基于python的最小二乘曲线拟合报告

陶睿

（中国海洋大学 信息科学与工程学院 山东省 青岛市 266100）

摘要： 本文利用最小二乘原理，使用python语言实现了对离散数据进行了线性最小二乘曲线拟合。此外，对于一些非线性问题，如经验公式形如 $y = ae^{bx}$ 的数据，实现了数据线性化技术，将其转化为线性问题进行拟合。拟合的程序均为自行实现。

关键词： 最小二乘法；线性拟合；数据线性化

0 引言

在科学实验中,经常可以得到一组实验数据($(x_i, y_i)(i = 1, 2, 3, \ldots\ldots, n)$),如何根据所给的这些数据点,找出大致描述这些变量间的函数关系,从而进行预测,这就是曲线拟合所要解决的问题。由于在实际生活中,变量与变量之间的关系往往是非线性的,这时,通常是选配一条比较接近的曲线,通过变量变换把非线性方程线性化,然后对线性化的函数应用最小二乘法求解拟合函数。

应用最小二乘法的一个前提条件是拟合函数y=(x)的具体形式为已知,即要求首先确定x与y之间的内在关系,函数的形式是成千上万的,具体形式的确定或假设,一般有两种途径。当人们对研究对象的内在特性和各因素间的关系有比较充分的认识时,一般用机理分析的方法建立描述y=(x)的数学模型,再用曲线拟合的方法确定模型中的参数。但如果由于客观事物内部规律的复杂性及人们认识程度的限制,无法建立合乎机理规律的数学模型,这时,只有对观测数据(x_i,y_i)进行分析,绘制散点图,先猜测y=(x)的类型,通过上机实验、误差计算和统计量分析,不断对比改进,才能选出拟合数据较好的函数类型。

1 最小二乘法拟合

线性最小二乘原理是解决曲线拟合最常用的方法，基本思路是，令

$$f(x) = a_1 r_1(x) + a_2 r_2(x) + \ldots + a_m r_m(x)$$

其中 $r_k(x)$ 是事先选定的一组线性无关的函数， a_k 是待定系数，拟合准则是 y_i ，$i = 1, 2, \ldots, n$ 与 $f_i(x)$ 的距离 δ_i 的平方和最小，称为最小二乘准则。

针对离散数据做线性拟合的时候，首要任务是确定 $r(x)$ 的选取，可以通过机理分析和散点图的形式来确定。常用的函数曲线有直线、多项式、双曲线、指数函数。以下以多项式函数为例，确定线性最小二乘曲线拟合。

若选取 $y^* = a + bx$ 为经验函数，误差平方和为 $Q(a, b) = \sum_{i=1}^{n} (y_i - a - bx_i)^2$

取误差平方和 $Q(a, b)$ 分别关于 a , b 的偏导数，并令它们等于0，得到 a , b 应满足方程

$$\begin{cases}\dfrac{\partial Q}{\partial a}=-2\sum_{i=1}^{n}\left(y_i-a-b\,x_i\right)=0\\ \dfrac{\partial Q}{\partial b}=-2\sum_{i=1}^{n}\left(y_i-a-b\,x_i\right)x_i=0\end{cases}\tag{2}$$

（2）式称为正规方程组。解此方程组即可确定 a , b ，从而得到直线方程

$$y^*=a+bx$$

。

对一组测定数据用最小二乘原理找出其合适的数学公式，可以分以下几步：

1．由观测数据作出散点图

2．根据散点图确定近似公式的函数类

3．用最小二乘原理确定函数中的未知参数

这一方法称为数据拟合法。

常用的曲线（函数类）有直线、多项式、双曲线、指数曲线等，实际操作中可以在直观判断的基础上，选几种曲线分别做拟合，然后比较看哪条曲线的最小二乘指标最小。

2 数据线性化

在许多实际问题中，变量之间内在的关系并不简单地呈线性关系，但是通常可以使用数据线性化技术来拟合各种曲线,如$y=ce^{ax}$,可两边取自然对数后,令X=x,Y=ln(y),$c=e^{B}$ 就可变换成线性表达式。当选定曲线后,可以为变量找一个合适的变换,以得到线性表达式,多种曲线如图一所示,其他一些变换如表一所示。

