数値計算 Class-8 演習

21T2166D 渡辺大樹

2023 年 7 月 27 日

1　演習内容

Class-8 でも引き続き、いくつかの xy 平面上のデータからそのデータをすべて通るなめらかな曲線を描くための手法、最小二乗法についてコードを実装し、実際に動かしていく。

最小二乗法はよく用いられる、データから関数を見つけ出す手法で、大雑把に言えばデータ点と予測される関数の y 座標の値の誤差を二乗し、それが最小になる関数を探す手法である。

まず、データ点から予測される関数が単一の一次方程式 $y = ax + b$ であるときを考える。データ点はそれぞれ $(x_i, y_i) \quad (i = 0, 1, \cdots, n)$ である。このときのデータと予測される関数との誤差 E_i は

$$E_i = ax_i + b - y_i \tag{1}$$

で表される。

これを二乗してすべてのデータ点で足し合わせることで二乗誤差 E を得る。

$$\begin{aligned} E &= \sum_{i=0}^{n} (E_i)^2 \\ &= \sum_{i=0}^{n} (ax_i + b - y_i)^2 \end{aligned} \tag{2}$$

これを最小にする a, b を決めたいので a, b それぞれで (2) の式を偏微分することで

$$\begin{aligned} \frac{\partial E}{\partial a} &= \frac{\partial \sum_{i=0}^{n} (ax_i + b - y_i)^2}{\partial a} \\ &= \sum_{i=0}^{n} 2x_i (ax_i + b - y_i) \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{\partial E}{\partial b} &= \frac{\partial \sum_{i=0}^{n} (ax_i + b - y_i)^2}{\partial b} \\ &= \sum_{i=0}^{n} 2(ax_i + b - y_i) \end{aligned} \tag{4}$$

の (3),(4) の二式が得られる。

この二式イコール 0 を連立方程式として解けば求めたい a, b の係数を得られる。

これを拡張し、一般化すれば任意の関数の一次結合をした関数で誤差の二乗が最小の関数を得ることができる。

詳細は省略するが、任意の適当な n 個の関数を一次の係数 $a_i \quad (i = 1, 2, \cdots, n)$ で一次結合し、与えられた n 個のデータ点を代入した連立方程式に関して、二乗誤差の足し合わせ E を $a_i \quad (i = 1, 2, \cdots, n)$ で偏微分することによって得られる連立方程式を掃き出し法などを用いて解くことで一次の係数 $a_i \quad (i = 1, 2, \cdots, n)$ を求めることができる。

以上の最小二乗法によるなめらかな関数を求めるアルゴリズムは以下のソースコード 1 で実装される。

ソースコード 1　minjijo.c

```
1  #include <stdio.h>
2  #include <math.h>
3  #define N 10
4      /*基本関数の関数値を求める*/
5      void
6      ffv(int p, double a, double *b)
7  {
8      switch (p)
9      {
10     case 1:
11         *b = a;
12         break;
13     case 2:
14         *b = 1.0 / a;
15         break;
16     case 3:
17         *b = exp(a);
18         break;
19     case 4:
20         *b = 1.0;
21         break;
22     default:
23         *b = a;
24         break;
25     }
26 }
27 int main(void)
28 {
29     int f, g, n, i, j, l;
30     double xx, yy, p, q, h, s, fx, gx, c[4][4], d[4];
```

```
31      double x[N], y[N], x2[N], x3[N], a[N][4], b[4][N];
32      double xi;
33      char z, zz;
34      printf("このプログラムは最小 2乗法によって \n");
35      printf("y = a*f(x) + b*g(x) \n");
36      printf("の形の曲線をあてはめるものです．\n\n");
37      printf("基本関数f(x), g(x)を１～４の番号で選択してください\n");
38      while (1)
39      {
40          printf("f(x)=[1:(x), 2:(1/x), 3:(e^x)]--> ");
41          scanf("%d%c", &f, &zz);
42          if ((1 <= f) && (f <= 3))
43              break;
44      }
45      while (1)
46      {
47          printf("g(x)=[1:(x), 2:(1/x), 3:(e^x), 4(定数)]--> ");
48          scanf("%d%c", &g, &zz);
49          if ((1 <= g) && (g <= 4))
50              break;
51      }
52      /*** データの入力 ***/
53      while (1)
54      {
55          printf("データの個数は何個ですか？(1<n<10)n = ");
56          scanf("%d%c", &n, &zz);
57          if ((n <= 1) || (10 <= n))
58              continue;
59          printf("\n データ x の値は小から大の順に入力する．\n");
60          for (i = 1; i <= n; i++)
61          {
62              printf("X = ");
63              scanf("%lf%c", &x[i], &zz);
64              printf("Y = ");
65              scanf("%lf%c", &y[i], &zz);
66              /*** 関数を呼び出す ***/
67              xi = x[i];
68              ffv(f, xi, &fx);
69              ffv(g, xi, &gx);
70              x2[i] = fx;
71              x3[i] = gx;
72              a[i][1] = x2[i];
```

```
73              a[i][2] = x3[i];
74              a[i][3] = y[i];
75              b[1][i] = a[i][1];
76              b[2][i] = a[i][2];
77          }
78          printf("\n 正しく入力しましたか？ (y/n) ");
79          scanf("%c%c", &z, &zz);
80          if (z == 'y')
81              break;
82      }
83      /*** tA・A を計算して配列 c[2][3]に入れる ***/
84      for (i = 1; i <= 2; i++)
85      {
86          for (j = 1; j <= 3; j++)
87          {
88              s = 0.0;
89              for (l = 1; l <= n; l++)
90              {
91                  s = s + b[i][l] * a[l][j];
92              }
93              c[i][j] = s;
94          }
95      }
96      /*** 正規方程式をガウス・ジョルダン法で解く ***/
97      for (i = 1; i <= 2; i++)
98      {
99          p = c[i][i];
100         for (j = 1; j <= 3; j++)
101         {
102             c[i][j] = c[i][j] / p;
103         }
104         for (l = 1; l <= 2; l++)
105         {
106             if (l != i)
107             {
108                 q = c[l][i];
109                 for (j = i; j <= 3; j++)
110                 {
111                     c[l][j] = c[l][j] - q * c[i][j];
112                 } // j
113             } // if
114         } // l
```

```
115     } // i
116     /*** 答を配列 d[1], d[2]に入れる ***/
117     for (i = 1; i <= 2; i++)
118     {
119         d[i] = c[i][3];
120     }
121     printf("\n 求めた基本関数の係数の出力\n");
122     printf("a = d[1] = %lf\n", d[1]);
123     printf("b = d[2] = %lf\n", d[2]);
124     printf("\n エンターキーを押せば数表を出力します. \n");
125     scanf("%c", &zz);
126     /** グラフを描くための準備（数表を出力）**/
127     h = (x[n] - x[1]) / 50.0;
128     xx = x[1];
129     for (i = 0; i <= 50; i++)
130     {
131         ffv(f, xx, &fx);
132         ffv(g, xx, &gx);
133         yy = d[1] * fx + d[2] * gx;
134         printf("%lf\t%lf\n", xx, yy);
135         xx = xx + h;
136     }
137 }
```

