统计推断在数模转换系统中的应用

组号 36　　　姓名 赵怿龙 学号 5140309296，姓名 刘韫彧 学号 5140309297

摘要：本报告以实验数据为基础，探讨统计推断在数模转换系统中的应用。本报告的研究对象是某产品中的一个检测模块，该监测模块中传感器部件的输入输出特性呈明显的非线性。本课题要求为该模块的批量生产设计一种成本合理的传感特性校准（定标工序）方案。由于每次的输入输出测定成本较高，因此需要用较少的测量次数得到较高的校准精度。
关键词：多项式拟合、样条插值拟合、模拟退火算法

1 引言

本课程研究的问题的背景是在某种电子元件的生产中，其内部有一个传感器模块，但其输入与输出明显不成线性，因此需要校准。校准过程中，已经测得了大量样本，但是单个样本不足以代表整体的输入输出关系，因此需要将大量的样本进行拟合。由于测定次数为有限次数，故样本为离散的变量。本课程研究的问题则是对此拟合过程提出一种较好的方案。

2 标准样本数据

以几个传感器的 X-Y 图像为例

由图中可看出，样本数据具有以下特点：

(1)单个传感器的样本 Y 随 X 单调递增；

(2)X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；

(3)不同的传感器的图像趋势相似但各有差异；

(4)由整体的形状大致可以分成首中尾三段，其中中段斜率略小于另两段；

(5)各个传感器的中段起止有随机差异。

3 特征点的选取

3.1 概述[1]

为评估和比较不同的校准方案，特制定以下成本计算规则。

1.单点定标误差成本

$$
s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.4\\ 0.1 & \text{if } 0.4<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.6\\ 0.7 & \text{if } 0.6<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.8\\ 0.9 & \text{if } 0.8<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1.5 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 6 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 12 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 25 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \tag{4-1}
$$

单点定标误差的成本按式（4-1）计算，其中表示第 i 个样本之第 j 点 Y 的实测值，表示定标后得到的估测值（读数），该点的相应误差成本以符号记。

2.单点测定成本

实施一次单点测定的成本以符号 q 记。本课题指定 q=12。

3.某一样本个体的定标成本

$$
S_{i,j}=\sum_{j=1}^{51}S_{i,j}+q\bullet n_i \tag{4-2}
$$

对样本 i 总的定标成本按式（4-2）计算，式中表示对该样本个体定标过程中的单点测定次数。

4.校准方案总成本

按式（4-3）计算评估校准方案的总成本，即使用该校准方案对标准样本库中每个样本个体逐一定标，取所有样本个体的定标成本的统计平均。

$$
C=\frac{1}{M}\sum_{i=1}^{M}S_i \tag{4-3}
$$

总成本较低的校准方案，认定为较优方案。

3.3 模拟退火算法

特征点选择需要一种算法。首先可以通过暴力穷举法通过数学方法可以得出，计算需要达到亿次以上，过于复杂因此不可行。其次，遗传算法尽管结果较好，但需要代码太过复杂

模拟退火算法来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部粒子随温升变为无序状，内能增大，而冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。

