统计推断在数模转换系统中的应用

第 55 组　　闻天明 5130309660，吴宇昊 5130309281

摘要：运用统计推断的方法对实际问题进行估计以得到样本值从而减小误差

关键词：统计推断，模拟，分段，样本

1 引言

在数理统计学中我们总是从索要研究的对象全体中抽取一部分进行观察或试验以取得信息，从而对整体作出判断。显然这种判断含有一定程度的不确定性，而不确定性用概率的大小来表示，这种伴随有一定概率的推断统称为统计推断。

在这门课程中，我们将要对一个在十几种可能出现的情况进行分析，用统计推断的只是进行模拟，以便得到最优解决方案。

2 问题的提出和模型的建立

2.1 实际物理问题的提出和描述

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。

2.2 针对实际情况的模型建立

对于上述实际情况，我们有的测量值 469 组，其中每组有 51 测量数据，每个数据间隔 0.1. 考虑到在结算成本时，成本分为测量成本和误差成本两部分。所以问题的关键在于选取的节点位置和节点数量以及拟合时采用的公式选择。

模型的目的是根据检测的物理量 X 中若干个取值 x，确定一种方案，是的所得到的校准结果成本最低。在实际操作中，先考虑从给定的大量样本中归纳出其大致的表达式关系，在通过进一步测定少量样本其具体的关系表达式，从而通过这几个少量样本得到全部的 51 个点的数据。

3 基于模型的求最优方案过程

基于之前的模型建立过程，将整个问题的求解分为以下几个步骤:

3.1 通过直线拟合找出多组数据中的中段始末点位置，找出中段始末点大致满足的概率分布

为了判断大致的图形形状，首先以五个为一组，随机抽取了 5-10，200-205，310-315，460-465 这四组数据，将它们输出到图像上。如图 3.1.1 所示。

不难看出，作为每段曲线而言，都大致上可以分为三段，而其中，初步分析首段和中段线性度较好，而末段有些近似于曲线。由于最后的成本计算是要考虑到测定点成本和误差成本。所以决定对于曲线首先进行分段。从选取的几组数据的图像中可以看出，虽然数据在 Y 轴上有较大的上下波动，但是在 X 轴上的分段大致一致。所以分段的方法可行。

