



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

论文题目：统计推断在数模转换系统中的应用

作者：楼家辉 5140109116

高楚枫 5140109114

学院：电子信息与电气工程学院

专业：信息工程

目录

摘要.....	3
一、 引言.....	3
二、 算法的确定.....	3
三、 成本的计算.....	4
四、 样本数据拟合方式.....	4
五、 三种拟合方法的比较与选择成本计算.....	5
六、 样品选点：选点个数方案的确定.....	5
七、 解决问题的过程.....	6
八、 结论.....	7
九、 总结.....	7
十、 参考资料.....	7
十一、 致谢.....	8
附录.....	8

统计推断在数模转换系统中的应用

成员：楼家辉 5140109116 高楚枫 5140109114 组号 47

学院：电子信息与电气工程学院 专业：信息工程

摘要：工业生产中，常常需要对批量试生产的电子传感器进行输入输出特性的校准。本次实践中，我们组利用所学的统计推断的知识，我们在试验与比较中，在多项式拟合、三次插值拟合和三次样条插值拟合间选取了后者作为数据的拟合方案；并在实践后，选取了遗传算法进行数据的筛选。在实际的计算与对比中，我们组最终选择了如下的一套科学而高效的定标方案。

关键词：统计推断 多项式拟合 三次插值拟合 遗传算法 定标

一、引言：

对于一批试生产的电子传感器，现发现起输入输出特性呈明显的非线性，需要进行特性的校准。而三次样条插值拟合结果与样品数组符合得很好。拟合过程中，由于每组数据中都有 51 个点，一方面考虑到尽量多取点来增加提高成拟合的准确度，另一方面考虑到尽可能少地取点来节约成本。在编程计算后，我们最终决定选取 5 个点，使用三次插值拟合，又在 Matlab 中使用了遗传算法，计算出理想的选点方案。

二、算法的确定

1、枚举算法：在进行归纳推理时，如果逐个考察了某类事件的所有可能情况，因而得出一般结论，那么这结论是可靠的，这种归纳方法叫做枚举法。但枚举法需要计算的次数太多，例如在该问题中，如果取 7 个点，那么计算时间可能要达到几万年，这是不可接受的，因此我们在简单地考虑后放弃了这种办法。

2、遗传算法：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。遗传算法的主要过程是产生随机种群→产生后代(包含交叉、变异)→计算后代适应度，选择适应度高的后代，再进行上述过程，最终得到高适应度的个体，也就是近似的最优解。这种算法比简单的枚举要高效很多。我们所遇到的明显是一个 NP 问题，因此用遗传算法是一种很好的选择。

遗传算法的主要过程如下



【1】

三、 成本的计算

课题要求的成本包括了定标成本和误差成本，即需要综合考虑。
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

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

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

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

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。

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

总成本较低的校准方案，认定为较优方案。【2】

首先写了 `matlab` 函数计算成本，然后取成本的倒数，就可以作为适应度加入到遗传算法中。

四、 样品数据拟合方式

拟合：科学或工程上可以通过实验等方法获得关于某个问题的若干离散数据。依靠数学方法，使用连续函数（也就是曲线）或者更加密集的离散方程尽量逼近（即最小二

乘意义上的差别最小化) 这些已知离散数据点集, 此过程称为拟合。

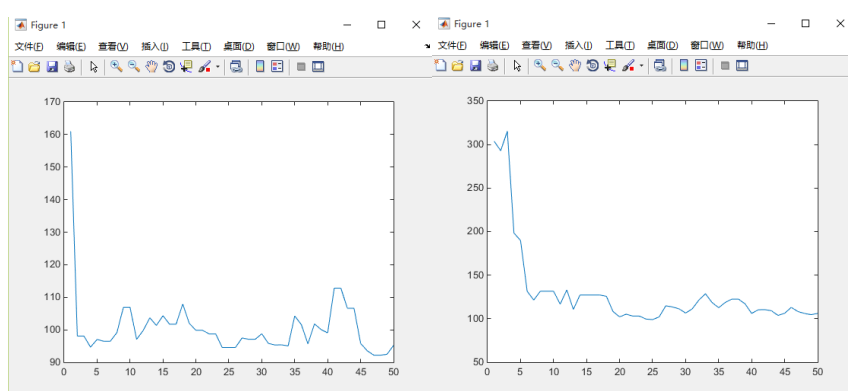
1.多项式拟合: 多项式拟合是一种简单的拟合方式, 通常用来拟合一些简单的数据点, 及用某些多项式来近似表示曲线, 但在实际应用中, 多项式拟合误差往往较大。

2.三次插值拟合: 三次插值拟合指用多段三次多项式来拟合数据点图线。

3.三次样条插值拟合: 三次样条插值(简称 Spline 插值)是通过一系列形值点的一条光滑曲线, 数学上通过求解三弯矩方程组得出曲线函数组的过程。**【3】**

