[N][N] = {{1,2,3},{4,5,6},{7,8,9}};

    //転置行列
    double a_t[N][N];
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<N; j++)
        {
            a_t[i][j] = a[j][i];
        }
    }

    //結果をターミナルに出力
    for(int i=0; i<N; i++)
    {
        for(int j=0; j<N; j++)
        {
            printf("%.1f␣", a_t[i][j]);
        }
        printf("\n");
    }
}
```

問題2(回帰問題)

(1)

$$\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2 = (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) \quad (1)$$

$$= \boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y}-\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \quad (2)$$

ここで、$\boldsymbol{X}^T\boldsymbol{y} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$、$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}$ と置く。

$\boldsymbol{\beta} = (\beta_0, \beta_1)^T$ の成分ごとに微分することで、次の式が得られる。

$$a_1 = \beta_0 x_1 + \beta_1 x_2 \tag{3}$$
$$a_2 = \beta_0 x_3 + \beta_1 x_4 \tag{4}$$

以上より、

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{5}$$
$$\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \tag{6}$$
$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{7}$$

(2)

作成したプログラムをコード 2 に示す。

コード 2: パラメータ β の決定

```
#include <stdio.h>
#define N 4
#define M 2

int main(void)
{
    //行列の定義
    double x[N][M] = {{1,1},{1,2},{1,3},{1,4}};
    double y[4] = {1.5, 3.2, 4.1, 5.2};

    //転置行列
    double tx[N][N];
    for(int i=0; i<M; i++)
    {
        for(int j=0; j<N; j++)
        {
            tx[i][j] = x[j][i];
        }
    }

    //転置行列と元の行列の積
    double txx[M][M];
    for(int i=0; i<M; i++)
    {
        for(int j=0; j<M; j++)
        {
            txx[i][j] = 0;

            for(int k=0; k<N; k++)
            {
                txx[i][j] = txx[i][j] + tx[i][k]*x[k][j];
            }
        }
    }

```

```
36      //掃出し法に用いる行列
37      double sweep[M][M*2];
38      for(int i=0; i<M; i++)
39      {
40          for(int j=0; j<M; j++)
41          {
42              sweep[i][j] = txx[i][j];
43              sweep[i][M+j] = (i==j) ? 1 :0;
44          }
45      }
46
47      //掃出し法
48      for(int k=0; k<M; k++)
49      {
50          double a = 1/sweep[k][k];
51
52          for(int j=0; j<M*2; j++)
53          {
54              sweep[k][j] *= a;
55          }
56
57          for(int i=0; i<M; i++)
58          {
59              if(i == k)
60              {
61                  continue;
62              }
63
64              a = -sweep[i][k];
65
66              for(int j=0; j<M*2; j++)
67              {
68                  sweep[i][j] += sweep[k][j]*a;
69              }
70          }
71      }
72
73      //逆行列
74      double inv[M][M];
75      for(int i=0; i<M; i++)
76      {
77          for(int j=0; j<M; j++)
78          {
79              inv[i][j] = sweep[i][M+j];
80          }
81      }
82
83      //転置と元の行列の積の逆行列と転置行列の積
84      double xxx[M][N];
85      for(int i=0; i<M; i++)
86      {
87          for(int j=0; j<N; j++)
88          {
89              xxx[i][j] = 0;
90
91              for(int k=0; k<M; k++)
92              {
93                  xxx[i][j] = xxx[i][j] + inv[i][k]*tx[k][j];
94              }
95          }
96      }
97
98      //ベータ
99      double b[N];
100     for(int i=0; i<M; i++)
101     {
102         b[i] = 0;
103
```

```
104             for(int k=0; k<N; k++)
105             {
106                 b[i] = b[i] + xxx[i][k]*y[k];
107             }
108         }
109
110         //結果を出力
111         for(int i=0; i<M; i++)
112         {
113             printf("%.1f␣\n", b[i]);
114         }
115     }
```

結果は、$\beta_0 = 0.5$, $\beta_1 = 1.2$ となる。

(3)

元のデータとフィッティングしたモデルの比較を図に示す。

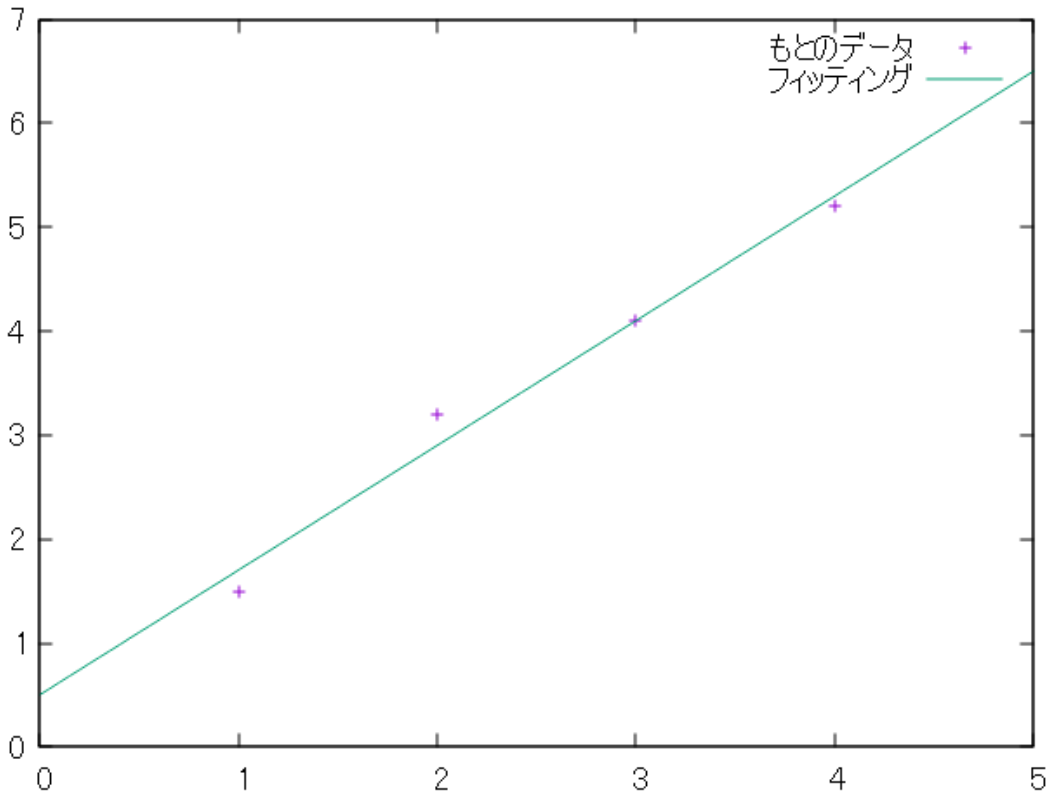

図 1: 元のデータとフィッティングしたモデルの比較