统计推断在模数、数模转换系统中的应用

第 56 组 沈剑清 5130309661 高源 5130309670

摘要：为了完成一个校准问题，通过思考建立模型并寻找算法，最后得出一个可行的方法。

关键词：监测模块，建立模型最优方案，拟合，遗传算法，89.24

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 问题的提出

为了对本课题展开有效讨论，需建立一个数学模型，对问题的某些方面进行必要的描述和限定。

监测模块的组成框图如图 1。其中，传感器部件（包含传感器元件及必要的放大电路、调理电路等）的特性是我们关注的重点。传感器部件监测的对象物理量以符号 Y 表示；传感部件的输出电压信号用符号 X 表示，该电压经模数转换器（ADC）成为数字编码，并能被微处理器程序所读取和处理，获得信号 $\hat{Y}$ 作为 Y 的读数（监测模块对 Y 的估测值）。

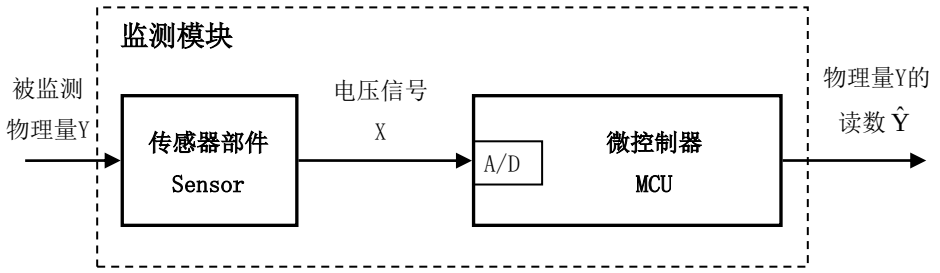


图 1 监测模块组成框图

所谓传感特性校准，就是针对某一特定传感部件个体，通过有限次测定，估计其 Y 值与 X 值间一一对应的特性关系的过程。数学上可认为是确定适用于该个体的估测函数 $\hat{y}=f(x)$的过程，其中 x 是 X 的取值，$\hat{y}$ 是对应 Y 的估测值。

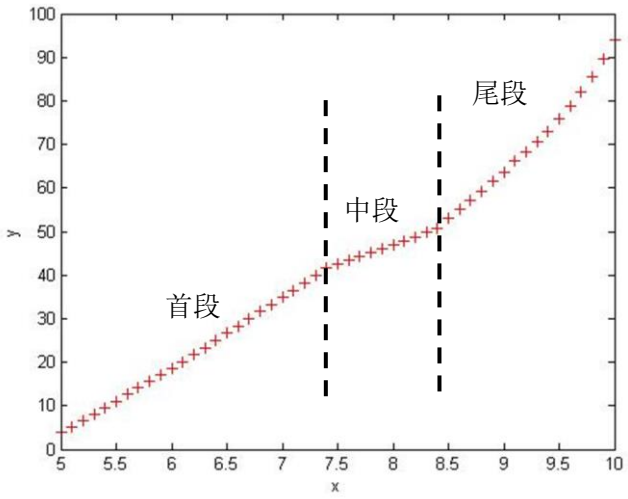
考虑实际工程中该监测模块的应用需求，同时为便于在本课题中开展讨论，我们将问题限于 X 为离散取值的情况，规定

$$X\in\{x_1,x_2,x_3,...,x_{50},x_{51}\}=\{5.0,5.1,5.2,...,9.9,10.0\}$$

相应的 Y 估测值记为 $\hat{y}_i=f(x_i)$， Y 实测值记为 y_i， $i=1,2,3,...,50,51$。

3.实现过程中一些可能的需要

3.1 选取合适的拟合或插值方法



首

图 2 传感特性图示

一个传感部件个体的输入输出特性大致如图 2 所示，有以下主要特征：

- Y 取值随 X 取值的增大而单调递增；
- X 取值在[5.0,10.0]区间内，Y 取值在[0,100]区间内；
- 不同个体的特性曲线形态相似但两两相异；
- 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；
- 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；
- 不同个体的中段起点位置、终点位置有随机性差异。