模拟退火算法流程图如下：

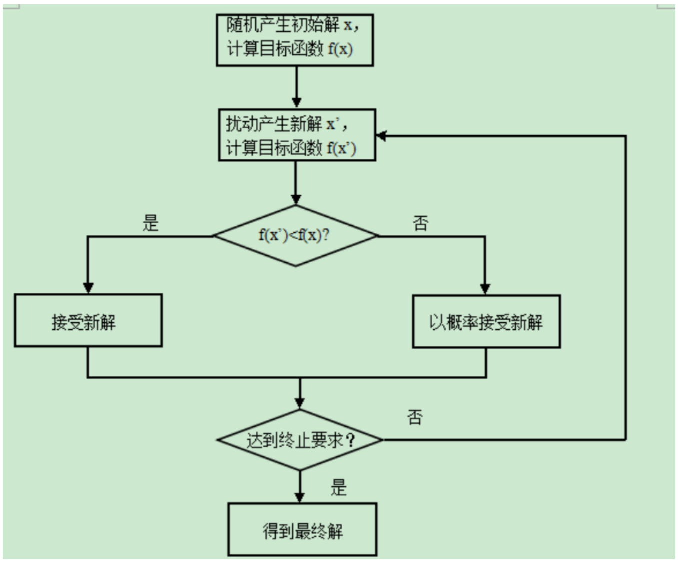

图 4-1 模拟退火算法流程图

此实验中接受新解的概率为 exp((t-t’)/T)。由于实验中发现模拟退火算法结果的随机性比较大，因此需要多次试验以选取成本最低的特征点。

4 拟合方式

以第一组为例。首先利用 matlab 绘制出 XY 图线如下：

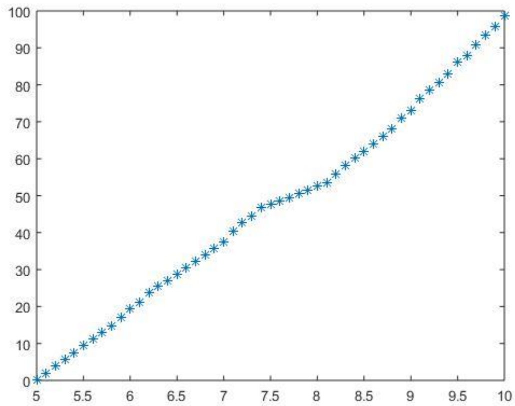

图 3-1：第一组实验数据的图像

4.1 多项式拟合

多项式拟合在生活中十分常见，通过计算机利用最小二乘法很容易获得较好的拟合函数。而且在 matlab 中有多项式拟合的函数，可以直接利用，因此多项式拟合很方便。

观察图像，可得其大致分为三段，与三次图线类似，因此可以至少将其拟合成三次图线。我们尝试对第一组进行拟合尝试，发现在 3-5 次多项式拟合时图像比较接近，但次数再增加时，拟合曲线震荡变大，与数据图像的差异也变大。经百度得这被称为龙格现象。因此实验中可以拟合成二次，三次，四次，五次，六次图线来观察成本，选取最优解。

4.2 样条插值拟合

多项式拟合是将整个函数用一个函数式来拟合，在次数较高时可能会产生龙格现象。因此，如果使用分段的拟合方式是否会更准确呢？于是想到了样条插值拟合方法。

样条插值拟合方法为，选取的七个特征点中每相邻两个和与二者相邻的两点，共四点，

进行计算参数，得到三次函数，将此函数作为此相连的两点间的拟合函数。由此得到的函数在任何一点都是光滑的。在 matlab 中也有此种拟合方式的函数，也使实验方便不少。

5 拟合方式的最终选择

5.1 取点算法的实现

最终确定的是利用模拟退火算法，在实验中具体操作如下：

(1)定义并初始化变量：初始温度 Tk=100，室温 Tc = 0.1，初始解特征点集 S=[1,10,18,26,34,43,51]，每次个温度迭代次数 L=1；

(2)对 k 取 1 至 L 做第(3)至第(5)步：

(3)计算内能增量 ΔE′ =C(S′)-C(S)，其中 C(S)为成本的计算函数；

(4)如果 ΔE′ <0 则 S′ 作为新的当前解，否则以概率 exp(-ΔE′ /T)接受 S′ 作为新的当前解；

(5)随机扰动，产生新解 S′ ；

(6)T 降温，T=T*0.97。

5.2 拟合方式的选取

5.2.1 多项式拟合

首先选用多项式拟合进行试验，几次试验后发现，模拟退火算法每次实验得到结果差异较大，也就是随机性较大。比如说以三次多项式拟合为例，结果多在 130 到 140 之间，但有时也会出现成本这样的数据。为寻找最优的特征点，因此我们用对每种拟合方式的 20 次的实验结果以选取成本最小的特征点。决定试验步骤如下：