表一 在数据线性化中的变量变换

函数 $y=f(x)$	线性变换形式 $Y=AX+B$	变量与常数的变化
$y=\frac{A}{x}+B$	$y=A\frac{1}{x}+B$	$X=\frac{1}{x},Y=y$
$y=\frac{D}{x+C}$	$y=\frac{-1}{C}(xy)+\frac{D}{C}$	$X=xy,Y=y,C=\frac{-1}{A},D=\frac{-B}{A}$
$y=\frac{1}{Ax+B}$	$\frac{1}{y}=Ax+B$	$X=x,Y=\frac{1}{y}$
$y=\frac{x}{Ax+B}$	$\frac{1}{y}=A\frac{1}{x}+B$	$X=\frac{1}{x},Y=\frac{1}{y}$
$y=A\ln(x)+B$	$y=A\ln(x)+B$	$X=\ln(x),Y=y$
$y=Ce^{Ax}$	$\ln(y)=Ax+\ln(C)$	$X=x,Y=\ln(y),C=e^{B}$
$y=Cx^{A}$	$\ln(y)=A\ln(x)+\ln(C)$	$X=\ln(x),Y=\ln(y),C=e^{B}$
$y=\frac{1}{(Ax+B)^2}$	$y^{-\frac{1}{2}}=Ax+B$	$X=x,Y=y^{-\frac{1}{2}}$
$y=Cxe^{-Dx}$	$\ln\left(\frac{y}{x}\right)=-Dx+\ln(C)$	$X=x,Y=\ln\left(\frac{y}{x}\right),C=e^{B},D=-A$
$y=\frac{L}{1+Ce^{-Ax}}$	$\ln\left(\frac{L}{y}-1\right)=Ax+\ln(C)$	$X=x,Y=\ln\left(\frac{L}{y}-1\right),C=e^{B}$

3 程序实现

3.1程序代码

```
# -*- coding:utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt

x1 = np.array([1, 2, 3, 4, 5, 6, 7, 8])
x2 = np.array([2, 3, 4, 5, 7, 8, 10, 11, 14, 15, 16, 18, 19])
y1 = np.array([15.3, 20.5, 27.4, 36.6, 49.1, 65.6, 87.8, 117.6])
y2 = np.array([106.42, 108.20, 109.58, 109.50, 110.00, 109.93, 110.49,
110.59, 110.60, 110.90, 110.76, 111.00, 111.20])
x = x1
y = y1

print('经验公式:')
print('1、y = a+b*x')
print('2、y = a+b*x^2')
print('3、y = a*e^(bx)')
print('4、y = a*x^b')
print('5、y = a+b/x')

def fit(x, y, choose):
    if (choose == 1):
        u = y
        v = x
    elif (choose == 2):
        u = y
        v = x*x
    elif (choose == 3):
        u = np.log(y)
        v = x
    elif (choose == 4):
        u = np.log(y)
        v = np.log(x)
    elif (choose == 5):
        u = y
        v = 1.0/x

    #u = A + Bv
    n = v.shape[0]
    v1 = np.ones((n))
    v2 = v
    V = np.column_stack((v1, v2))
    VT = np.transpose(V)
    #VT*V*alpha = VT*u
    alpha = np.linalg.solve(np.dot(VT, V), np.dot(VT, u))