在考虑分段方法的过程中，根据图线首先想到的是用拟合直线求交点的方法。由于在计算过程中，如果采取二次曲线会产生结果的二异性，所以计划采用一次曲线拟合的方法。

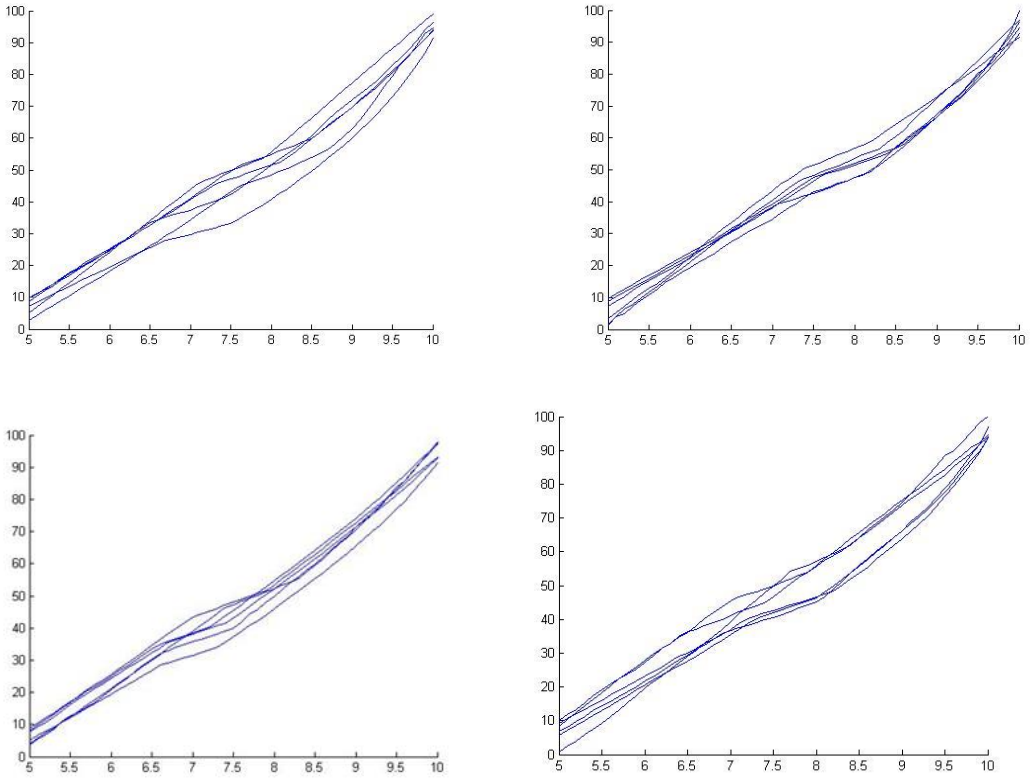

图 3.3.1 随机抽取的四组数据的图像

为了使用于求交点的拟合直线更加精确，应当选取更多的拟合点。但是由于中段起始点的位置范围并不明确，所以应当注意取点以免将终端首末点列入到拟合的选点中。所以选点应遵循尽可能多并避开首末段点。

在此基础上，观察图像的拐点接近于 6.5 和 7 之间但不超过 6.5（6.5 代表点 16）。因此决定首段选取点 1-16。

中段大致范围波动较大，从 6.5-8.5 不等。折中选取 7-8 的部分。即中段选点（21-31）

末段总体线性度一般，所以选取尽可能多的点。考虑到图像上 8.5（也就是点 36）后都为末段的范围。选取了点 36-51。

针对上述选取的点，进行曲线拟合可获得在 469 组数据中每个点作为首段和中段交点出现的次数和作为中段和末段交点出现的次数。如图 3.3.2(a)和 3.3.2（b）所示。

由图可以看出，中段的末段选取点没有问题，32，33，34 三个点出现的次数远多于其他点，而数据在 33 周围分布也比较均匀。所以选取 33 作为中段的末点。

可是在观察 3.3.2（a）的时候，发现测得的中段起点出现次数最多的为 22，处于选择的范围内。所以重新取点。根据末段的测量结果，中段重新取点为 25-33。得到图 3.3.2（c）。

观察图 3.3.2（c）。发现 1 和 51 多次出现。考虑 matlab 的机制，是由于所计算交点的横坐标小于 1 或者大于 51 而产生的结果。从结果可以看出，样本的中段起点出现非常不规律。在图 3.3.2（c）中抛开无效点之后的出现结果为 25 最多。但是由于结果存在不确定性。所以考虑对成本进行一次验证。验证方法是拟合时首段取 1-24 拟合，中段取 25-33 拟合，如果拟合出来的交点正好是 25 次数最多。说明 25 是符合大多样本的中段起点。测量结果如图 3.3.2（d）。

