统计推断应用

摘要：本文讨论了一种插值方法应用的实际案例，用 matlab 辅助工具，采用遗传算法来完成对一组产品的定标工序，实现统计推断在工程技术中的应用。

关键词：matlab,插值，定标，遗传算法

一、引言

假定有某型投入批量试生产的电子产品，其内部有一个模块，功能是监测某项与外部环境有关的物理量（可能是温度、压力、光强等）。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。

二、本课题的限制条件

数学模型：最小二乘法（线性拟合）

名词：实测值 Y，输入数字信号 X,输出值（预测值） $\hat{Y}$,估测函数 $\hat{Y}$ =f(x),{(Xi，Yi)}为给定的样本数据，Si 为第 i 个样本的定标成本，C 为校准方案总成本

样本特性：

① Y 取值随 X 取值的增大而单调递增；

② X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；

③ 不同样本的特性曲线形态相似但两两相异；

④ 特性曲线按斜率变化大致可以区分为首段、中段、尾段三部分，中段的平均斜率小于首段和尾段；

⑤ 首段、中段、尾段单独都不是完全线性的，且不同个体的弯曲形态有随机性差异；

⑥ 不同个体的中段起点位置、终点位置有随机性差异。

成本计算公式：

- 单点定标误差成本

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

单点定标误差的成本按式（1）计算，其中 $y_{i,j}$ 表示第 i 个样本之第 j 点 Y 的实测值，$\hat{y}_{i,j}$ 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 $s_{i,j}$ 记。

- 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。

- 某一样本个体的定标成本

$$S_i = \sum_{j=1}^{51} s_{i,j} + q \cdot n_i \qquad (2)$$

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

- 校准方案总成本

按式（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。

$$C = \frac{1}{M} \sum_{i=1}^{M} S_i \qquad (3)$$

总成本较低的校准方案，认定为较优方案。

三、解决问题的基本思想

本问题为定标工序问题，主要分为两个部分：选点方法和定标方法。我采用的选点方法为遗传算法；而定标方法，为本课题讨论的问题。使用的辅助工具为 matlab。

遗传算法（Genetic Algorithm）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。（注②）

从理论上来讲，只要选择了恰当的取点、定标方法（最终使得成本最小），那么，通过遗传算法就能够取得一个较优的取点组合（最终使得成本最小），从而确定定标方案。因此，问题的关键在于选取怎样的定标方法，而选取点的优化方法的实现都是一样的，下面先给出遗传算法产生优化组合的实现，然后讨论取点和定标方法。

四、遗传算法产生优化组合

（一）采用[0,1]数组表示一个样本中的各个数据点，其中 1 表示该点被选取参与拟合参数的计算，0 表示未被选取，该数组表示这个样本参与拟合点的选取方式。如一个样本有 51 对（x_i, y_i），则随机产生一个数组 a[1,51]表示这组数据。

（二）步骤（一）中产生的数组为一种取点方式，在遗传算法中，以数个这样的随机数组表示其初始种群。如 father[:,N]=[$a_1 \ldots a_n$]表示有 N 个个体的初始种群，其中 a_i 是随机产生的个体，代表一种取点方式。

（三）在循环代数内，对族群进行交叉、变异、选择复制操作，从而产生较优的选点方式。

① 交叉：

保留第一、二个体的情况下，产生一个 N-2 维的 3~N 随机数组，然后据此数组将 当代族群分为随机的两个部分，便于后续交叉。在概率 crossrate 下，随机决定数组 A、B 两个的各个个体是否交叉。最后重新组合 A、B 数组为交叉的结果。

