

# 数学模型与数学软件

## 第 8 次作业

1907402030

熊 雄<sup>\*</sup>



2022 年 5 月 10 日

---

<sup>\*</sup>mrxiong@foxmail.com 苏州大学数学科学学院本科生

### Problem 1

(Page 169 Ex.4.)

某海岛上有 12 个主要的居民点, 每个居民点的位置 (用平面坐标  $x, y$  表示, 距离单位: km) 和居住的人数  $R$  如表 7.7 所示. 现在准备在岛上建一个服务中心为居民提供各种服务, 那么服务中心应该建在何处?

表7.7

居民点	$x$	$y$	$R$	居民点	$x$	$y$	$R$
1	0	0	600	7	4.43	3.26	600
2	8.20	0.50	1000	8	2.58	9.32	800
3	0.50	4.90	800	9	0.72	9.96	1000
4	5.70	5.00	1400	10	9.76	3.16	1200
5	0.77	6.49	1200	11	3.19	7.20	1000
6	2.87	8.76	700	12	5.55	7.88	1100

**Solution.**

#### • 建立模型

设第  $i$  个居民点的位置为  $(x_i, y_i)$ , 第  $i$  个居民点的人数为  $R_i$ , 则建一个服务中心  $\text{pos}(x, y)$  的最优方案是每个人到服务中心的距离之和最短, 即求解如下的无约束优化问题:

$$\min \text{SUM} = \sum_{i=1}^{12} \sqrt{(x_i - x)^2 + (y_i - y)^2}. \quad (1)$$

显然这是一个非线性最小二乘拟合问题.

#### • 代码实现

##### a) 函数 M 文件

```

1 function y = Func(pos, X, Y, R)
2 y = 0;
3 for i = 1 : length(X)
4     y = y + sqrt((pos(1) - X(i))^2 + (pos(2) - Y(i))^2) * R(i);
5 end

```

##### b) 寻找最小值点

```

1 %初始数据
2 X = [0, 8.20, 0.50, 5.70, 0.77, 2.87, 4.43, 2.58, 0.72, 9.76, 3.19, 5.55];
3 Y = [0, 0.50, 4.90, 5.00, 6.49, 8.76, 3.26, 9.32, 9.96, 3.16, 7.20, 7.88];
4 R = [600, 1000, 800, 1400, 1200, 700, 600, 800, 1000, 1200, 1000, 1100];
5 %调用 fminunc 确定最优位置
6 pos = fminunc(@Func, [0, 0], [], X, Y, R);
7 %作图

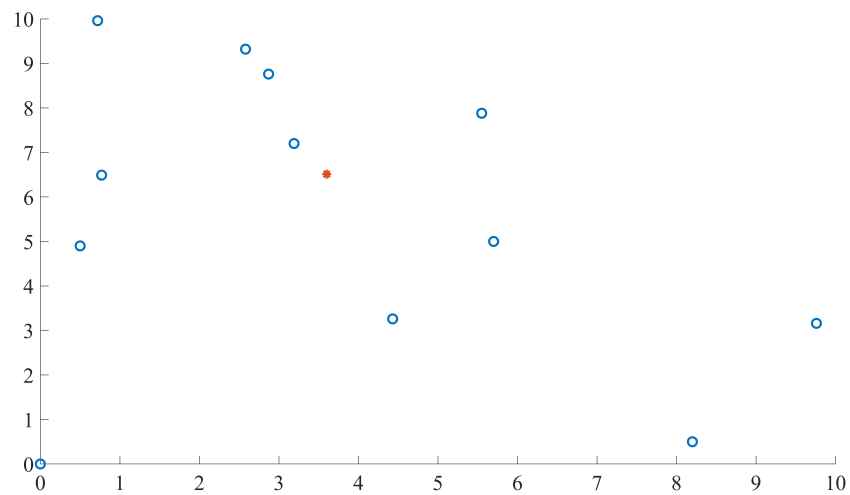
```

```
8 scatter(X,Y,70)
9 hold on;
10 scatter (pos(1) , pos(2) ,70)
```

### • 结果分析

运行上述代码可以得到

pos = 3.6010 6.5142.



即故服务中心应设立在 (3.6010,6.5142) 坐标处, 如上图红色标记点. ■

## Problem 2

(Page 170 Ex.6.)

有一组数据  $(t_i, y_i)$  ( $i = 1, 2, \dots, 33$ ), 其中  $t_i = 10 * (i - 1)$ ,  $y_i$  由表 7.9 给出, 现要用这组数据拟合函数

$$f(x, t) = x_1 + x_2 e^{-x_4 t} + x_3 e^{-x_5 t} \quad (2)$$

中的参数  $x$ , 初始值取  $(0.5, 1.5, -1, 0.01, 0.02)$ , 试用两种方法求解 (两种 Matlab 命令).

