统计推断在数模转换系统中的应用

组号：54　姓名：钟淋 学号：51040309411，姓名：胡轩浩 学号：5140309416

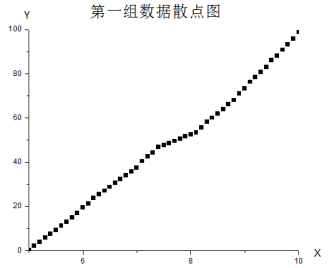
摘要：本课题目标为该模块的批量生产设计一种成本合理的传感特性校准（定标工序）方案。由于每次的输入输出测定成本较高，因此需要用较少的测量次数得到较高的校准精度。综合运用多项式拟合、三次样条插值和模拟退火算法等方法进行取点方案的选择，最终提出一种合理的校准方案。

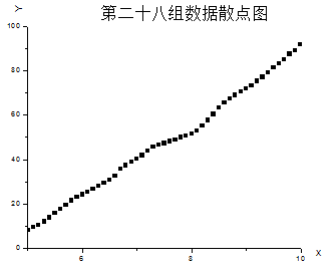
关键词：统计推断，拟合,三次样条插值，模拟退火算法

1、引言

为某产品内部的一个测量模块寻求定标工序的优选方案，已获得 400 个样品的测定数据（标准样本库），利用拟合、插值的数学原理和方法和搜索算法、启发式搜索算法、遗传算法的计算机辅助分析和求解算法寻找 X-Y 近似关系形式进行定标，并利用成本计算公式对定标方案进行评估。本课题要求为某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。

2、数据样本的分析





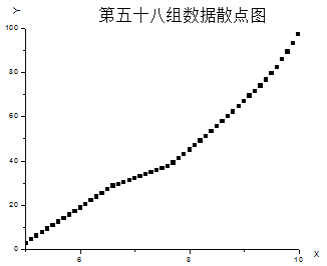


图 2-1 第 1 组数据散点图　图 2-2 第 28 组数据散点图　图 2-3 第 58 组数据散点图

结合课件中给出的几组数据的散点图，再结合由我们随机选取的三组数据画出的散点图，我们可以大致得出 X-Y 特性曲线是单调递增的，X 取值在 5 － 10 之间；Y 取值大致在 0 － 100 之间，个体样品的特性曲线形态相似但两两相异，大致都可以分为三段，三段都不是线性的，有一定弯曲度，中段的斜率小于首段和尾段的斜率，且中段的起点位置和长度都带有随机性。

3、成本计算函数[1]

单点定标误差成本

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

下标 i 代表样品序号
下标 j 代表观测点序号

对某一样品 i 的定标成本

$$S_i=\sum_{j=1}^{51}s_{i,j}+12N_i \tag{2}$$

Ni 是该样品定标时所测定的点数
第一项：误差成本
第二项：测定成本

定标方案总成本

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

在本次课题中，算法中的成本计算函数我们采用了老师提供的成本计算标准函数。

4、拟合方法

4.1、多项式拟合

多项式拟合是较常见的拟合方法，根据最小二乘法，由已知的的若干数据点得到曲线的方程。

4.2、三次样条插值法

三次样条插值拟合，即用若干段三次曲线来近似表示。选出若干个特征点，对于中间的点而言，取中间点加上两侧的与中间点相邻的两个点作为研究对象。通过四个点确定出函数关系方程作为中间点之间曲线方程，以此类推。例如，对于一列点 S1、S2、S3、S4、S5、S6 来说，对于 S3、S6 之间而言，由 S3、S4、S5、S6 确定一条 3 次曲线，而后仅在 S4 与

S5 之间以该曲线表示，其它点同理。而对于边缘上的点而言，如对点 S1 和 S2，由 S1、S2、S3 确定一条 2 次曲线，而后仅在 S1 与 S2 之间以该曲线表示。其优点在于它对于相邻两点之间都采用一条 2 次或 3 次曲线来表示,这样就能使整体曲线非常平滑,且准确度高。

经过分析比较以及上一届同学的做法，我们得出，三次样条插值算法比三次多项式拟合误差更小，因此在本次课题中，我们选择三次样条插值算法进行拟合。

5、启发式算法

5.1、遗传算法[2]

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰 遗传机制）演化而来的随机化搜索方法。其主要特点是直接对结构对象进行操作，不存在 求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻 优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。