コード上では 77 行目まででデータの関係関数の入力と、実際のデータの入力を行っている。

83 行目からは二乗誤差を偏微分し、足し合わせることで出てくる $\mathbf{a}$ の係数行列 A^TA を計算し、それをもとに 97 行目からガウスジョルダン法を用いて連立方程式を解いている。

解けた解は d[1],d[2] として出力され、グラフを書くためにデータの x の最大値最小値を 50 分割した関数の値を出力できるようにしている。

2 演習結果

演習として以下のデータ表から最小二乗法でなめらかな関数を出力する。

x	0.01	0.1	0.5	1.0	2.0	4.0	8.0	10.0
y	218	26	8	4	1	8	31	230

Excel などでプロットするとわかるがこれは関係関数として $f(x) = \frac{1}{x}$, $g(x) = e^x$ と定めることができる。

実際にこの条件で入力すると

$$\mathrm{a} = \mathrm{d}[1] = 2.185093$$
$$\mathrm{b} = \mathrm{d}[2] = 0.010431$$

のような出力を得ることができる。

Python を用いてデータ点とこの係数で関係関数を結合した関数 $\frac{2.185093}{x} + 0.010431e^x$ をプロットすると以下図 1 のようなグラフを得られる。

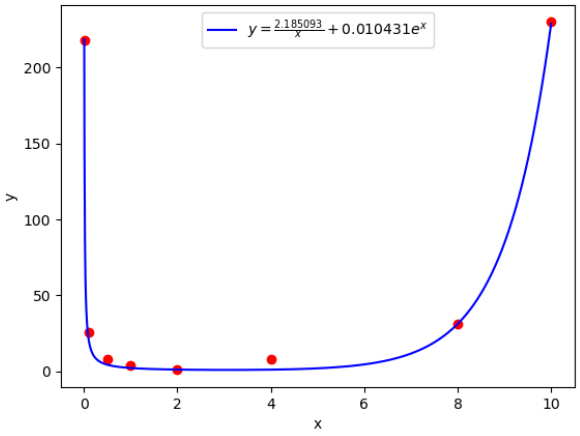

図 1　ソースコード 1 で得られた関数とデータ点のプロット

この図 1 より誤差の少ない関数を取得できていることが確認できる。

教科書の演習問題も解いていく。

(a)

以下の表を $y = ax + b$ で解く。

x	1.0	2.0	3.0	4.0	5.0
y	2.0	2.5	2.9	3.5	4.4

結果はこのように得られた。

$$\mathrm{a} = \mathrm{d}[1] = 0.580000$$
$$\mathrm{b} = \mathrm{d}[2] = 1.320000$$

答えと一致する値となった。

(b)

以下の表を $y = ax + \frac{b}{x}$ で解く。

x	0.2	0.5	1.0	2.0	4.0	8.0	10.0
y	12.1	4.9	2.9	2.1	2.1	3.4	4.3

結果はこのように得られた。

a = d[1] = 0.398093
b = d[2] = 2.401050

答えと一致する値となった。

(c)

以下の表を $y = ax + b$ で解く。

x	1.0	1.5	2.0	3.0	5.0
y	2.4	4.0	6.2	9.5	17.1

結果はこのように得られた。

a = d[1] = 3.680000
b = d[2] = -1.360000

答えと一致する値となった。

(d)

以下の表を $y = ax + b$ で解く。

x	18	28	38	48	58	68	78	88	98
y	15.2	15.9	16.7	16.9	17.4	18.0	18.7	19.3	19.8

結果はこのように得られた。

a = d[1] = 0.056167
b = d[2] = 14.286778

答えと一致する値となった