五、 三种拟合方法的比较与选择成本计算:

我们首先写好了遗传算法, 通过遗传算法对 上述三种拟合方式进行比较, 其中多项式拟合的效果非常差, 在这里就不列举了。下面列举一下三次插值拟合(pchip 或 cubic)和三次样条插值拟合(spline), 我们分别用两种拟合, 取点个数都为 6 遗传 50 代, 并重复 10 次, 下面两张图是两种拟合方式分别在 10 次中最好的情况, 横坐标为遗传代数, 纵坐标为成本。



这是选点数为 6 的 pchip 拟合

可以看出成本的最小值出现在第 39 代, 其中第 39 代的选点方案为

[36 27 16 47 31 5]

成本为 93.5548

这是选点数为 6 的 spline 拟合

可以看出成本的最小值出现在第 25 代, 其中第 25 代的选点方案为

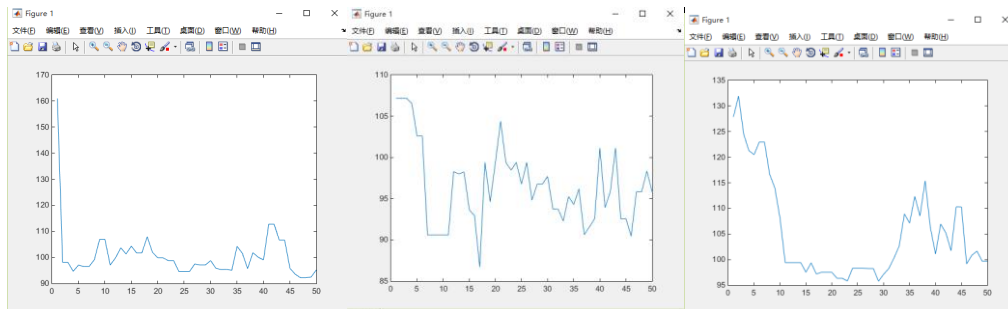
[32 20 3 49 11 43]

成本为 98.9310

从上述的比较可以看出, pchip 拟合略优于 spline 拟合, 因此我们最终选择了三次插值拟合即 pchip 作为我们的拟合方式。

六、 样品选点: 选点个数方案的确定

在这个问题中, 我们事先不知道要取几个点, 取点越多, 误差自然会越小, 但单个点定标也是有 12 成本的, 因此这之间的取舍只能通过计算来确定。通过大致的计算, 我们将选点个数确定在 5,6,7 之间, 因为其他选点个数计算出来的平均定标成本都会大于这三种, 而 5,6,7 之间的差距就较小, 下面是重复 10 次以后, 选点个数分别为 5,6,7 中的最好的情况。



这是取点个数为 6 的 pchip 拟合，方便起见，我们直接采用了在上文中进行拟合方式选择时的图和结论，即最小成本为 93.5548，选点方案为：

[36 27 16 47 31 5]

从上面可以看出，在三种选点个数的比拼中，选点个数为 5 的要略胜一筹，因此我们最终决定选取 5 个点，并跑多次，选择最好的那次。

七、 解决问题的过程

下面具体介绍我们用遗传算法解决问题的过程。

首先，要用遗传算法，我们需要将数字转化成二进制。这是我们面临的第一个问题。因为样本一共有 51 个点，而从 00000-11111，一共为 32 个数，从 000000-111111，一共为 64 个数，不能采用简单的二进制编码，否则在交叉和变异过程中会出现问题。经过考虑，我们采用一种特殊的编码方式，即从 000000000000-111111111111，其中前五个数字为一组，之后的四个为一组，之后的 2 个为一组，最后的一个为一组，这样前 5 个从 0 到 31，之后 4 个从 0 到 15，之后 2 个从 0 到 3，最后一个 0 到 1，把他们全部加起来，刚好是 0 到 50 即 51 个数，符合问题的要求，即一个 1 到 51 的数可以用一串长度为 12 的“基因”来表示。

确定好编码以后，就可以生成初始种群了，我们采用随机生成 0 或 1 的方式生成了包含 16 个个体的初始种群。

对于这个种群，用适应度函数计算适应度。我们将成本的倒数作为适应度，即先写出成本函数，然后取其倒数即可。

计算适应度之后要进行选择。我们采用的是轮盘选择赌法，即将所有个体的适应度加起来，再将个体的适应度除以总适应度就是这个个体存活下来的概率。然后用生成随机数的方式来确定该个体是否被选择。

在选择之后，就要进行交叉。我们的交叉概率选为 0.5，用生成随机数的形式确定两个个体之间是否交换，交换哪一部分。我们用随机数选择一个节点，将两条二进制链在该节点以后的部分互换。