    A = alpha[0]
    B = alpha[1]

    u = A + B*v
    testX = np.linspace(x.min(), x.max(), 100)
    if (choose == 1):
        yfit = u
        a = A
        b = B
    elif (choose == 2):
        yfit = u
        a = A
        b = B
    elif (choose == 3):
        yfit = np.e**u
        a = np.e**A
        b = B
    elif (choose == 4):
        yfit = np.e**u
        a = np.e**A
        b = B
    elif (choose == 5):
        yfit = u
        a = A
        b = B
    Q = sum((yfit-y)**2)
    return (yfit, Q)

#----------main------------
plt.plot(x, y,    'o', color = 'r', label = u'data')
minChoose = 1
minQ = 9999999
for choose in range(1, 6):
    label = ['',
             r'$y^* = a + bx$  ',
             r'$y^* = a + bx^2$',
             r'$y^* = ae^{bx}$   ',
             r'$y^* = ax^b$     ',
             r'$y^* = a + \frac{b}{x}$    ']
    color = ['',
             'blue',
             'green',
             'grey',
             'yellow',
             'pink']
    ystar = fit(x, y, choose)
    if (ystar[1]<minQ):
        minQ = ystar[1]
        minChoose = choose
    plt.plot(x, ystar[0], '-', lw = 1.5, color = color[choose],
label = label[choose]+' Q = %.2f' %ystar[1])
plt.plot(x, y,    'o', color = 'r')
plt.legend(loc = 'upper left', fontsize = '10')
plt.show()

label = ['',
         r'$y^* = a + bx$  ',
         r'$y^* = a + bx^2$',
         r'$y^* = ae^{bx}$   ',
         r'$y^* = ax^b$     ',
         r'$y^* = a + \frac{b}{x}$    ']
color = ['',
         'blue',
         'green',
         'grey',
         'yellow',
         'pink']
choose = minChoose
ystar = fit(x, y, choose)
plt.plot(x, y,    'o', color = 'r', label = u'data')
plt.plot(x, ystar[0], '-', lw = 1.5, color = color[choose], label =
label[choose]+' Q = %.2f' %ystar[1])
plt.legend(loc = 'upper left', fontsize = '8')
plt.show()
```

