MATLAB最小二乘法

作者：凯鲁嘎吉 - 博客园
http://www.cnblogs.com/kailugaji/

一、实验目的

对于一组数据$(x_i, y_i)(i=1,2,\cdots,m)$,这些数据在$xOy$平面直角坐标系下对应点的趋势，求出与这些点适当接近的曲线所对应的函数,以达到用曲线逼近数据点的目的，称为拟合。这次实验,主要是用多项式拟合所给的数据。采用的方法称为最小二乘法。

二、实验原理

设$(x_i,y_i)(i=1,2,\cdots,m)$,选取$n$次多项式$P(x)=\sum_{j=0}^{n} a_j x^j$,记$S=\sum_{i=1}^{m}[P(x_i)-y_i]^2$,选取$a_j$使$S$达到最小值,这种方法就是最小二乘法.令

$$\frac{\partial S}{\partial a_k}=\sum_{i=1}^{m}[P(x_i)-y_i]\frac{\partial P(x_i)}{\partial a_k}=0,\quad k=0,1,2\cdots,n.$$

即 $$\sum_{i=1}^{m}[\sum_{j=0}^{n}a_j x_i^j-y_i]x_i^k=\sum_{j=0}^{n}a_j(\sum_{i=1}^{m}x_i^{j+k})-\sum_{i=1}^{m}x_i^k y_i=0, k=0,1,2\cdots,n$$

令$S_l=\sum_{i=1}^{m}x_i^l$, $t_k=\sum_{i=1}^{m}x_i^k y_i$,则上式化为$\sum_{j=0}^{n}S_{j+k}a_j=t_k, k=0,1,2\cdots,n.$

或 $$AX=b. \qquad (*)$$

其中 $$A=\begin{pmatrix} m & \sum_{i=1}^{m}x_i & \sum_{i=1}^{m}x_i^2 & \cdots & \sum_{i=1}^{m}x_i^n \\ \sum_{i=1}^{m}x_i & \sum_{i=1}^{m}x_i^2 & \sum_{i=1}^{m}x_i^3 & \cdots & \sum_{i=1}^{m}x_i^{n+1} \\ \sum_{i=1}^{m}x_i^2 & \sum_{i=1}^{m}x_i^3 & \sum_{i=1}^{m}x_i^4 & \cdots & \sum_{i=1}^{m}x_i^{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum_{i=1}^{m}x_i^n & \sum_{i=1}^{m}x_i^{n+1} & \sum_{i=1}^{m}x_i^{n+2} & \cdots & \sum_{i=1}^{m}x_i^{2n} \end{pmatrix}_{(n+1)\times(n+1)}, b=\begin{pmatrix} \sum_{i=1}^{m}y_i \\ \sum_{i=1}^{m}x_i y_i \\ \sum_{i=1}^{m}x_i^2 y_i \\ \vdots \\ \sum_{i=1}^{m}x_i^n y_i \end{pmatrix}.$$

显然A是$n+1$阶对称矩阵.方程组(*)称为最小二乘法的正规方程组,或正则方程组.解之得$a_j(j=1,2,\cdots,n)$便可得最小二乘拟合多项式$P(x)$.

三、实验程序

输入　数据$(x_i, y_i)(i=1,2,\cdots,m)$.

Step 1　对$k=0,1,2,\cdots,n$,做Step 2

Step 2　对$j=1,2,\cdots,m$做

$$A[k,j]=\sum_{i=1}^{m} x_i^k$$

$$b[j]=\sum_{i=1}^{m} x_i^k y_i$$

Step 3　解$Ax=b$,其中$x=(a_0,a_1,\cdots,a_n)^T$ ——用直接或选列主元的紧凑格式等方法

输出　$P(x)$.

四、实验内容

设有如下数据:

x_j	-3	-2	-1	0	1	2	3
$f(x_j)$	-1.76	0.42	1.2	1.34	1.43	2.25	4.38

用3次多项式拟合这组数据.

五、解答(按如下顺序提交电子版)

1.(程序)

LSM1.m:

```
function p=LSM1(x,y,m)  %x,y为序列长度相等的数据向量，m为拟合多项式次数
format short;
A=zeros(m+1,m+1);
for i=0:m
    for j=0:m
        A(i+1,j+1)=sum(x.^(i+j));
    end
```

```
    b(i+1)=sum(x.^i.*y);
end
a=A\b';
p=fliplr(a');
```

2.(运算结果)

```
>> x=[-3,-2,-1,0,1,2,3];
>> y=[-1.76,0.42,1.2,1.34,1.43,2.25,4.38];
>> p=LSM1(x,y,3)

p =

    0.1133   -0.0018    0.0035    1.3300
```

3.(拓展（方法改进、体会等）)

MATLAB中有关于最小二乘法的现成的函数，如下编写程序：

ploy1.m：

```
function [p]=ploy1(x,y)
P=polyfit(x,y,3);
xi=-4:.2:4;
p=polyfit(x,y,3);
yi=polyval(P,xi);
plot(xi,yi,x,y,'r*');
```

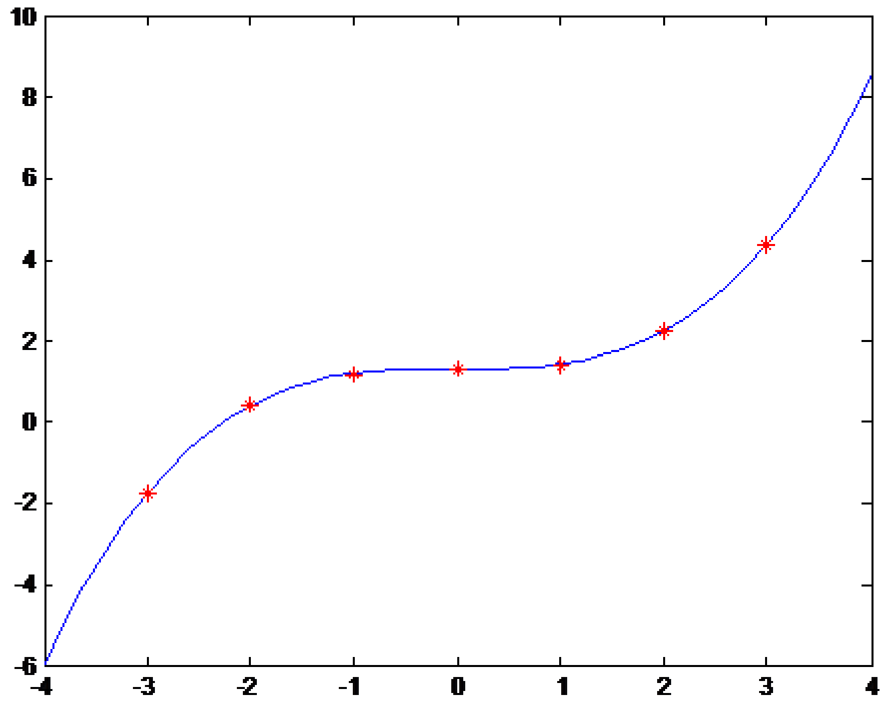
结果：

```
>> x=[-3,-2,-1,0,1,2,3];
y=[-1.76,0.42,1.2,1.34,1.43,2.25,4.38]; [p]=ploy1(x,y)

p =

    0.1133   -0.0018    0.0035    1.3300
```

则y=0.1133*x^3-0.0018*x^2+0.0035*x+1.33



注：非线性曲线拟合见：MATLAB实例：非线性曲线拟合