参考他人报告或代码的申明

统计推断课程
2015 年秋季学期第 18 组
成员：周志明 学号 5140309059, 丁恒哲 学号 5140309057
在报告编写过程中，以下方面参考了往届报告，现列表说明：

主要参考项目	说明
代码方面	《统计推断在数模模数转换中的应用》，谢昊男，2014 年秋季学期，组号 06 在该组报告附录提供的程序代码基础上，进行了少量修改。
算法描述方面	《统计推断在数模转换系统中的应用》，谢昊男，2014 年秋季学期，组号 06 参考了该组报告的算法描述文字

除了以上注明的参考内容，和报告中列出的引用文献注解，本报告其他部分都不包含任何其他个人或集体已经发表或撰写过的作品成果。

统计推断在数模转换系统中的应用

组号：18　姓名：周志明　学号：5140309059，姓名：丁恒哲 学号：5140309057

摘要：统计推断在数模转换系统中的一种不可或缺的应用方法。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。本文将尝试通过选取尽量少的数据点，来推断大量数据的统计特性，从而得到最小的总成本，为该假定模块的批量生产设计一种成本合理而高效的传感特性校准（定标工序）方案。

关键词：统计推断，定标，模拟退火算法，MATLAB，拟合

1 引言

在工程实践与科学实验中，我们通常面对的对象是一个由多个变量组成的一个系统。我们通常要研究的就是这些变量之间的函数关系，我们最常遇到的是两个变量的系统，一个输入变量，一个输出变量。在理论上我们往往可以通过直接的计算就能得出两个变量之间很确切的函数关系，然而在实际的实践中，由于误差等原因，我们很难能够找到一种通用的、能够很进准的描述两个变量以及他们之间关系的函数。但是如果我们可以通过多种不同曲线拟合的方法来酸楚多种不同的反感，然后通过各方面综合的比较，给出一个对拟合方法的评价从而选出最优的方案。

2 题解与分析

2.1 题目分析

本实验一共给我们提供了 400 个样本，通过理解，我认为这次的定标实验的最终目的在于选取一种特定的拟合方式，通过一定的算法找到最优的取点方案，使得总定标数量最少，从而成本最小（其中总定标成本包括误差成本和取点成本）。误差成本就是先对每一个样本取出同样的位置点数，再从样本库之中获取并按选定的拟合方式来计算该样本的所有拟合值，然后对所有样本逐一完成定标，计算总误差和取点成本，最后对样本个数取平均值。找到最优的取点方案（最小成本的取点方案），就是才去相互比较的方法得到的，是优化组合的问题。

由于从 400 组样本中每个样本五十多个点数选取若干个点的方案数太过巨大，成本过高，所以不能使用穷举法，而要采用一定的启发式搜索算法来代替。在取点时，一方面取的点数尽可能少，因为这样可以降低取点成本，另一方面拟合要越接近越好，这样可以降低误差成本。