(1)分别对 2，3，4，5，6 次多项式拟合进行步骤(2)至(5)：

(2)对每组数据，对选取的初始特征点计算拟合函数；

(3)由每组数据与拟合函数计算成本；

(4)对初始特征点进行模拟退火算法，直至选出最优特征点；

(5)重复(2)至(4)步 20 次，选出使平均成本最小的特征点。

5.2.2 三次样条插值拟合

之后对三次样条插值拟合法进行实验，同样对其 20 次的实验结果取最低值进行评价拟合方式的好坏。决定试验步骤如下：

(1)分别对三次样条插值拟合进行步骤(2)至(5)：

(2)对每组数据，对选取的初始特征点计算拟合函数；

(3)由每组数据与拟合函数计算成本；

(4)对初始特征点进行模拟退火算法，直至选出最优特征点；

(5)重复(2)至(4)步 20 次，选出使成本最小的特征点。

5.2.3 拟合方式的确定

进行以上步骤，对三次，四次，五次，六次多项式拟合以及三次样条插值拟合进行试验后，得到如下结果：

(1)二次多项式拟合：最低成本 260.33，特征点[5,17,23,29,41,43,48]，运行时间 166.17s；

(2)三次多项式拟合：最低成本 136.53，特征点[3,12,21,27,35,44,48]，运行时间 77.44s；

(3)四次多项式拟合：最低成本 122.58，特征点[2,7,13,22,29,40,51]，运行时间 133.27s；

(4)五次多项式拟合：最低成本 119.17，特征点[1,3,10,21,31,45,48]，运行时间 144.50s；

(5)六次多项式拟合：最低成本 128.41，特征点[3,7,16,25,40,49,51]，运行时间 191.85s；

(6)三次样条插值拟合：最低成本 99.40，特征点[1,9,20,29,33,41,50]，运行时间 29.95s。

由此可见三次样条插值拟合效果较好，多项式拟合中，二次到五次多项式拟合成本降低，但当次数增加到六次时，成本却并没有明显减小，出现龙格现象。而三次样条插值拟合比其他都低,而且运行时间极快，故选取三次样条插值拟合作为最终的拟合方式。

对于三次样条插值拟合，成本最低的一次试验所得的成本与退火次数的关系图如下

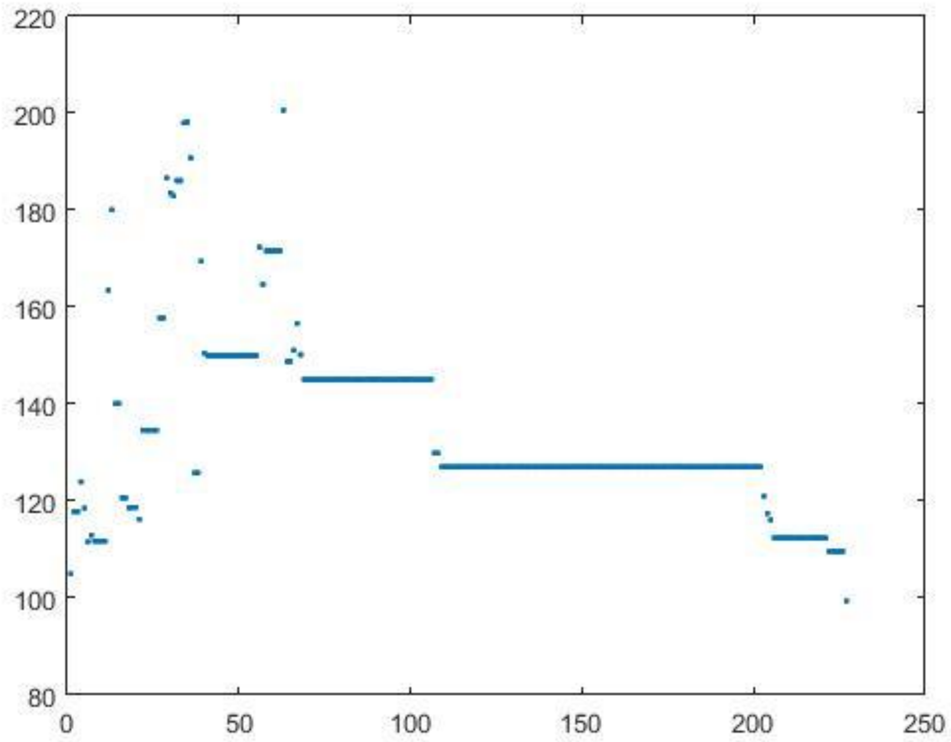

图 5-1:成本与退火次数的关系图

可见，退火 0-50 次时，成本上下跳动较大，因为此时温度较高，当新解大于旧解时容易被接受，而 50 次以后，温度较低，新解大于旧解时接受的概率较小，从而产生收敛的趋势。