为进一步说明情况，图 3 对比展示了四个不同样品个体的特性曲线图示。

观察到图像的特点，决定用分段多项式曲线来拟合。

首先将区间划分如下：

表 1 曲线拟合区间的划分

区间	电压范围	对应的采样点序号	多项式曲线类型
1	[5.0V，V_1]	1,2,…,N_1	n_1次多项式
2	[V_1，V_2]	N_1,…,N_2	一次多项式
3	[V_2，10.0V]	N_3,46,…,53	n_2次多项式

易见，表中的电压范围、采样点序号和多项式曲线类型还没确定。

3.2 特征点的选取

特征点的选取又有两个因素——数量和组合，根据老师的提示，这类 NP-hard 问题将使用“启发式搜索”遗传算法（GA）或模拟退火算法（SA）来解决。这样以便在能够较为准确找到数据分布规律的同时，降低所需要的成本。

为评估和比较不同的校准方案，特制定以下成本计算规则。

- 单点定标误差成本

$$s_{i,j}=\begin{cases}0 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|\le 0.5\\ 0.5 & \text{if } 0.5<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 1\\ 1.5 & \text{if } 1<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 2\\ 6 & \text{if } 2<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 3\\ 12 & \text{if } 3<\left|\hat{y}_{i,j}-y_{i,j}\right|\le 5\\ 25 & \text{if } \left|\hat{y}_{i,j}-y_{i,j}\right|>5\end{cases} \quad (1)$$

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

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

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。

- 校准方案总体成本

按式（3）计算评估校准方案的总体成本，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。

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

总体成本较低的校准方案，认定为较优方案。

4. 实践过程

4.1 暴力穷举法

若用暴力穷举法，解决以上问题（51 中取 7 的组合数）需要尝试约 1.15 亿种不同组合，显然这是不切实际的，故未尝试。

4.2 第一次尝试

输入的数据量十分大，我们便尝试缩小以减少运算过程。我们想到了拟合平均行向量的方法，即对每列求平均值从而得到一个行向量，然后对平均行进行了 4 次拟合，然后再算出每一点的方差，按方差从小到大的顺序排列，取方差最小的 7 个点作为结果进行运算，但运算后发现与预期差距很大。可见这种方法离最优方案差距极大，故不得不放弃。（代码见附录）

4.3 方法改变

经过第一次的失败和考虑了老师的教导，决定在程序中运用遗传算法以优化得到较理想的结果。

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。它是由美国的 J.Holland 教授 1975 年首先提出，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。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。

对于一个求函数最大值的优化问题(求函数最小值也类同)，一般可以描述为下列数学规划模型：式中 x 为决策变量，式 2-1 为目标函数式，式 2-2、2-3 为约束条件，U 是基本空间，R 是 U 的子集。满足约束条件的解 X 称为可行解，集合 R 表示所有满足约束条件的解所组成的集合，称为可行解集合。

遗传算法也是计算机科学人工智能领域中用于解决最优化的一种搜索启发式算法，是进化算法的一种。这种启发式通常用来生成有用的解决方案来优化和搜索问题。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优[1] ，而不能达到全局最优。

遗传算法的基本运算过程如下：

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。

遗传算法 $\begin{cases} \max f(X) & 2\text{-}1 \\ x \in R & 2\text{-}2 \\ R \subset U & 2\text{-}3 \end{cases}$

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：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

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)。

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。

我们完成了遗传算法，但在尝试将其与其他程序融合完成整体代码时遭到了巨大的阻碍，bug 很多。由于 MATLAB 是从头学起，我们前进十分困难，最后由于时间问题不得不放弃。但在尾部我还是附上了我们完成的遗传部分。

4.4 最终结果

在无法正确将遗传程序与其他程序融合成一体的情况下，我们直接使用了 METLAB 自带内建遗传函数，使用这个程序不仅减少了运算 bug 出现的几率，而且相当程度上节省了代码空间。通过网上资料了解到 MATLAB 自带遗传函数默认数据：交叉概率为 0.6，变异概率为 0.05.