② 变异：

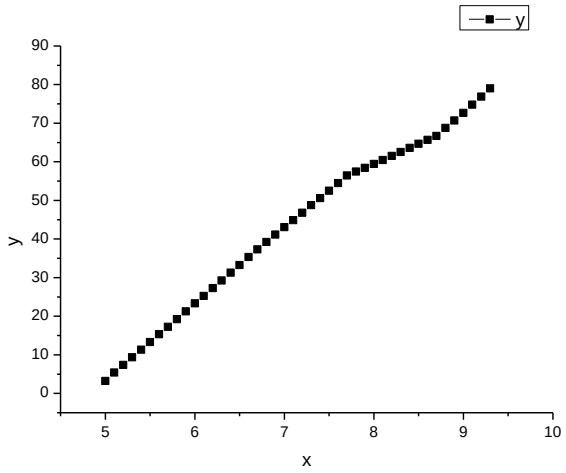
对于初始族群中的每个个体，在 muterate 的变异概率下，随机决定每个个体各个基因是否发生变异

③ 选择复制：

用 scorefun 计算该族群的所有个体的成本，用一个精英数组保存当前最佳个体（最优选点）然后用赌盘轮转法对本代进行选择复制，产生后代。（scorefun 函数里面用到了之后的定标函数，这个函数即为我们讨论的问题的关键：定标方法）

（四）将最终产生的精英数组转化为取点方式。

五、定标



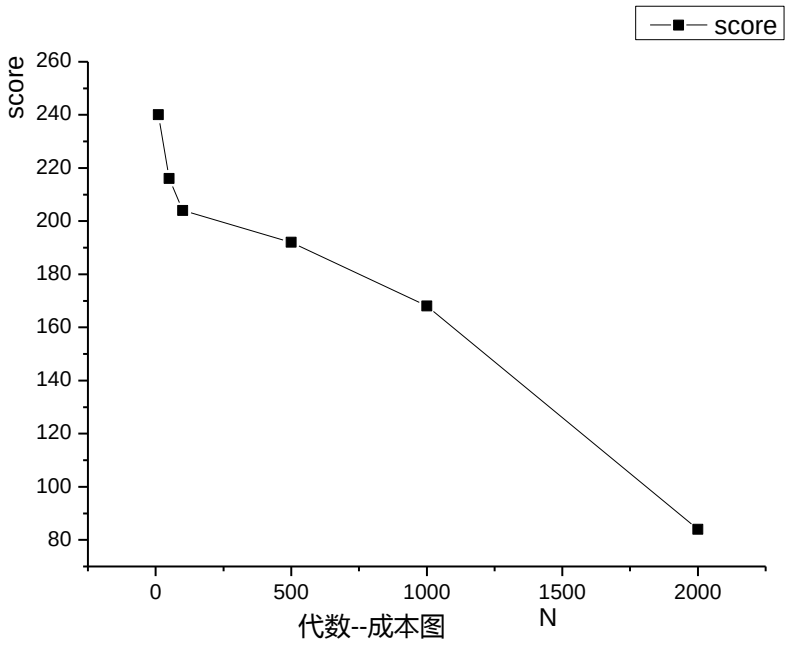
某个样本数据图

① **设计思想**：从直观上看数据的分布特点，我发现其呈现明显的“三段”分布，但是各转折点的位置，以及各段的长度有很大差异。于是我决定将数据分为三段，然后各段用最小二乘法计算（见代码 mycurvefitting2）。

② **实现**：对于遗传算法某代产生的点的随机组合，我将这些点均匀的分成 3 部分，然后对这 3 部分进行拟合、成本计算后，反馈回遗传算法中，产生更优的下一代。其中，线性拟合部分，我采用 matlab 自带的 interp1 函数。

③ **分析讨论**：以下是遗传算法经过 N 代后的结果：

代数N	10	50	100	500	1000	2000
取点	3 9 10 11 13 16 17 18 19 20 25 27 28 29 31 32 34 37 44 51	2 3 4 8 10 16 17 18 19 25 26 28 29 35 38 43 44 51	1 2 6 7 10 11 12 13 20 21 22 33 36 37 38 40 41	1 2 10 12 14 15 18 19 22 27 28 31 34 43 44 47	1 18 19 20 21 23 28 30 31 34 38 41 42	1 18 19 30 31 35 42
成本	240.00	216.00	204.00	192.00	168.00	84.00