交叉之后，就有一步变异的工作。其实在一开始，我们并没有写变异的部分，但后来发现若没有变异，遗传代数高了以后所有个体会全变成一样，这样遗传算法就失效了。通过查资料得知，这是因为陷入了“局部最优”，因此在后来我们补写了变异部分，遗传算法才得以很好地运转，我们选择了变异概率为 0.01。

进过选择——交叉——变异的过程，我们就可以得到新一代的种群，这一代种群再进行

这样的操作，就可以形成下一代。如此往复，就可以逼近最优解。

八、 结论

经过遗传算法的计算，我们最终得出了我们找到的最优解，最优解如下：

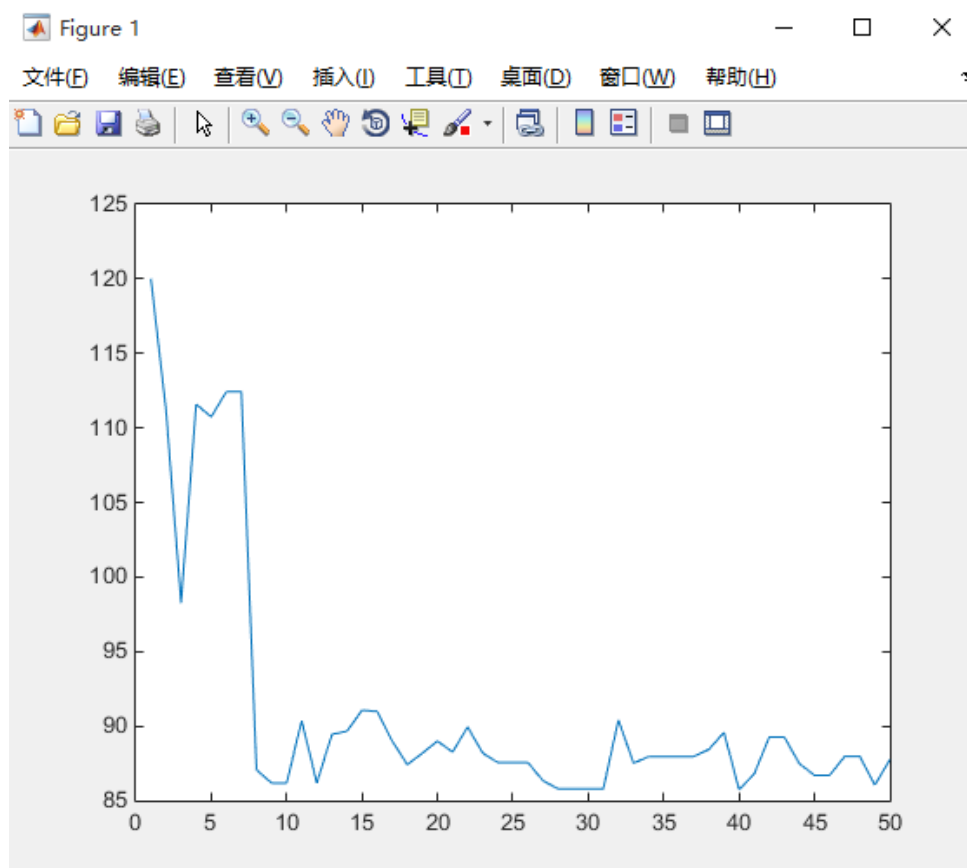
拟合方式：三次插值拟合，即 `pchip`

选点方案：选取[4 17 26 35 47]5 个点

平均定标成本： 85.7910

出现代数：第三十代

下面是出现我们所得最优解的遗传图像，横坐标为遗传代数，纵坐标为成本



九、 总结

通过这门课的学习，我们初步掌握了 `matlab` 软件的使用方法及基本编程语言，当然，最重要的还是通过自身实践理解了遗传算法。遗传算法是解决较复杂问题的一种很好的算法，虽然无法求出精确解，但可以让我们逼近最优解。就像我们所求得的选点方案，一定不是最优解，但已经接近了，已经可以运用到工业生产中。这种从生物中学来的想法也给了我们一种启示，当我们难以解决一些问题时，可以从大自然中寻找灵感。

十、 参考资料：

- 【1】好搜百科——遗传算法
- 【2】课程设计要求

【3】百度百科——三次样条插值

十一、 致谢：

感谢学院开设了统计推断在数模转换中的应用这门课程，在老师的讲解与指导下，在我们小组的交流讨论与合作实践中，我们对统计推断在工程实际中的应用有了更深刻的了解，并且，我们的将算法运用到实践的能力有了极大的提高，团队合作意识有也得到了增强。