5.3 特征点的选取

5.3.1 特征点个数的选取

对确定好的拟合方式，分别在选取 5，6，7，8，9 个特征点时，对三次样条插值拟合进行 20 次实验，选取最低成本作为评价拟合点个数方案的好坏，结果如下：

(1)5 个特征点：初始特征点[1,12,24,37,49]最低成本 114.12，特征点[5,11,22,39,49]，运行时间 27.85s；

(2)6 个特征点：初始特征点[1,11,21,31,41,51]最低成本 98.02，特征点[2,10,22,30,42,51]，运行时间 32.64s；

(3)7 个特征点：初始特征点[1,9,17,26,35,44,51]最低成本 99.40，特征点[1,9,20,29,33,41,50]，运行时间 29.95s；

(4)8 个特征点：初始特征点[1,8,15,22,29,36,43,51]最低成本 108.17，特征点[4,9,21,29,37,38,44,50]，运行时间 28.13s；

(5)9 个特征点:初始特征点[1,7,13,20,26,32,39,45,51]最低成本 113.69,特征点[2,6 14,21,26,32,42,46,50]，运行时间 34.74s。

可见，当选取 6-7 个特征点时成本较小。虽然随着特征点选取数量的增多，定标成本降低，但是 8 个特征点以上时测定成本变高，导致总成本升高，因此，特征点选 6-7 个为宜，最终决定选取 6 个特征点。

5.3.2 退火算法迭代次数 L 的确定

此前实验选取的 L 为 1，此时结果随机性较大，因此想得到成本较低的特征点，少量实验很可能得出错误的结果。如果增加 L 的大小，是否会使结果的随机性下降呢，于是我们选取不同的 L 值进行对三次样条插值拟合，6 个特征点情况进行退火实验，每个 L 值实验 10 次，求取结果成本的方差，结果如下：

(1)L=1：成本方差 555.81；

(2)L=2：成本方差 37.39；

(3)L=3：成本方差 89.10；

(4)L=4：成本方差 19.97；

(5)L=5：成本方差 8.93；

(6)L=10：成本方差 11.84。

L 为不同值的一次试验成本收敛情况随退火次数的关系如下图，可见收敛明显加快。但是同时 L 增加，也就是说未收敛区域中实际上是考虑了更多的解，也就是说容易跳出某个局部最优解较差的区域，跳入局部最优解更优的区域。但是运行时间成倍增加，因此为获得整体最优解，我们还是决定通过 L 取 10 并多次实验的方法。