使用遗传算法的运行结果如下：

七个数据点：1 13 21 31 41 51

最佳成本：95.13

5 结论

经过多次实践，我们最终运用 MATLAB 的内建函数得出了 95.13 这一结果。这堂课虽然不是每周都上，却也付出了我们很多的学习，尤其在学习 MATLAB 并编写比较优化和正确的运行代码。有付出方有收获，珍惜我们的成果，也感谢老师和助教的帮助。

6 参考文献

《机械工程师》2004 年 第 11 期 27-28 页 **主　　编：**杨桂霞

- 高尚.基于 MATLAB 遗传算法优化工具箱的优化计算.微型电脑应用.2002.52-54
- 姜阳[1];孔峰[2].基于 MATLAB 遗传算法工具箱的控制系统设计仿真.广西工学院学报.2001.6-9
- 周明,孙树栋. 遗传算法原理及应用[M].北京:国防工业出版社,1999.
- 张志涌,徐彦琴. Matlab 教程[M].北京:北京航空航天大学出版社,2001.
- Christopher R Houck,Jeffery A Joines,Michael G Kay. Agenetical Gorithm for Function Optimization:A Vatlab Implementation[Z].

附录：

一、取平均行向量拟合

```
format long
```

```
data=xlsread('dataform.csv');
b = mean (data)';c = var(data)';d = [5:0.1:10]';
b = [b,c,d];

x = b (:,1)';y = b(:,3)'
ft = fittype('poly4');
yourLine = fit(x',y',ft);

hold on
plot(yourLine)
coeffvalues(yourLine)
d = [5:0.1:10];
c = 0;
 for  n = 1:141
c  = c +sum (( yourLine (data(n,:))'-d).^2);
end
c
%拟合得到曲线 y = 0.000000058258685x^4 + (-0.000017160288059)x^3 +
0.001407302647484x^2 + 0.024553959799427x^1 + 4.917853349522235

%用 51 个点拟合四次曲线可得到 c1 值为 2.810157113742719e+02
A=(yourLine (b (:,1))'-d).^2;
[B,id]=sort(A)

%得到  39 14 27 49 48 4 38 组为方差最小。
%找到对应的电压值和均值，
% 8.8 6.3 7.6 9.8 9.7 5.3 8.7
% 65.3126 26.9295 45.6698 88.5236 85.9337 10.0845 63.2355

x = [65.3126 26.9295 45.6698 88.5236 85.9337 10.0845 63.2355];
y = [8.8 6.3 7.6 9.8 9.7 5.3 8.7];

ft = fittype('poly4');
yourLine = fit(x',y',ft);
%对比下拟合曲线和原来的点
plot(x,y,'*');
hold on
plot(yourLine)

coeffvalues(yourLine)
d = [5:0.1:10];
c = 0;
 for  n = 1:141
c  = c +sum (( yourLine (data(n,:))'-d).^2);
```

```
 end
c
%拟合得到曲线 y = 0.000000076195043x^4 + (-0.000020608171535)x^3 +
0.001625265069711x^2 + 0.019389647463606x^1 + 4.959894264808884
%用这 7 个点拟合四次曲线可得到 c2 值 2.809940480326293e+02

%总代价 c=c1+c2=5.6200975593e+02
```