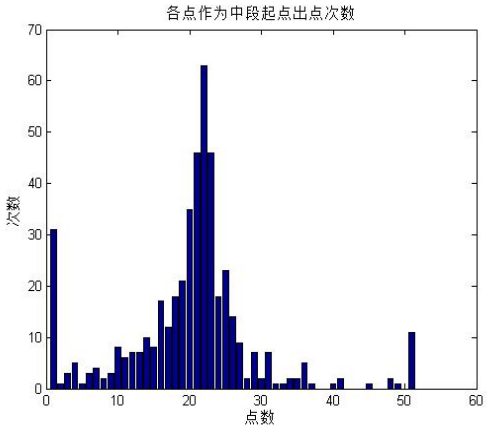

图 3.3.2（a）

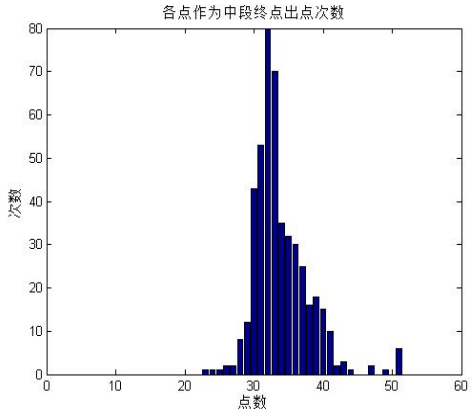

图 3.3.2（b）

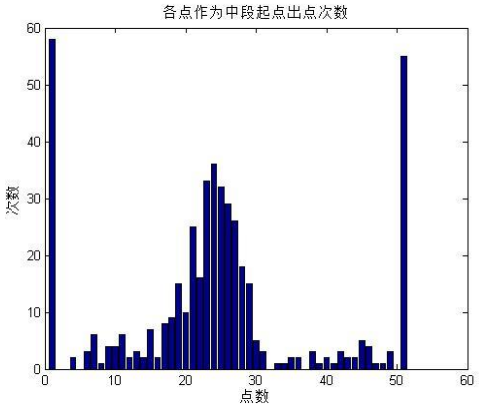

图 3.3.2（c）

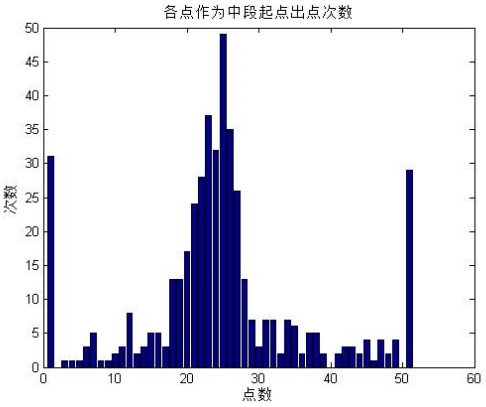

图 3.3.2（d）

观察图 3.3.2（d），发现果然无效点大大减少，而 25 出现的次数大大增多，说明所选取的 25 的确是符合大部分样本的中段起点。

至此，选取的过程结束。得到的最终中段起点为 25，中段中点为 33。在这一步之后，图像被分为了 1-24，25-33，34-51 这三个部分。

3.2 针对每段选择最合适的方案，并将方案进行整合。

3.2.1 第一段

针对第一段，首先考虑的第一种方案是只取一个点，然后和点 25 一齐进行一阶曲线拟合求误差。由于只可能有 24 种情况。所以一一测定。然后得到了图 3.2.1（a）的曲线。可见误差非常之大，并且随着两个点的距离越接近，误差就越大。最小的时候也在 31.05。所以可以增加取点。即在 1-24 范围内去两个点。由于 24C2=276，数字并不非常大。所以使用穷举法计算。首先测试拟合曲线为直线时候的结果。运行过程中，整个程序的运行时间大概在 3 分钟左右。

直线时的测试结果为：

直线拟合首段最小误差为 15.75

此时首段第一点为 4.00

此时首段第二点为 18.00

首段第三点（中段起点）为 25.00

虽然从整体上而言，误差有所减小，但是还是处在一个较大的范围。所以有两种方案：一是增加取点，继续用直线拟合，二是减小取点，使用二次曲线拟合。由于第一种方法要将整体误差控制在 3.75 之内，实现起来可能较为困难。所以

采用第二种方法，依然用穷举的方法，只是改为二次曲线拟合。

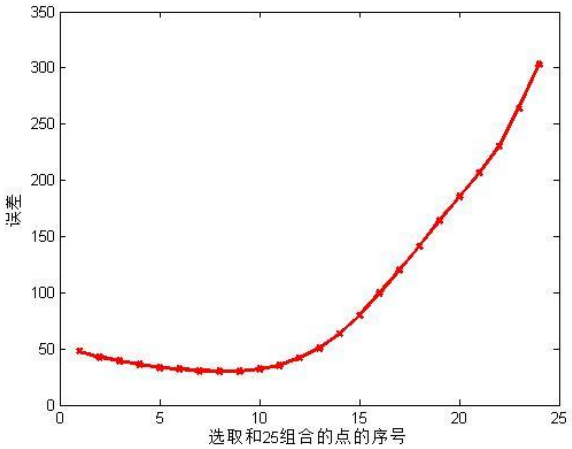

图 3.2.1（a）两点一阶拟合时首段误差

以下为二次曲线拟合结果：

直线拟合首段最小误差为 5.73

此时首段第一点为 4.00

此时首段第二点为 15.00

首段第三点（中段起点）为 25.00

结果误差非常之小，令人满意。所以最终第一段的方案为选点 4，15，25，进行二次曲线拟合。

3.2.2 第二段

通过图像观察，首先计算在之取中段首末点进行直线拟合下的中段误差。根据成本函数测试，得到中段的平均误差为 3.83。此误差非常小。并且远小于单点的成本测定值 12。所以说明针对 25-33 这部分点，方案比较正确。

所以中段的最终方案为选点 25，33，也即中段始末点，进行-直线拟合。

3.2.3 第三段

考虑到第三段在 3.1 中随机选取组的图像上接近于曲线，所以直接考虑用二次曲线拟合来进行分析。即在 34-51 的范围内取 2 个点，和 33 点一齐进行曲线拟合。考虑 18C2=153，所以和 3.2.1 一样，也运用穷举法。