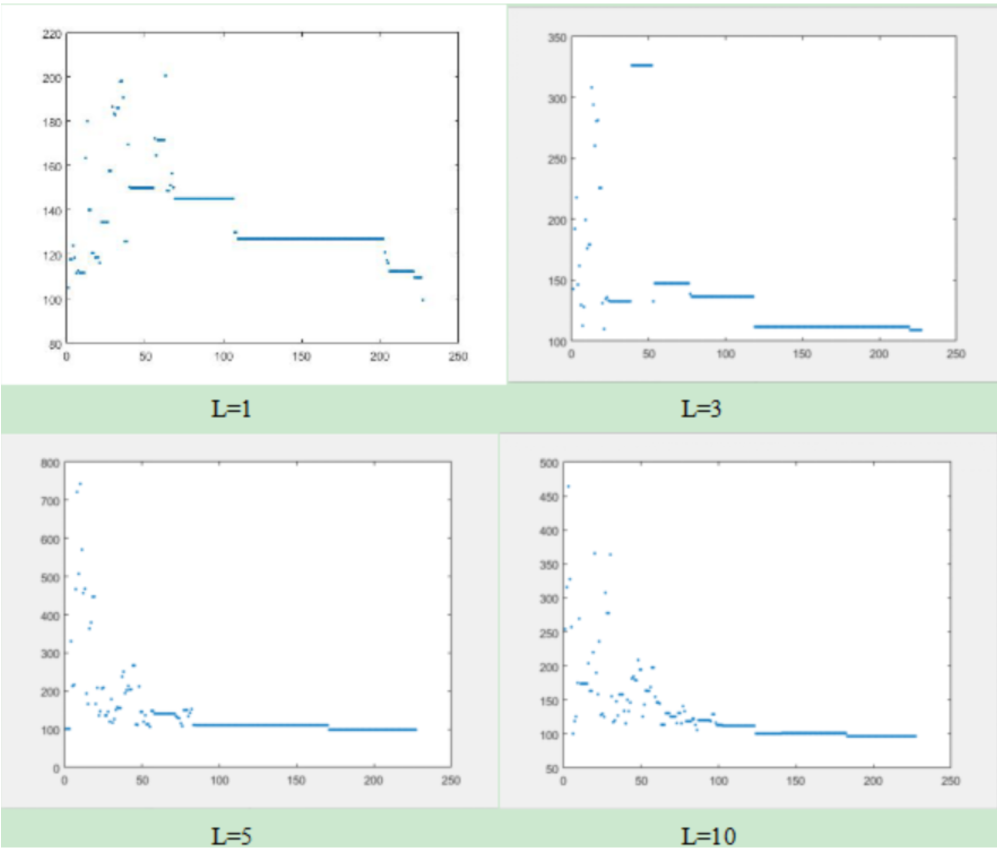

图 5-2：不同 L 值下成本与退火次数的关系图

6 实验结果

确定好拟合方式为三次样条插值拟合，特征点选取个数为 6 个，L=10 后，进行 10 次退火，以选取几组最优的结果，结果如下：

拟合方式：三次样条插值拟合

特征点：［3, 11, 23, 32, 44, 49］

成本 96.83

运行时间 314.53s

7 参考文献

[1] 上海交大电子工程系统计推断在模数转换系统中的应用课程设计课题和要求(V2.2 2015-9-22 更新). ftp://202.120.39.248

8 附录

8.1 老师提供的代码

本代码中的数据是本小组的最终结果数据。

```
%%%%%%% 答案检验程序 2015-11-04 %%%%%%%%

my_answer=[  3,11,23,32,44,49  ];%把你的选点组合填写在此
my_answer_n=size(my_answer,2);

% 标准样本原始数据读入
minput=dlmread('20150915dataform.csv');
[M,N]=size(minput);
nsample=M/2; npoint=N;
x=zeros(nsample,npoint);
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
for i=1:nsample
    x(i,:)=minput(2*i-1,:);
    y0(i,:)=minput(2*i,:);
end
my_answer_gene=zeros(1,npoint);
my_answer_gene(my_answer)=1;


% 定标计算
index_temp=logical(my_answer_gene);
x_optimal=x(:,index_temp);
y0_optimal=y0(:,index_temp);
for j=1:nsample
    % 请把你的定标计算方法写入函数 mycurvefitting
    y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

% 成本计算
Q=12;
errabs=abs(y0-y1);

le0_4=(errabs<=0.4);
le0_6=(errabs<=0.6);
le0_8=(errabs<=0.8);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
```

```
le5_0=(errabs<=5);
g5_0=(errabs>5);

sij=0.1*(le0_6-le0_4)+0.7*(le0_8-le0_6)+0.9*(le1_0-le0_8)+1.5*(le2_0-
le1_0)+6*(le3_0-le2_0)+12*(le5_0-le3_0)+25*g5_0;
si=sum(sij,2)+Q*ones(nsample,1)*my_answer_n;
cost=sum(si)/nsample;

% 显示结果
fprintf('\n 经计算，你的答案对应的总体成本为%5.2f\n',cost);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y1 = mycurvefitting( x_premea,y0_premea )

x=[5.0:0.1:10.0];

% 将你的定标计算方法写成指令代码，以下样式仅供参考
%y1=interp1(x_premea,y0_premea,x,'spline');
Yfit = polyfit(x_premea,y0_premea,2);
y1=polyval(Yfit,x);


end
```