表7.9

$i$	$y_i$	$i$	$y_i$	$i$	$y_i$
1	0.844	12	0.718	23	0.478
2	0.908	13	0.685	24	0.467
3	0.932	14	0.658	25	0.457
4	0.936	15	0.628	26	0.448
5	0.925	16	0.603	27	0.438
6	0.908	17	0.580	28	0.431
7	0.881	18	0.558	29	0.424
8	0.850	19	0.538	30	0.420
9	0.818	20	0.522	31	0.414
10	0.784	21	0.506	32	0.411
11	0.751	22	0.490	33	0.406

**Solution.**

- 利用 lsqcurvefit 函数

- a) 函数 M 文件

```

1 %编写 f 函数
2 function f = fl(x, t)
3     f = x(1) + x(2) * exp(- x(4) * t) + x(3) * exp(- x(5) * t);
4 end

```

- b) 利用 lsqcurvefit 函数求解的代码

```

1 %lsqcurvefit
2 x0 = [0.5 1.5 -1 0.01 0.02] ;
3 i = 1: 33;
4 t = 10 * (i - 1) ;
5 y = [0.844 0.908 0.932 0.936 0.925 0.908 0.881 0.850 0.818 0.784 0.751
        0.718 0.685 0.658 0.628 0.603 0.580 0.558 0.538 0.522 0.506 0.49
        0.478 0.467 0.457 0.448 0.438 0.431 0.424 0.42 0.414 0.411 0.406]
        ;
6 x = lsqcurvefit(@fl, x0, t , y )
7 yy = fl(x, t);

```

```
8 plot(t, y, 'k+', t, yy, 'r')
```

运行后可以得到输出为

$$x_1 = 0.375410049840934,$$

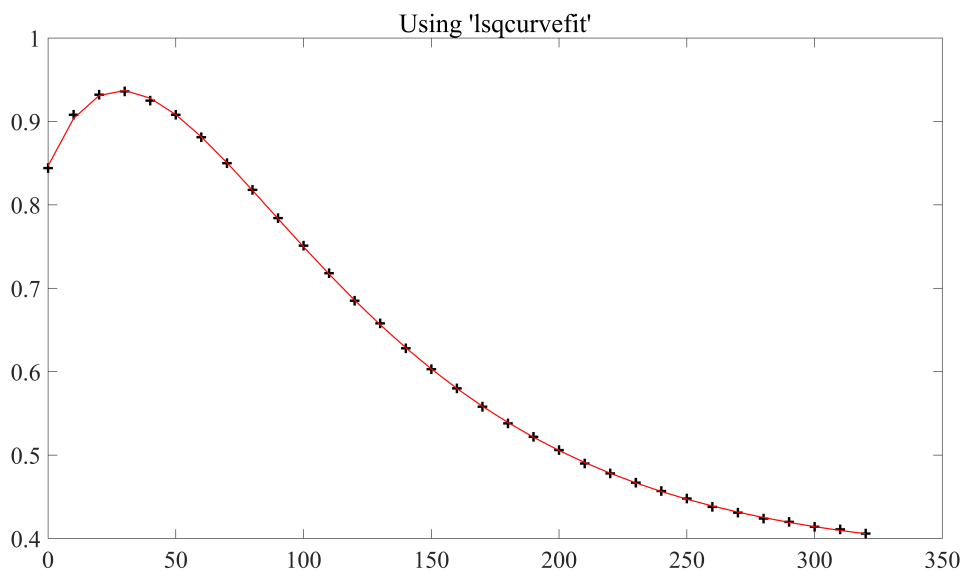
$$x_2 = 1.93584671104716,$$

$$x_3 = -1.46468693267017,$$

$$x_4 = 0.0128675342017908,$$

$$x_5 = 0.0221227004151004.$$

拟合的图像如下



## • 利用 lsqnonlin 函数

### a) 函数 M 文件

```
1 %编写 f 函数
2 function f = f2(x, t, y)
3     f = x(1) + x(2) * exp(-x(4) * t) + x(3) * exp(-x(5) * t) - y;
4 end
```

### b) 利用 lsqnonlin 函数求解的代码

```
1 %lsqnonlin
2 x0 = [0.5 1.5 -1 0.01 0.02] ;
3 i = 1: 33;
4 t = 10 * (i - 1) ;
```

```

5 y = [0.844 0.908 0.932 0.936 0.925 0.908 0.881 0.850 0.818 0.784 0.751
      0.718 0.685 0.658 0.628 0.603 0.580 0.558 0.538 0.522 0.506 0.49
      0.478 0.467 0.457 0.448 0.438 0.431 0.424 0.42 0.414 0.411 0.406 ]
      ;
6 x = lsqnonlin(@f2, x0, [], [], [], t, y)
7 yy = f1(x, t);
8 plot(t, y, 'k+', t, yy, 'b')