以下为测试结果：

首段最小误差为 3.18

末段第一点为 33.00（中段终点）

此时末段第二点为 43.00

此时末段第三点为 50.00

误差远小于 12，说明努力方向正确。末端的选取方案为选点 33，43，50，进行二次曲线拟合。

4 方案的最终确定

根据上一部分的模拟测试结果，取得最终的方案为：选取共六个点，分别为【4，15，25，33，43，50】将曲线分成三个部分，具体描述如下：

将样本分成三段：

第 1-24 个点为第一段，取点 4,15,25 进行二阶线性拟合

第 25-33 个点为第二段，取点 25,33 进行一阶线性拟合

第 34-51 个点为第三段，取点 33,43，50 进行二阶线性拟合

样本测试结果为：

经计算，你的答案对应的总体成本为 84.76

该成本除去 12*6 的测定成本外，误差成本仅为 12.76，可以接受。

从而得出平均样本估计函数为：

$$F(x)=\begin{cases}-0.843923x\text{^}2+\ 26.545757x+\ -106.892593\\ 12.998241x+\ -52.854211\\ 3.106194x\text{^}2\ +-33.879846x+\ 122.685608\end{cases}$$

5 对出现问题的再讨论分析

5.1 对穷举法的说明

值得注意的是，在整个问题的求解过程中，并未使用遗传或者退火算法，而是从图像入手，用更加具象的方法对问题进行切入。整个求解问题的精髓所在，其实是 3.3.1 和 3.3.3 中所使用的穷举法。

不同于退火算法和遗传算法的以一定随机概率接近最优解，穷举法是测试所有可能的情况，其结果更加准确。而在统计推断的实例中，由于穷举法往往意味着大量的运算时间和很低的运算效率，所以并不能广泛采用。但是针对此次的模型，在 3.1 将样本分段的基础上，成功地将使用穷举法的时间成本大大降低。整个 3.2 测试程序的总的运算时间在 10 分钟之内，其实在时间上还可以进一步缩短。

以 1-24 点为例。由于目的是在已有点 25 的基础上，再取两点进行二次曲线拟合，而考虑到二次拟合曲线的特性与图 3.2.1（a）的结果，所取的点分布应在样本内均匀。所以其实只要将 1-24 的分为三段，即 1-12 和 13-24 两段，共有 12*12=144 种可能，相较于完全穷举，时间缩短了一半，34-51 相同。而最终的取点结果证实了这一想法。

5.2 对样本分段的再讨论

如果在方案的制定中还存在问题，那么应该是分段的方法上。由 3.3 的拟合可知，事实上第一段和第三段样本在取点较少的情况下，更接近于二次曲线，这从 3.1 的图中也能看出。当然，在分段时采取了大量的拟合点，所以保证将这种误差降到最低。从求直线交点的结果来看，中段直线末端点的求解情况比较令人满意，而其起点的情况并不尽如人意。

其实从随机选取的机组样本数据的图像中也不难看出，其实从中段起点的情况来看，其分布有很大的波动性，无怪求解的波动很大。虽然从最后的成本来看，结果还是比较良好。但是是否可以换种方法来求中段的起点呢？由于时间问题，并没能针对这个问题进行进一步测量，但是有一些设想，或许在之后可以有机会再次测量。一种或许可行的方法是在 3.2.1 和 3.3.2 中，在 25 周围更改中点的位置，然后反复进行测量，求得最近接真实情况的中点位置。这种方法的时间复杂度非常之高，经过测试未能实现。有待于进一步的分析。

6 总结与感悟

在整个样本估计的过程中，采取的主要是最基本的数学思想。切入点是曲线图像。此方法具有其局限性，不适用于所有的样本估值方法。但是由于本次样本的形式特殊，所以该方法得以实现，并且具有很小的成本和较快的计算速度。

统计推断本质上是一门很高深的学问。在现代生活中有广泛的运用。尤其在

提倡云端大数据的当下，统计推断更是具有其得天独厚的优势。本次的报告内容其实只是冰山一角，希望日后能有机会再次接触。

最后感谢在中期面谈中老师给与的宝贵意见和帮助。

7 附录（代码清单）

7.1 随机选取几组数据画出图像以便观察

```
function draw(data)
hold on
    for i=310:315
        plot(data(i*2-1,:),data(i*2,:))
    end
hold off
end
```