2.2 典型样本曲线分析

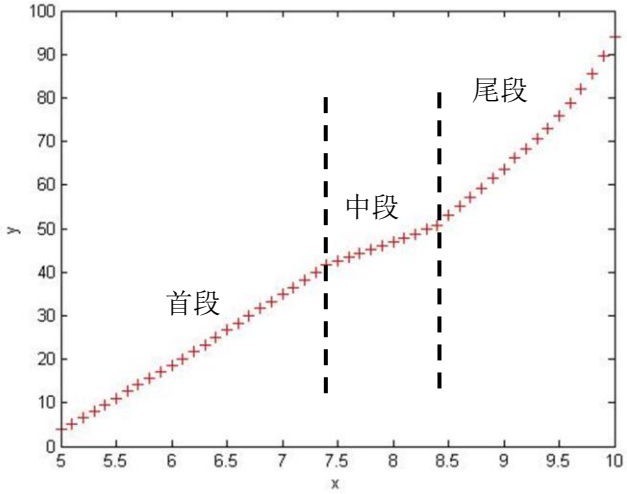
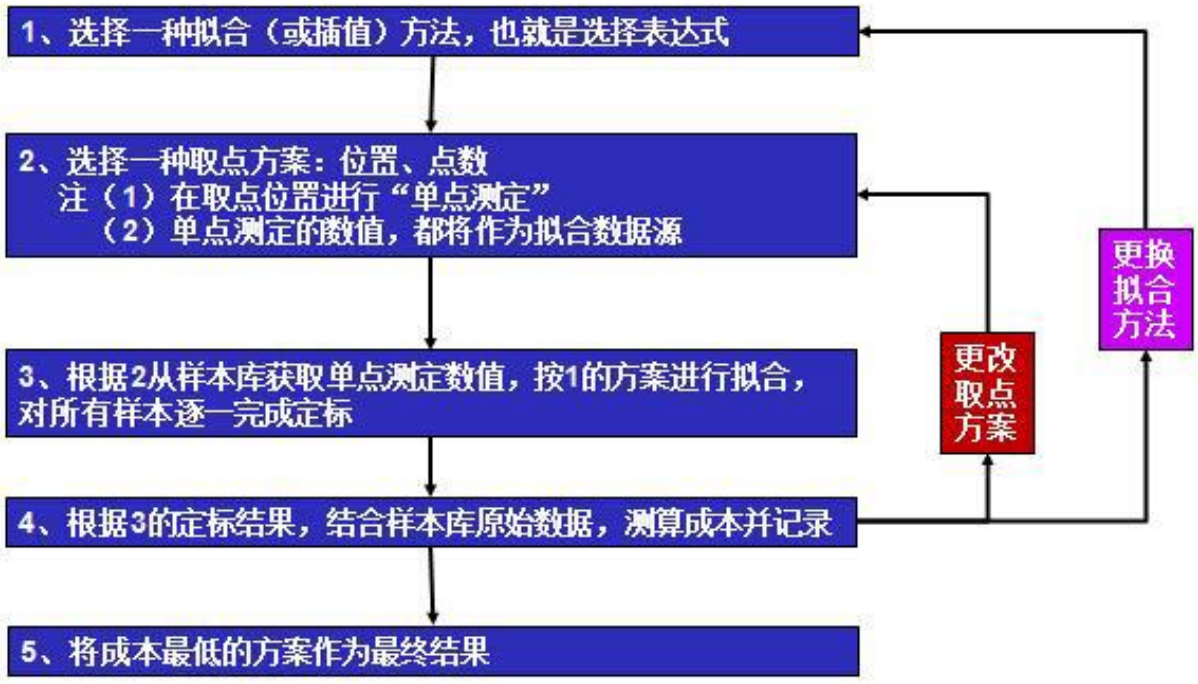


图 1 传感特性图示

故传感部件的输入输出特性大致如上面几幅图所示，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。

2.3 求解路径



3 成本计算

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

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

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。

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

总成本较低的校准方案，认定为较优方案。

4 算法选择

4.1 暴力穷举算法

穷举法的基本思想是根据题目的部分条件确定答案的大致范围，并在此范围内对所有可能的情况逐一验证，直到全部情况验证完毕。若某个情况验证符合题目的全部条件，则为本

问题的一个解；若全部情况验证后都不符合题目的全部条件，则本题无解。穷举法也称为枚举法。穷举法适用于条件比较少的情况，但是在本次试验中，要取得最优解，使用暴力穷举算法的时候，计算机必须要完成 1.15*10^8 左右次拟合才能逐一判断并找出最优解，因此，虽然理论上暴力穷举算法最为准确，但考虑到其时间空间复杂度，为了解决本实验的问题，我们采用了搜索启发式算法：模拟退火算法和遗传算法。

4.2 遗传算法

遗传算法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。其通常实现方式为一种计算机模拟。对于一个最优化问题，一定数量的候选解（称为个体）的抽象表示（称为染色体）的种群向更好的解进化。传统上，解用二进制表示（即 0 和 1 的串），但也可以用其他表示方法。进化从完全随机个体的种群开始，之后一代一代发生。在每一代中，整个种群的适应度被评价，从当前种群中随机地选择多个个体（基于它们的适应度），通过自然选择产生新的生命种群，该种群在算法的下一次迭代中成为当前种群。