```

运行后可以得到输出为

$$x_1 = 0.375410049840934,$$

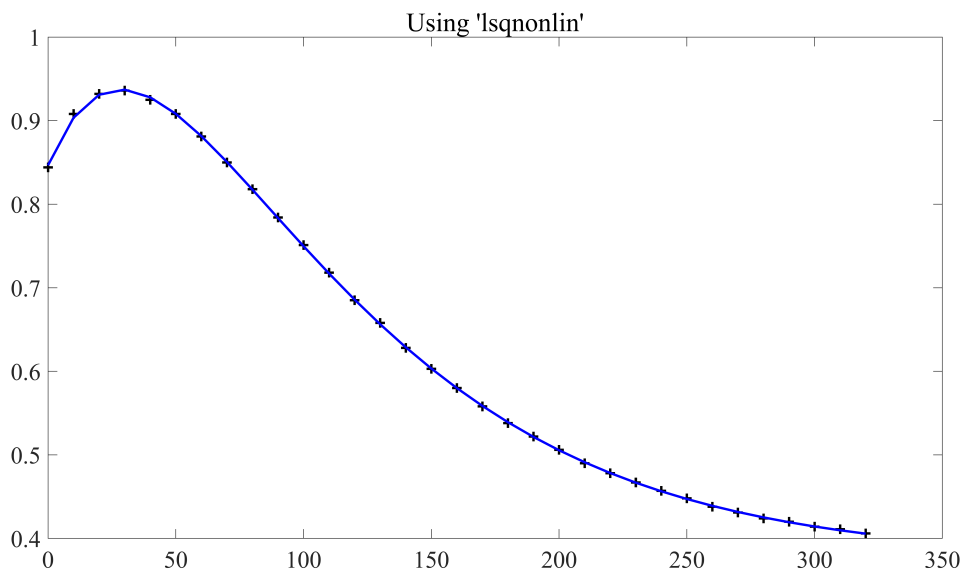
$$x_2 = 1.93584671104716,$$

$$x_3 = -1.46468693267017,$$

$$x_4 = 0.0128675342017908,$$

$$x_5 = 0.0221227004151004.$$

拟合的图像如下



### • 两种方法运行结果的比较

利用 `lsqcurvefit` 函数与利用 `lsqnonlin` 函数方法得到的**系数是相同的** (实际上, 两者使用的算法是一样的), 但是 `lsqcurvefit` 函数的结果更为精确.

### • 对 $y_i$ 作扰动的结果

我们选择利用 `unifrnd` 函数在  $(-0.05, 0.05)$  中产生随机数, 给题目中  $y_i$  增加扰动.

```

1 x0=[0.5,1.5,-1,0.01,0.02];
2 i = 1: 33;
3 t = 10 * (i - 1) ;

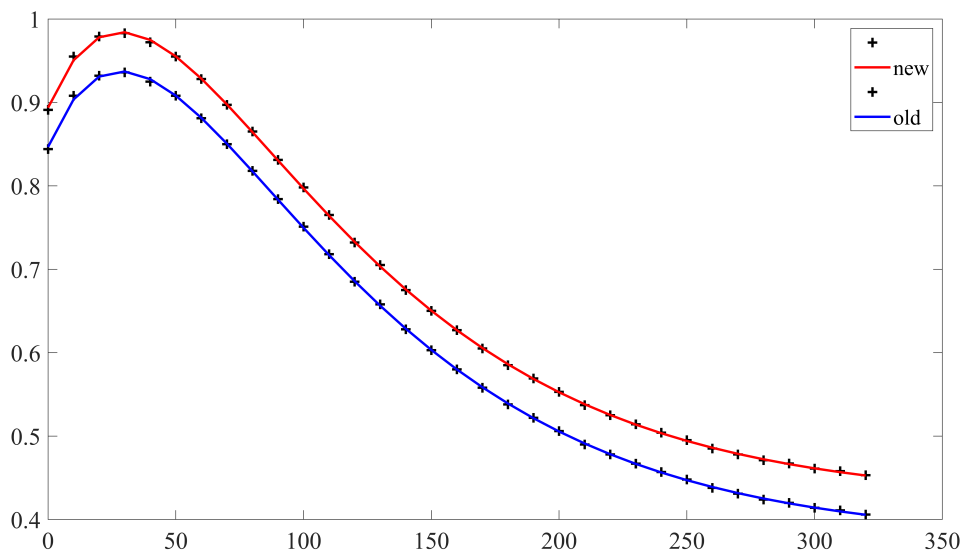
```

```

4 y = [0.844 0.908 0.932 0.936 0.925 0.908 0.881 0.850 0.818 0.784 0.751 0.718
      0.685 0.658 0.628 0.603 0.580 0.558 0.538 0.522 0.506 0.49 0.478 0.467
      0.457 0.448 0.438 0.431 0.424 0.42 0.414 0.411 0.406 ] ;
5 %加上扰动
6 y1 = y + unifrnd(-0.05,0.05);
7 x1 = lsqcurvefit(@f1, x0, t, y1);
8 yy1 = f1(x1, t);
9 %未加扰动
10 x = lsqcurvefit(@f1, x0, t, y);
11 yy = f1(x, t);
12 %作图, 蓝色为原曲线, 红色为加上扰动后的曲线
13 plot(t, y1, 'k+', t, yy1, 'r', t, y, 'k+', t, yy, 'b')

```

拟合的图像如下:



由于增加的是随机扰动, 因此每次函数图像都不同. ■