3.2程序介绍

该python程序实现了对输入(x,y)数据，选择$y = a+bx$，$y = a+bx^2$，$y = ae^{bx}$，$y = ax^b$，$y = a + \frac{b}{x}$五种经验函数，分别进行拟合，绘图输出。并且选择这五种中误差最小的一种，单独进行绘图。

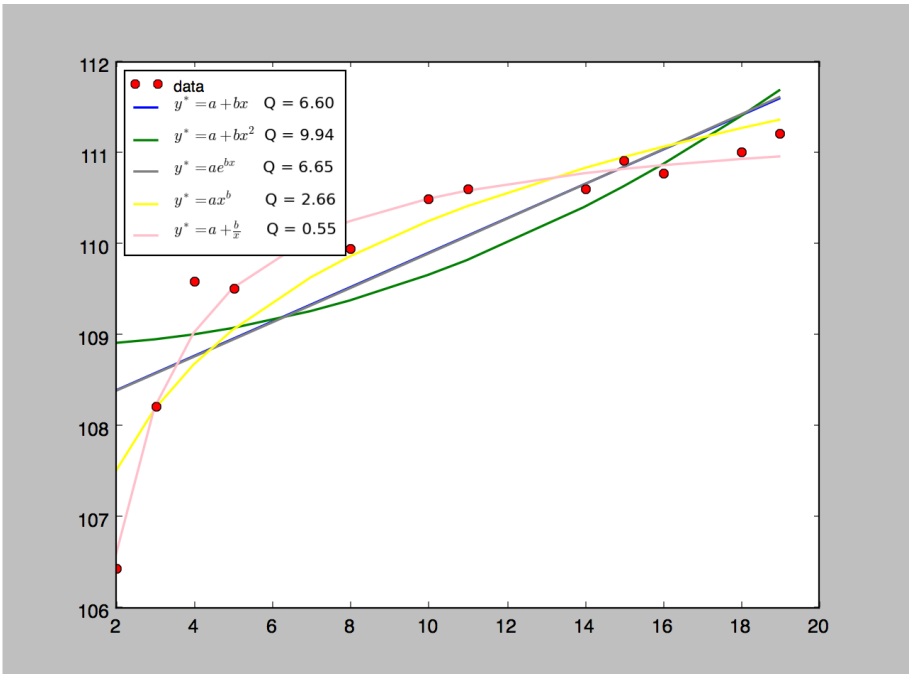
3.3程序结果

例1（书中P55例3）：

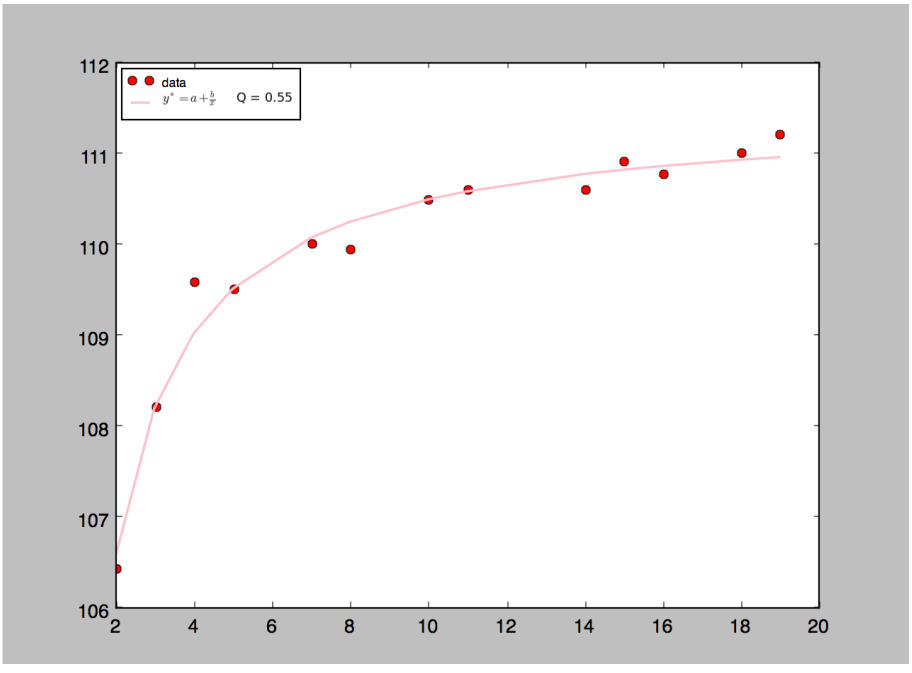
炼钢厂出钢时所用的盛钢水的钢包，在使用过程中，由于钢液及炉渣对包衬耐火材料的浸蚀，使其容积不断增大。下面列举某钢包的容积与使用次数的关系。

x使用次数	2	3	4	5	7	8	10	11	14	15	16	18	19
y容积	106.42	108.20	109.58	109.50	110.00	109.93	110.49	110.59	110.60	110.90	110.76	111.00	111.20