8.2 本小组的程序代码

1. 模拟退火算法

```
warning off all
format long
%定义初始特征点矩阵 charp
charp = [1,11,21,31,41,51];%此处输入初始特征点矩阵
charpl = charp;
numofpnt=size(charp,2);
avecstl = 10000;
avecst = 0;
fcharp = [];
favecst = 10000;
favecost = [];
fcstl = [];
numofexp = 1;
for iii = 1:numofexp
   tic;
   avecost = [];
   i = 1;
   %模拟退火过程
```

```
Tk = 100;%初始温度
Te = 0.1;%结束温度
L = 10;%每个温度下迭代次数
while Tk>Te
    for kk = 1:L
        %计算新解成本
        avecst = culavecst(charp);
        %判断是否接受新解
        if avecst<=avecstl
            charpl = charp;
            avecstl = avecst;
        else
            if rand(1,1)<(exp((avecstl-avecst)/Tk))
                charpl = charp;
                avecstl = avecst;
            end
        end
        %随机扰动，以获得新解
        k = randi([1,numofpnt],1,1);
        if(k>1&&k<numofpnt)
            charp(k) = randi([charp(k-1)+1,charp(k+1)-1],1,1);
        end
        if(k==1)
            charp(k) = randi([1,charp(2)-1],1,1);
        end
        if(k==numofpnt)
            charp(k) = randi([charp(numofpnt-1)+1,51],1,1);
        end
    end
    %退火
    avecost(i) = avecstl;
    i = i+1;
    Tk = Tk*0.97;
end
%输出单次试验结果
fprintf('\n\n 第%d 次实验结果\n 特征点',iii);%输出实验次数特征点结果成本
disp(charpl)
fprintf('成本%5.2f\n',avecstl);
fcstl(iii) = avecstl;
if avecstl<favecst
    fcharp = charpl;
    favecst = avecstl;
    favecost = avecost;
end
```

```
    plot(avecost,'.')%绘制此次试验成本的收敛图
    toc%输出此次试验时间
end
%输出最终试验结果
fprintf('\n\n 最佳实验结果\n 特征点');%输出最终特征点结果成本
disp(fcharp)
fprintf('成本%5.2f,各次结果方差%5.2f\n',favecst,var(fcstl));
favecost;
plot(favecost,'.')%绘制成本最低一次试验的成本收敛图
```

2. culavecst 函数实现

```
function [rst] = culavecst(charp)

my_answer = charp;
my_answer_n = size(my_answer,2);

% 标准样本原始数据读入
minput = dlmread('20150915dataform (1).csv');
[M,N] = size(minput);
nsample = M/2;
npoint = N;
x = zeros(nsample,npoint);
y0 = zeros(nsample,npoint);
y1 = zeros(nsample,npoint);
for i=1:nsample
    x(i,:) = minput(2*i-1,:);
    y0(i,:) = minput(2*i,:);
end
my_answer_gene = zeros(1,npoint);
my_answer_gene(my_answer) = 1;


% 定标计算
index_temp = logical(my_answer_gene);
x_optimal = x(:,index_temp);
y0_optimal = y0(:,index_temp);
for j=1:nsample
    xxx = [5.0:0.1:10.0];
    %此处为拟合方式
    %三次样条插值拟合
    y1(j,:)=interp1(x_optimal(j,:),y0_optimal(j,:),xxx,'spline');
    %多项式拟合
    %Yfit = polyfit(x_optimal(j,:),y0_optimal(j,:),2);%最后一个系数为多项
式次数
    %y1(j,:) = polyval(Yfit,xxx);
```

```
end

% 成本计算
Q = 12;
errabs = abs(y0-y1);

le0_4 = (errabs<=0.4);
le0_6 = (errabs<=0.6);
le0_8 = (errabs<=0.8);
le1_0 = (errabs<=1);
le2_0 = (errabs<=2);
le3_0 = (errabs<=3);
le5_0 = (errabs<=5);
g5_0 = (errabs>5);

sij = 0.1*(le0_6-le0_4 ) + 0.7*(le0_8-le0_6) + 0.9*(le1_0-le0_8) +
1.5*(le2_0-le1_0) + 6*(le3_0-le2_0) + 12*(le5_0-le3_0) + 25*g5_0;
si = sum(sij,2) + Q*ones(nsample,1)*my_answer_n;
cost = sum(si) / nsample;
rst = cost;
```