算法实现过程：

a)初始化：设置进化代数计数器 t=0，设置最大进化代数 T，随机生成 M 个个体作为初始群体 P(0)。

b)个体评价：计算群体 P(t)中各个个体的适应度。

c)选择运算:将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代 或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评 估基础上的。

d)交叉运算：指把两个父代个体的部分结构加以替换重组而生成新个体的操作。

e)变异运算:将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因 值作变动。群体 P(t)经过选择、交叉、变异运算之后得到下一代群体 P(t+1)。

可以有以下的算法：

a)实值变异

b)二进制变异。

一般来说，变异算子操作的基本步骤如下：

a)对群中所有个体以事先设定的变异概率判 断是否进行变异。

b)对进行变异的个体随机选择变异位进行变异。

f)终止条件判断:若 t=T,则以进化过程中所得到的具有最大适应度个体作为最优解输出，终止计算。

优缺点分析：此方法借鉴生物进化理论，通过选择，交叉配对，变异和淘汰，可以得到最优解。并且，相比于穷举法，大大减少了计算时间。

遗传算法的一般算法：

a）建初始状态：初始种群是从解中随机选择出来的，将这些解比喻为染色体或基因， 该种群被称为第一代，这和符号人工智能系统的情况不一样，在那里问题的初始状态已经 给定了。

b）评估适应度：对每一个解(染色体)指定一个适应度的值，根据问题求解的实际接 近程度来指定(以便逼近求解问题的答案)。不要把这些“解”与问题的“答案”混为一谈， 可以把它理解成为要得到答案，系统可能需要利用的那些特性。

c）繁殖（包括子代突变）：带有较高适应度值的那些染色体更可能产生后代(后代产 生后也将发生突变)。后代是父母的产物，他们由来自父母的基因结合而成，这个过程被称 为“杂交”。

d）下一代：如果新的一代包含一个解，能产生一个充分接近或等于期望答案的输出，

那么问题就已经解决了。如果情况并非如此，新的一代将重复他们父母所进行的繁衍过程，一代一代演化下去，直到达到期望的解为止。

e）并行计算：非常容易将遗传算法用到并行计算和群集环境中。一种方法是直接把 每个节点当成一个并行的种群看待。然后有机体根据不同的繁殖方法从一个节点迁移到另 一个节点。另一种方法是“农场主/劳工”体系结构，指定一个节点为“农场主”节点，负 责选择有机体和分派适应度的值，另外的节点作为“劳工”节点，负责重新组合、变异和 适应度函数的评估。

5.2、模拟退火算法[3]

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，而徐徐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。用固体退火模拟组合优化问题，将内能 E 模拟为目标函数值 f，温度 T 演化成控制参数 t，即得到解组合优化问题的模拟退火算法：由初始解 i 和控制参数初值 t 开始，对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代，并逐步衰减 t 值，算法终止时的当前解即为所得近似最优解，这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。退火过程由冷却进度表(Cooling Schedule)控制，包括控制参数的初值 t 及其衰减因子 Δt、每个 t 值时的迭代次数 L 和停止条件 S。

模拟退火的基本思想:

(1) 初始化：初始温度 T(充分大)，初始解状态 S(是算法迭代的起点)，每个 T 值的迭代次数 L

(2) 对 k=1，……，L 做第(3)至第 6 步：

(3) 产生新解 S′

(4) 计算增量 Δt′ =C(S′)-C(S)，其中 C(S)为评价函数

(5) 若 Δt′ <0 则接受 S′ 作为新的当前解，否则以概率 exp(-Δt′ /T)接受 S′ 作为新的当前解.

(6) 如果满足终止条件则输出当前解作为最优解，结束程序。

终止条件通常取为连续若干个新解都没有被接受时终止算法。

(7) T 逐渐减少，且 T->0，然后转第 2 步。

我们在经过比较之后，选择了相对简单且快速的模拟退火算法。

6、实际算法程序与运行结果

程序采取模拟退火法作为取点方法，三次样条插值作为拟合方法。

先确定取样个数、初始温度、最低温度等参数，然后随机取样并进入循环。每次通过三次样条插值法计算出成本，与上一次成本进行比较，若低于上一次成本则接受新的取样，反之则以一定概率接受新的取样，最后改变原样本中的一个点作为新的样本并退火，直至温度低于最低温度退出循环，得出此次实验的最小成本和其所对应的的取样点。具体程序代码见附录。