从算法上分析：遗传算法的选择方式，会很好地保证转折点的随机性，即第一段与

第二段的转折点、第二段与第三段的转折点的选取都会越来越精确。但此算法有个非常致命的缺陷，就是选点的时候，在分划的三段拟合区间上，选取的点数都是相同的。但事实上，最优解和较优解中，各区间的取点个数比应该是由遗传算法计算给出的。由此，对于一个随机的数据样本，会出现以下两种状况：

a. 在三个区间中，若数据具有很好的线性相关性，那么在每个区间中，由于我采用的是 matlab 自带的 interp1 的线性拟合方式，那么在每段的取点个数将会趋近于 2，这种算法产生的结果不会有较大出入。

b. 在三个区间中，某区间的数据分布线性相关性较差，而其他区间数据具有高度线性相关性，那么此时进行线性拟合时，在高度线性相关区间取点数仍应趋近于 2，但另一区间的取点个数则不能确定。采用此算法就不一定能得到很好的结果。

但是本课题中，数据比较特殊，三个数据区间的数据线性相关度虽然不是 1，但是都在 1 附近，因此本课题的较优解在各区间取点数都是差不多的（理想的结果是总取点数为 6）。因此本课题采用此算法是可行的。

从某次实验结果分析：分析结果见上表和图。随着遗传算法代数的增加，最终得到的优化取点组合为【1 18 19 30 31 35 42】，而成本为 84.00,这一结果在代数达 2000 以上时趋于平稳（图表中为给出），因此可以认为，本结果即为采用此种算法得出的一组优化解。但从图表中仍可以看出一个问题：算法产生优化解的收敛速度太慢，最终进化了 2000 代以上，非常耗时（一次实验总计耗时 5h 以上）。一下是遗传算法采用的一些参数：

种群大小	变异概率	交叉概率（片段交叉）	后代选择方式
51	0.08	0.5	赌盘轮转法

从时间复杂度上考虑，改变遗传算法的参数应该会提高收敛速度，但算法的时间耗费与本课题结果的得出没有太大关系，所以就没有进一步优化。

六、最终结果

选取用于定标的点：【1 18 19 30 31 35 42】

定标方法：将选取的点均匀分为 3 部分（向下取整），然后在分取的 3 个区间内求线性拟合函数，即为定标函数。

定标成本：84.00.

七、反思评价

在实验之初，我错误地将本课题的讨论重点放在“用遗传算法选取优化点”上，导致后期工作开展时遇到很多麻烦。比如，本课题除了用线性拟合的定标方法外，应该还存在其余可能更优的方法，本课题未予讨论；辅助用的遗传算法时间复杂度太高，最后没有优化等等。虽然得到了一个问题的优化解，但并没有进行拓展讨论。