4.3 模拟退火算法

模拟退火来自冶金学的专有名词退火。退火是将材料加热后再经特定速率冷却，目的是增大晶粒的体积，并且减少晶格中的缺陷。材料中的原子原来会停留在使内能有局部最小值的位置，加热使能量变大，原子会离开原来位置，而随机在其他位置中移动。退火冷却时速度较慢，使得原子有较多可能可以找到内能比原先更低的位置。

模拟退火的原理也和金属退火的原理近似：我们将热力学的理论套用到统计学上，将搜寻空间内每一点想像成空气内的分子；分子的能量，就是它本身的动能；而搜寻空间内的每一点，也像空气分子一样带有“能量”，以表示该点对命题的合适程度。算法先以搜寻空间内一个任意点作起始：每一步先选择一个“邻居”，然后再计算从现有位置到达“邻居”的概率。

可以证明，模拟退火算法所得解依概率收敛到全局最优解。

4.4 模拟退火算法和遗传算法的差异

模拟退火算法进化是由参数问题 t 控制的，然后通过一定的操作产生新的解，根据当前解的优劣和温度参数 t 确定是否接受当前的新解。

遗传算法主要由选择，交叉，变异等操作组成，通过种群进行进化。

主要不同点是模拟退火是采用单个个体进行进化，遗传算法是采用种群进行进化。模拟退火一般新解优于当前解才接受新解，并且还需要通过温度参数 t 进行选择，并通过变异操作产生新个体。而遗传算法新解是通过选择操作进行选择个体，并通过交叉和变异产生新个体。

相同点是都采用进化控制优化的过程。

我们组将模拟退火算法和遗传算法相比较得出，模拟退火算法更简洁，实现起来更加方便，而且更能找到准确的最优解，因此选择了模拟退火算法来解决本课题。

5 拟合方法

所谓拟合是指已知某函数的若干离散函数值{f1,f2,…,fn}，通过调整该函数中若干待定系数 f(λ1, λ2,…, λn)，使得该函数与已知点集的差别(最小二乘意义)最小。如果待定函

数是线性，就叫线性拟合或者线性回归(主要在统计中)，否则叫作非线性拟合或者非线性回归。表达式也可以是分段函数，这种情况下叫作样条拟合。

5.1 最小二乘法多项式曲线拟合

原理

给定数据点 pi(xi,yi)，其中 i=1,2,…,m。求近似曲线 y= Φ(x)。并且使得近似曲线与 y=f(x)的偏差最小。近似曲线在点 pi 处的偏差 δi= Φ(xi)-y，i=1,2,...,m。

常见的曲线拟合方法:

1.使偏差绝对值之和最小

2.使偏差绝对值最大的最小

3.使偏差平方和最小

按偏差平方和最小的原则选取拟合曲线,并且采取二项式方程为拟合曲线的方法,称为最小二乘法。

5.2 样条差值拟合

在数值分析这个数学分支中，样条插值是使用一种名为样条的特殊分段多项式进行插值的形式。由于样条插值可以使用低阶多项式样条实现较小的插值误差，这样就避免了使用高阶多项式所出现的龙格现象。从多项式拟合的失败，分析认为此处可以采用三次样条差值拟合计算成本。

由于每个三次多项式需要四个条件才能确定曲线形状，所以对于组成 S 的 n 个三次多项式来说，这就意味着需要 4n 个条件才能确定这些多项式。但是，插值特性只给出了 n + 1 个条件，内部数据点给出 n + 1 − 2 = n − 1 个条件，总计是 4n − 2 个条件。我们还需要另外两个条件，根据不同的因素我们可以使用不同的条件。

通俗一点讲，如果是去六点进行三次样条差值，则取相邻的四点得到的三次曲线就作为中间两点见的拟合曲线。由于成本与取点个数相关，这里并不能分析出取点多少对于成本结果的影响，故在后面实际计算时进行比较。