二、不取平均行向量，采用遗传算法（未能最终完成整个程序，但给出了遗传算法主体部分

```
%IAGA
function best=ga
clear
MAX_gen=200;              %最大迭代步数
best.max_f=0;             %当前最大的适应度
STOP_f=14.5;              %停止循环的适应度
RANGE=[0 255];            %初始取值范围[0 255]
SPEEDUP_INTER=5;         %进入加速迭代的间隔
advance_k=0;              %优化的次数

popus=init;               %初始化
for gen=1:MAX_gen
    fitness=fit(popus,RANGE);        %求适应度
    f=fitness.f;
    picked=choose(popus,fitness);    %选择
    popus=intercross(popus,picked); %杂交
    popus=aberrance(popus,picked); %变异
    if max(f)>best.max_f
        advance_k=advance_k+1;
        x_better(advance_k)=fitness.x;
        best.max_f=max(f);
        best.popus=popus;
        best.x=fitness.x;
    end
    if mod(advance_k,SPEEDUP_INTER)==0
        RANGE=minmax(x_better);

        RANGE

        advance=0;
    end
```

```
end
return;
function popus=init%初始化
M=50;%种群个体数目
N=30;%编码长度
popus=round(rand(M,N));
return;

function fitness=fit(popus,RANGE)%求适应度
[M,N]=size(popus);
fitness=zeros(M,1);%适应度
f=zeros(M,1);%函数值
A=RANGE(1);B=RANGE(2);%初始取值范围[0 255]

for m=1:M
    x=0;
    for n=1:N
        x=x+popus(m,n)*(2^(n-1));
    end
    x=x*((B-A)/(2^N))+A;
    for k=1:5
        f(m,1)=f(m,1)-(k*sin((k+1)*x+k));
    end
end
f_std=(f-min(f))./(max(f)-min(f));%函数值标准化
fitness.f=f;fitness.f_std=f_std;fitness.x=x;
return;

function picked=choose(popus,fitness)%选择
f=fitness.f;f_std=fitness.f_std;
[M,N]=size(popus);
choose_N=3;                    %选择 choose_N 对双亲
picked=zeros(choose_N,2);      %记录选择好的双亲
p=zeros(M,1);                  %选择概率
d_order=zeros(M,1);

%把父代个体按适应度从大到小排序
f_t=sort(f,'descend');%将适应度按降序排列
for k=1:M
    x=find(f==f_t(k));%降序排列的个体序号
    d_order(k)=x(1);
end
for m=1:M
    popus_t(m,:)=popus(d_order(m),:);
```

```
end
popus=popus_t;
f=f_t;

p=f_std./sum(f_std);                          %选择概率
c_p=cumsum(p)';                               %累积概率

for cn=1:choose_N
    picked(cn,1)=roulette(c_p); %轮盘赌
    picked(cn,2)=roulette(c_p); %轮盘赌
    popus=intercross(popus,picked(cn,:));%杂交
end
popus=aberrance(popus,picked);%变异
return;

function popus=intercross(popus,picked) %杂交
[M_p,N_p]=size(picked);
[M,N]=size(popus);
for cn=1:M_p
    p(1)=ceil(rand*N);%生成杂交位置
    p(2)=ceil(rand*N);
    p=sort(p);
    t=popus(picked(cn,1),p(1):p(2));
    popus(picked(cn,1),p(1):p(2))=popus(picked(cn,2),p(1):p(2));
    popus(picked(cn,2),p(1):p(2))=t;
end
return;
function popus=aberrance(popus,picked) %变异
P_a=0.05;%变异概率
[M,N]=size(popus);
[M_p,N_p]=size(picked);
U=rand(1,2);

for kp=1:M_p
    if U(2)>=P_a        %如果大于变异概率，就不变异
        continue;
    end
    if U(1)>=0.5
        a=picked(kp,1);
    else
        a=picked(kp,2);
    end
    p(1)=ceil(rand*N);%生成变异位置
    p(2)=ceil(rand*N);
```

```
    if popus(a,p(1))==1%0 1 变换
        popus(a,p(1))=0;
    else
        popus(a,p(1))=1;
    end
    if popus(a,p(2))==1
        popus(a,p(2))=0;
    else
        popus(a,p(2))=1;
    end
end
return;

function picked=roulette(c_p) %轮盘赌
[M,N]=size(c_p);
M=max([M N]);
U=rand;
if U<c_p(1)
    picked=1;
    return;
end
for m=1:(M-1)
    if U>c_p(m) & U<c_p(m+1)
        picked=m+1;
        break;
    end
end
```

三、最终结果

```
   function value=custom_fitness(x)
value=0;
dataform =
importdata('C:\Users\lenovo\Documents\MATLAB\dataform.csv');
for i=1:469

y=dataform(2*i,x);
fitobject=fit(dataform(1,x)',y','smoothingspline');
sum=0;
for j=1:51
dif=abs(dataform(2*i,j)-fitobject(dataform(1,j)));
if dif<=0.5
    sum=sum+0;
elseif dif<=1
    sum=sum+0.5;
```

```
elseif dif<=2
    sum=sum+1.5;
elseif dif<=3
    sum=sum+6;
elseif dif<=5
    sum=sum+12;
else sum=sum+25;
end
end
value=value+sum+72;
end
value=value/469;
end

nvars=6;
LB=[1 1 1 1 1 1];
UB=[51 51 51 51 51 51];




options=gaoptimset('MutationFcn',@mutationadaptfeasible, ...
   'Display','iter');
fitnessfcn=@(x)custom_fitness(x);
[x,fval]=ga(fitnessfcn,nvars,[],[],[],[],LB,UB,[],1:6,options);
```

带入数据跑出结果为成本 89.24