八、参考文献

1. 百度百科“遗传算法”，matlab 入门
2. 上海交通大学统计推断课程
3. 《DS 证据理论在雷达体制识别中的应用》王勇 毕大平

【附录】

```
function myanswer=main()%遗传算法主程序，产生取点位置
minput=dlmread('20150915dataform.csv');%读入文件
[~,NN]=size(minput);
num=NN;   %每个个体基因数
N=50;   %种群个体数
maxgen=2000;     %更替代数
crossrate=0.5;
muterate=0.08;
generation=1;    %当前代数
fatherrand=randint(num,N,2);     %初始族群
score=zeros(maxgen,N);      %成本矩阵
elite=zeros(maxgen,num);   %每一代的精英矩阵
perelite=zeros(num+1);  %当前最优解
while generation<=maxgen
    ind=randperm(N-2)+2;   %产生随机交叉配对的方法矩阵
    A=fatherrand(:,ind(1:(N-2)/2)); %A 为 24 个随机个体
    B=fatherrand(:,ind((N-2)/2+1:end)); %B 为另外 24 个随机个体

    %多点交叉
    rnd=rand(num,(N-2)/2);
    for pp=1:num
        for qq=1:(N-2)/2
            if rnd(pp,qq)>crossrate
                tmp=A(pp,qq);
                A(pp,qq)=B(pp,qq);
                B(pp,qq)=tmp;
            end
        end
    end
    fatherrand=[fatherrand(:,1:2),A,B];


    %变异
    rnd=rand(num,N);
    tmp=zeros(num,N);
    for tt=1:num
        for ww=3:N
            if rnd(tt,ww)>muterate
                tmp(tt,ww)=1;
            end
        end
    end
```

```
    fatherrand=tmp+fatherrand;
    fatherrand=mod(fatherrand,2);

    %后代选择以及评价函数
    scoreN=scorefun(fatherrand);    %计算每个个体的成本函数
    score(generation,:)=scoreN;
    [scoreSort,scoreind]=sort(scoreN); %排序
    sumscore=cumsum(scoreSort);
    sumscore=1.-sumscore./sumscore(end);
    childind=zeros(1,N);
    childind(1:2)=scoreind(1:2);    %子代的第一、二个个体确定
    elite(generation,:)=fatherrand(:,scoreind(1))'; %本代的精英
    if generation==1
        perelite=[elite(generation,:),scoreSort(1)];%当前最优解
    else
        if scoreSort(1)<perelite(end)
            perelite=[elite(generation,:),scoreSort(1)];%当前最优解
        end
    end


     %赌盘轮转法选取后代
    for k=3:N
        tmprand=rand;
        jj=3;
        t=sumscore(jj);
        while t<tmprand
            jj=jj+1;
            if jj>N
                break
            end
            t=sumscore(jj);
        end
        jj=jj-1;
        childind(k)=scoreind(jj);
    end
    fatherrand=fatherrand(:,childind);
    generation=generation+1;
end
pq=0;
for n=1:num
    if perelite(n)==1
        pq=pq+1;
    end
```

```
end
myanswer=zeros(1,pq);
pq=0;
for n=1:num
    if perelite(n)==1
        pq=pq+1;
        myanswer(1,pq)=n;
    end
end

function scoreN=scorefun(father)%main()函数中 scorefun 的实现
[num,N]=size(father);
scoreN=zeros(1,N);
for nn=1:N
    tmp=father(:,nn)';
    pq=0;
    for n=1:num
        if tmp(n)==1
            pq=pq+1;
        end
    end
    my_answer=zeros(1,pq);
    pq=0;
    for n=1:num
        if tmp(n)==1
            pq=pq+1;
            my_answer(1,pq)=n;
        end
    end
    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
    for j=1:nsample
        y1(j,:)=mycurvefitting2(x(j,:),y0(j,:),my_answer(1,:));
    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;
    scoreN(1,nn)=sum(si)/nsample;
end
function cacul=mycurvefitting2(xx,yy,tt)%定标计算函数
[~,h2]=size(xx);
cacul=zeros(1,h2);
[~,r2]=size(tt);
j1=fix(r2/3);
cacul(1,1:tt(j1))=interp1(xx(1,tt(1):tt(j1)),yy(1,tt(1):tt(j1)),xx(1,
1:tt(j1)));
j2=2*j1;
cacul(1,tt(j1+1):tt(j2))=interp1(xx(1,tt(j1):tt(j2)),yy(1,tt(j1):tt(j
2)),xx(1,tt(j1+1):tt(j2)));
cacul(1,tt(j2+1):h2)=interp1(xx(1,tt(j2):tt(r2)),yy(1,tt(j2):tt(r2)),
xx(1,tt(j2+1):h2));



%%%%%%% 答案检验程序 %%%%%%%%
```

```
my_answer=[ 1 18  19 30 31 35  42];
%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,:)=mycurvefitting2(x(j,:),y0(j,:),my_answer(1,:));
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);
```