6 实验过程

(1) 读点 通过matlab内置函数xlsread或csvread读入数据；

(2) 取点 随机选取前七个点，作为初始选点方案。再次排序，得到随机的几个点；

(3) 循环 用模拟退火算法作为大循环，通过对退火初始温度和末温度的设定，以及每次降温幅度的设定，确定函数主体部分执行次数；

(4) 重组 在循环内前部分加入随机变化一个点，生成新的七个点组合；

(5) 拟合 用样条插值拟合，得到每点处拟合曲线的值，同时算出拟合值与实验值的差值；

(5) 计算 循环中间部分加入51个点的评价函数，计算并评测出当前取点方案的成本值。并计算400组数据的平均成本值，作为这次取点的总成本数；

(6) 评价 循环的后半部分决定是否接受这个取点方案。若计算成本分值小于当前最小成本分值，则接受该方案，同时将该方案作为当前最优方案，下次取点时以这个方案为基础；若大于，则以exp((score_save-cost)/Tk)的概率接受该方案。其中score_save 表示上一次接受的成本，cost 表示当前成本，Tk为温度；

(7) 结束 输出最优解及其所花费时间。

7 实验结果分析

7.1 程序运行结果演示

编号	运行结果	成本	时间
1	3 11 20 27 34 43 50	95.48	85.914
2	4 9 21 27 34 43 49	97.02	94.544
3	4 11 22 27 35 45 51	98.12	102.562
4	2 9 19 27 32 42 51	98.18	110.388
5	1 9 21 26 32 43 50	96.42	108.766
6	2 10 20 26 33 43 50	95.23	98.105

7.2 实验结果分析

由运行的结果可以看出，样条插值 7 点模拟退火的运行成本最低大概为 95.23，最高大概为 98.18；运行时间最少大概为 85.914 秒，最多大概为 110.388 秒。

由实验结果可以看出，7 个点大概可以较为精确的模拟出监测模块中传感器部件的输入输出特性，且用时较短，较为方便。

8 参考文献

[1]上海交大电子工程系. 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]“统计推断”课程设计的要求 *V2.2 2015-9-22*

[3]*http://blog.csdn.net/jairuschan/article/details/7517773/*

[5]维基百科 遗传算法、模拟退火算法

[6]《电脑知识与技术》 2013 年第 19 期 模拟退火算法与遗传算法的比较与思考

附录

```
DATA=csvread('20150915dataform.csv');%读入数据表中的数据
x=DATA(1:2:end,1:end);%将表格中的 x 分别取出
y=DATA(2:2:end,1:end);%将表格中的 y 分别取出
A=randperm(51);%随机打乱 51 个点
B=sort(A(1:7)); %随机取 7 个点，并进行排序
score_min=0;%总分数
score_save=0;%保存上一个数据成本
cost=0;%保存当次成本
num=0;%计数模拟的次数
B_min=B;%记录误差最小的情况
Tf=0.01;%末温度
Tk=10;%初始温度
tic;%开始运行记录时间
while Tk>Tf
num=num+1;
remain=setdiff(A,B);%寻求 A,B 差集
E=remain(randperm(44));%再次打乱顺序
F=randperm(7);
S=B;
S(1,F(1))=E(1,F(1)+1);
S=sort(S);%以上代码用于随机替换 7 个数中的一个数字

my_answer=S;%当前选点组合
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);%选点处的 x
y0_optimal=y0(:,index_temp);%选点处的 y
for j=1:nsample
```

```
% 通过调用 mycurvefitting 函数，实现三次样条插值拟合
y1(j,:)=mycurvefitting(x_optimal(j,:),y0_optimal(j,:));
end

%test_ur_answer:
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
score_min=cost;
score_save=cost;
B_min=S;
B=S;
elseif cost<score_min %花费代价小于上一次
%fprintf('\n 经计算，当前最优总成本为%5.2f\n',cost);
score_min=cost;
score_save=cost;
B_min=S;
B=S;
elseif rand<exp((score_save-cost)/Tk)%决定点是否变化
score_save=cost;
B=S;
end
Tk=Tk*0.97;%降温
end
toc;%记录结束的时间
fprintf('\n 经计算，最终最优总成本为%5.2f\n',score_min);
B_min%输出最优取点方案
```