拟合结果如下图



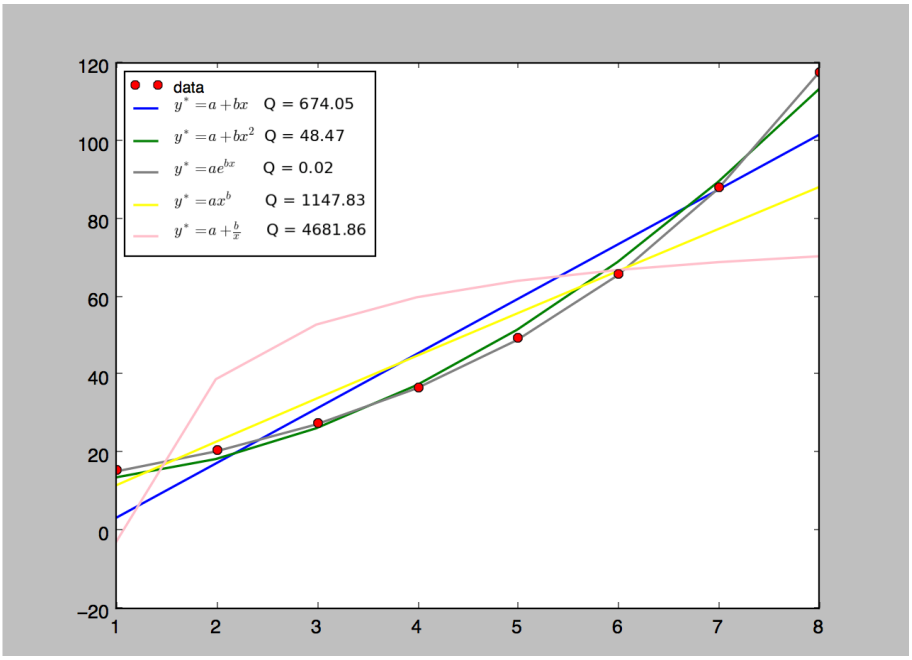
可见误差最小的曲线是 $y = a + \frac{b}{x}$，单独绘图如下



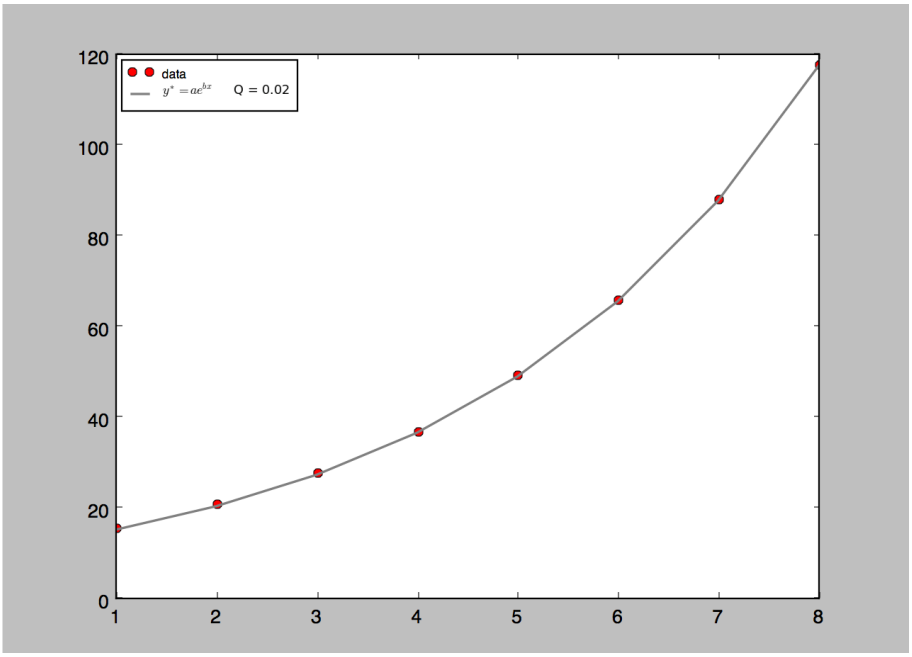
例2（书中P56例4）：

x	1	2	3	4	5	6	7	8
y	15.3	20.5	27.4	36.6	49.1	65.6	87.8	117.6

拟合结果如下图



可见误差最小的曲线是 $y = ae^{bx}$，单独绘图如下



4 总结

（1）当变量之间不成简单的线性关系时，通过数据线性化之后，使得非线性问题转化成线性问题，可以得到较好的拟合结果。

（2）最小二乘法拟合也可以用于海洋中的潮汐调和分析。但是由于时间原因，使用最小二乘法进行潮汐调和分析的python程序尚未完成。

(3)通过自己编写程序实现最小二乘法拟合，对最小二乘法的原理有了更深的理解，也留下了今后可以使用的拟合python代码资源。

5 参考文献：

[1]赵宝贵. Matlab在数据拟合中的应用[J]. 科技广场,2007,1:145-146.

[2]欧阳明松,徐连民. 基于MATLAB的试验数据拟合[J]. 南昌工程学院学报,2010,4:24-28.

[3]张庆.MATLAB 语言在非线性最小二乘估计中的应用[J].测绘与空间地理信息, 2004,27(3).

[4]唐家德.基于matlab的数据线性化变换[J].科技广场,2007,5:44-48.

[5] 徐萃薇,孙绳武. 计算方法引论[M]. 北京:高等教育出版社,2007.