最开始，我们猜测取点数为 7 时的成本可能为最低，并且考虑到实际情况，将第一个点与最后一个点设置为必取点，将程序中 n 设置为 7 后运行，得到结果如下。

表 6-1　7 点模拟退火结果

编号	运行结果	成本
1	1、6、19、25、34、43、51	98.47
2	1、9、21、25、32、43、51	97.95
3	1、7、18、26、33、44、51	97.32

但是仅仅如此我们并不能确定 7 点的结果为最优解，于是我们改变参数 n 的值，测试了 n 分别为 5,6 和 8 时的结果，结果如下所示。

表 6-2　5 点模拟退火结果

编号	运行结果	成本
1	1、13、25、39、51	112.89
2	1、12、23、39、51	113.44
3	1、13、25、40、51	113.78

表 6-3　6 点模拟退火结果

编号	运行结果	成本
1	1、10、22、30、41、51	99.08
2	1、8、20、29、44、51	101.93
3	1、7、21、31、43、51	98.97

表 6-4　8 点模拟退火结果

编号	运行结果	成本
1	1、6、16、23、32、39、46、51	105.28
2	1、10、17、23、32、38、44、51	105.33
3	1、9、19、22、29、35、45、51	104.64

从 5 点、6 点和 8 点的运行结果来看，6 点结果与 7 点相近但仍高出少许，5 点与 8 点则高出不少，因此我们确定取点数为 7 点时结果为最优解。

7、 总结

经过讨论分析与程序运行结果，我们最终确定取点数为 7 个，拟合方法为三次样条插值拟合法为成本最低的定标方法。

运行后所得的最佳取点方式为 1、7、18、26、33、44、51，成本为 97.32

8、 参考文献

[1] 袁焱. 统计推断在数模转换系统中的应用课程讲义[EB/OL].ftp://202.120.39.248.

[2] 百度百科. 遗传算法[J/OL]
.http://baike.baidu.com/link?url=6RDaZAurq6QmlJnwg3HkpxPxaGsQfeQ_rBwVClYXzbjj9_jjnxg0lvyKOvivA35mm1xcI4b8yAxo5Nld-EcgOa

[3] 百度百科. 模拟退火算法[J/OL]
.http://baike.baidu.com/link?url=8ZJG8noQ0cLzvBIWzhP31rigEyRrZuDnL8D3FBpztkcSVt4Udo1Ckluc3nI0HNjtiiEAYwgzwgCN1MV9q2LmAK

附录：

```
data=xlsread('20150915dataform.csv');
n=7;%取样个数
cost=0;%当前成本
cost_previous=0;%上一次成本
cost_min=0;%最小成本
temp_start=100;%初始温度
temp=temp_start;%当前温度
temp_end=0.01;%最低温度
r=0.99;%退火速度
num=0;%循环次数

%随机取 7 个点作为样品
A=randperm(49);
B=sort(A(1:n-2))+ones(1,n-2);
C=[1,B,51];
NEW=C;

while temp>temp_end
    num=num+1;
    my_answer=NEW;
    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
        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);

    if num==1
        cost_previous=cost;
        cost_min=cost;
        C=NEW;
        C_min=NEW;
    else if cost<cost_min%接受成本更小的取样方法
            cost_pevious=cost;
            cost_min=cost;
            C=NEW;
            C_min=NEW;
        else if rand<exp((cost_previous-cost)/temp_start)%一定概率接受成本更大的取样方法
                cost_pevious=cost;
                C=NEW;
            end
        end
    end

    %改变一个取样点
    D=setdiff(C,[1,51])-ones(1,n-2);
    E=randperm(n-2);
```

```
    F=zeros(1,n-2);
    F(E(1:n-3))=1;
    G=D(1,logical(F))+ones(1,n-3);
    H=setdiff([1:51],C);
    I=randperm(51-n);
    J=zeros(1,51-n);
    J(I(1))=1;
    K=H(1,logical(J));
    L=[1,G,K,51];
    NEW=sort(L);

    %退火
    temp=temp*0.99;
end

fprintf('\n 经计算，你的答案对应的最小总体成本为%5.2f\n',cost_min);
fprintf('\n 对应的取样点为%5.2f');
disp(C_min);
```