7.2 对选取的数据进行曲线拟合以求其始末点

```
function y=steptwo(data)
n=length(data);
m1=zeros(1,51);
m2=zeros(1,51);
for k=2:2:n;
e1=polyfit(data(k-1,1:24),data(k,1:24),1);
e2=polyfit(data(k-1,25:33),data(k,25:33),1);
e3=polyfit(data(k-1,36:51),data(k,36:51),1);
a1=[(e1(1)-e2(1)),(e1(2)-e2(2))];
p1=roots(a1);
p1=(round(10*p1)/10-5)*10;
a2=[(e2(1)-e3(1)),(e2(2)-e3(2))];
p2=roots(a2);
p2=(round(10*p2)/10-5)*10;
m1(1,k/2)=p1;
m2(1,k/2)=p2;

end
m6=1:1:51;
[y1,~]=hist(m1,m6);
bar(m6,y1);title('各点作为中段末尾点的出现次数');xlabel('点数');ylabel('次
数');
m2=m2(:);
[y2,~]=hist(m2,m6);
bar(m6,y2);title('各点作为中段起始点的出现次数');xlabel('点数');ylabel('次
数');
end
```

7.3 用曲线拟合的方法求方案（3.3.1 和 3.3.2 和 3.3.3 类似，为节省篇幅只上传 3.3.1）

```
function stepfour(data)
minput=data;
[M,N]=size(minput);
nsample=M/2; npoint=N;
y0=zeros(nsample,npoint);
y1=zeros(nsample,npoint);
for i=1:nsample
    y0(i,:)=minput(2*i,:);
end
y1(:,25:npoint)=y0(:,25:npoint);
mincost=100;
p1=20;
q1=20;
for p=1:24
```

```
    for q=p+1:24
for i=1:nsample
    k1=2*i-1;
    k2=2*i;
    m1=[minput(k1,p),minput(k1,q),minput(k1,25)];
    n1=[minput(k2,p),minput(k2,q),minput(k2,25)];
    c1=polyfit(m1,n1,2);
for j=1:25
    y1(i,j)=polyval(c1,j*0.1+4.9);
end
end
errabs=abs(y0-y1);
le0_5=(errabs<=0.5);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);
sij=0.5*(le1_0-le0_5)+1.5*(le2_0-le1_0)+6*(le3_0-le2_0)+12*(le5_0-le3
_0)+25*g5_0;
si=sum(sij,2);
cost=sum(si)/nsample;

if mincost>cost
    mincost=cost;
    p1=p;
    q1=q;
end
    end
end
fprintf('\nÊ×¶Î×îÐ¡Îó²îÎª%5.2f\n',mincost);
fprintf('\n´ËÊ±Ê×¶ÎµÚ»ãÎª%5.2f\n',p1);
fprintf('\n´ËÊ±Ê×¶ÎµÚ¶þµãÎª%5.2f\n',q1);

end
```

7.4 成本计算

```
function stepthree(data)
my_answer=[4,14,26,33,43,50];
my_answer_n=size(my_answer,2);
minput=data;
[M,N]=size(minput);
nsample=M/2; npoint=N;
y0=zeros(nsample,npoint);
```

```
y1=zeros(nsample,npoint);
d1=0;
d2=0;
d3=0;
for i=1:nsample
    y0(i,:)=minput(2*i,:);
end
for i=1:nsample
    k1=2*i-1;
    k2=2*i;
    m1=[minput(k1,4),minput(k1,14),minput(k1,25)];
    n1=[minput(k2,4),minput(k2,14),minput(k2,25)];
    m2=[minput(k1,25),minput(k1,33)];
    n2=[minput(k2,25),minput(k2,33)];
    m3=[minput(k1,33),minput(k1,43),minput(k1,50)];
    n3=[minput(k2,33),minput(k2,43),minput(k2,50)];
    c1=polyfit(m1,n1,2);
    c2=polyfit(m2,n2,1);
    c3=polyfit(m3,n3,2);
    d1=d1+c1;
    d2=d2+c2;
    d3=d3+c3;
for j=1:24
    y1(i,j)=polyval(c1,j*0.1+4.9);
end
for j=25:33
    y1(i,j)=polyval(c2,j*0.1+4.9);
end
for j=34:npoint
    y1(i,j)=polyval(c3,j*0.1+4.9);
end
end
Q=12;
errabs=abs(y0-y1);
le0_5=(errabs<=0.5);
le1_0=(errabs<=1);
le2_0=(errabs<=2);
le3_0=(errabs<=3);
le5_0=(errabs<=5);
g5_0=(errabs>5);
sij=0.5*(le1_0-le0_5)+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;
```

```
d1=d1/469;
d2=d2/469;
d3=d3/469;
fprintf('\ny=%fx^2+ %fx+ %f\n', d1)
fprintf('\ny=%fx+ %f\n', d2)
fprintf('\ny=%fx^2 +%fx+ %f\n', d3)
fprintf('\n您的样本成本为%5.2f\n',cost);
end
```