感谢袁焱老师和李安琪的辛勤教导和帮助。

附录：

下面是我们的 matlab 代码

选择函数：select.m

```
function[alive]=select(fitness)
```

```
sum=0;
```

```
for i=1:16
```

```
    sum=sum+fitness(1,i);
```

```
end
```

```
alive_1=zeros(1,16);
```

```
sum1=0;
```

```
for i=1:16
```

```
    sum1=sum1+fitness(1,i)/sum;
```

```
    alive_1(1,i)=sum1;
```

```
end
```

```
alive=zeros(1,16);
```

```
for i=1:16
```

```
    a=rand(1,1);
```

```
alive(i)=(a>alive_1(1))+(a>alive_1(2))+(a>alive_1(3))+(a>alive_1(4))+(a>alive_1(5))+(a>alive_1(6))  
+(a>alive_1(7))+(a>alive_1(8))+(a>alive_1(9))+(a>alive_1(10))+(a>alive_1(11))+(a>alive_1(12))+(a  
>alive_1(13))+(a>alive_1(14))+(a>alive_1(15))+(a>alive_1(16))+1;
```

```
end
```

适应度函数：fitness.m

```
function[fitness1]=cost(X)
```

```
dataform=xlsread('dataform1.csv');
```

```
fitness_s=0;
```

```
for n=1:400
```

```
Y1=dataform(2*n,:);
```

```
xx=zeros(1,51);
```



```

Y=zeros(1,5);
for i=1:5
    s=round((X(i)-4.9)/0.1);
    Y(i)=dataform(2*n,s);
end
for i=1:51
xx(i)=4.9+0.1*i;
end;
yy=zeros(1,51);
for i=1:51
yy(i)=interp1(X,Y,xx(i),'pchip');
end;
for n=1:51
    if abs(Y1(n)-yy(n))<=0.4
        fitness_s=fitness_s+0;
    elseif 0.4<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=0.6
        fitness_s=fitness_s+0.1;
    elseif 0.6<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=0.8
        fitness_s=fitness_s+0.7;
    elseif 0.8<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=1
        fitness_s=fitness_s+0.9;
    elseif 1<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=2
        fitness_s=fitness_s+1.5;
    elseif 2<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=3
        fitness_s=fitness_s+6;
    elseif 3<abs(Y1(n)-yy(n))&&abs(Y1(n)-yy(n))<=5
        fitness_s=fitness_s+12;
    else abs(Y1(n)-yy(n))>5;
        fitness_s=fitness_s+25;
    end
end
end
cost=fitness_s/400+12*5;
fitness1=1/cost;

```

交叉函数 exchange.m

```

function[alive_1]=exchange(alive)
a=round(rand(1,1)*59)+1;
alive_1=zeros(16,60);
for i=1:2:15
    b=rand(1,1);
    for n=a:60
        if b>0.5
            alive_1(i,n)=alive(i+1,n);alive_1(i+1,n)=alive(i,n);

```

```

        else alive_1(i,n)=alive(i,n);alive_1(i+1,n)=alive(i+1,n);
    end
end
for s=1:a
    alive_1(i,s)=alive(i,s);alive_1(i+1,s)=alive(i+1,s);
end
end
end

```

变异函数 variation.m

```

function[new]=variation(a)
new=zeros(16,60);
for i=1:16
    for n=1:60
        new(i,n)=a(i,n);
    end
end
for i=1:16
    for n=1:60
        b=rand(1,1);
        if b<0.01
            if a(i,n)==0
                new(i,n)=1;
            else new(i,n)=0;
            end
        end
    end
end
end
end

```

遗传主代码（50 代） gen.m

```

a=rand(16,60)>0.5;
answer=zeros(1,50);
for h=1:50
    b=zeros(16,5);
    for n=1:16
        for i=1:12:49
            b(n,(i+11)/12)=a(n,i)*16+a(n,i+1)*8+a(n,i+2)*4+a(n,i+3)*2+a(n,i+4)*1+a(n,i+5)*8+a(n,i+6)*4+a(n,i+7)*2+a(n,i+8)*1+a(n,i+9)*2+a(n,i+10)*1+a(n,i+11)+1;
        end
    end
    fitnessall=zeros(1,16);
    for i=1:16
        fitnessall(1,i)=fitness(b(i,:));
    end
    costall=zeros(1,16);

```

```

for i=1:16
costall(i)=1/fitnessall(i);
end
disp(h);
disp(costall);
disp(b);
answer(h)=min(costall);
c=select(fitnessall);
d=zeros(16,60);
for i=1:16
    d(i,:)=a(c(i),:);
end
e=exchange(d);
a=variation(e);
end
x=zeros(1,50);
for i=1:50
    x(i)=i;
end
plot(x,answer)

```