统计推断在数模转换系统中的应用

第 18 组 张晨飞 5120309424 张瑞 5130309470

摘要：本文是上海交通大学电子信息与电气工程学院课程设计《统计推断在模数、数模转换系统中的应用》的课程论文。本文根据近年来某实验所得的数据建立数学模型，结合统计方法，根据已知的少量数据实现对其余更多的未知数据的推断。我们对多种拟合方法进行了比较，运用 MATLAB 工具对多种拟合方式进行了评判，最终我们选取了分三段三次拟合，运用“遗传算法”探索最佳结果。

关键词：统计推断，MATLAB，遗传算法，拟合

Application of Statistical Inference in AD&DA Inverting System（first draft）

ABSTRACT: This article is the thesis for Course Design Application of Statistical Inference in AD&DA Inverting System, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. This report sets up a mathematical model based on an experimental data obtained in recent years. It is combined with statistical methods to infer the large amount of unknown data based on a small amount of known data. We compared several fitting methods and evaluated them with the help of MATLAB. And then we choose a method that is the three curve fitting with splitting the curve in three parts, searching for the best points using genetic algorithm.

Key words:statistical inference，MATLAB，genetic algorithm，fitting

1 引言

本课程要研究的问题是为某种产品内部的一个检测模块寻求校准工序的优化方案。虽然不同的样本间的数据参数各不相同，但其对应曲线的趋势基本相同。因此，在工程应用中便提出了一种需求，即通过测定分析少量的数据对，推断出剩余未知数据对的信息，从而实现分析少量数据点推断出大量数据点的目的。本论文正是着手这一目的对此问题进行统计分析。

通过检测很多个样品，发现每个样品的 X-Y 特性关系曲线形状有相似性，故可以用统一的曲线方程式来表示 X-Y 函数关系，如图 1 所示。

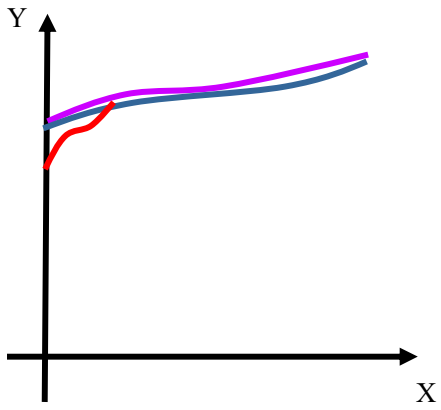

图 1 样品的 X-Y 函数图像

1.1 推断过程的数学模型

本文统计推断过程中的数学模型为：在样本点的采集数量有限定的情况下，确定一种样本点在曲线中分布以及由样本逼近曲线的方法所组成的联合方案，使得曲线能被最优逼近，即由样本点恢复的曲线与原曲线误差最小。

2 拟合方式与拟合函数的选择

拟合的方式有单段拟合和多段拟合两种方式。我们组最终选择了分三段分别三次拟合的方式。

2.1 单段拟合

我们组首先尝试了单段三次曲线拟合，首先先介绍一下三次拟合。

由 Taylor 公式

$$f(x)=\sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^n+\frac{f^{(k+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

令 $x_0=0$，有

$$f(x)=\sum_{k=0}^{n}\frac{f^{(k)}(0)}{k!}x^{n}+\frac{f^{(k+1)}(\xi)}{(n+1)!}x^{n+1}\text{，当}\frac{f^{(k+1)}(\xi)}{(n+1)!}x^{n+1}.$$

当 f(x)足够小时可以用多项式近似拟合图像曲线足够光滑的函数。

但是不是拟合的级数越高拟合的越准确呢？

在一定程度上是正确的，但是在特殊情况下是不符合物理规律的。所以单段曲线拟合不一定是最好的。况且考虑到计算量最多只能用三次拟合，故我们考虑使用多段曲线拟合。

2.2 分段拟合

由于对于该产品的几个样本的观察，我们得出样本的曲线大致可分为三个区间，我们假定整体系统特性也会呈现前后衔接的三个区间，每个区间适合分别用一段多项式曲线来拟合，前后区间对应的多项式曲线在区间衔接处取值连续。

2.2.1 概述

我们对 200 多组的图像进行了细致的观察，我们觉得应对三段都进行三次函数拟合，并通过比较每两个点之间的斜率来找出两个分界点，后面的遗传算法的应用中我们是采用分三段三次曲线拟合的。

我们应用相关软件对曲线进行了大致的拟合，结果如图 2-1 所示，

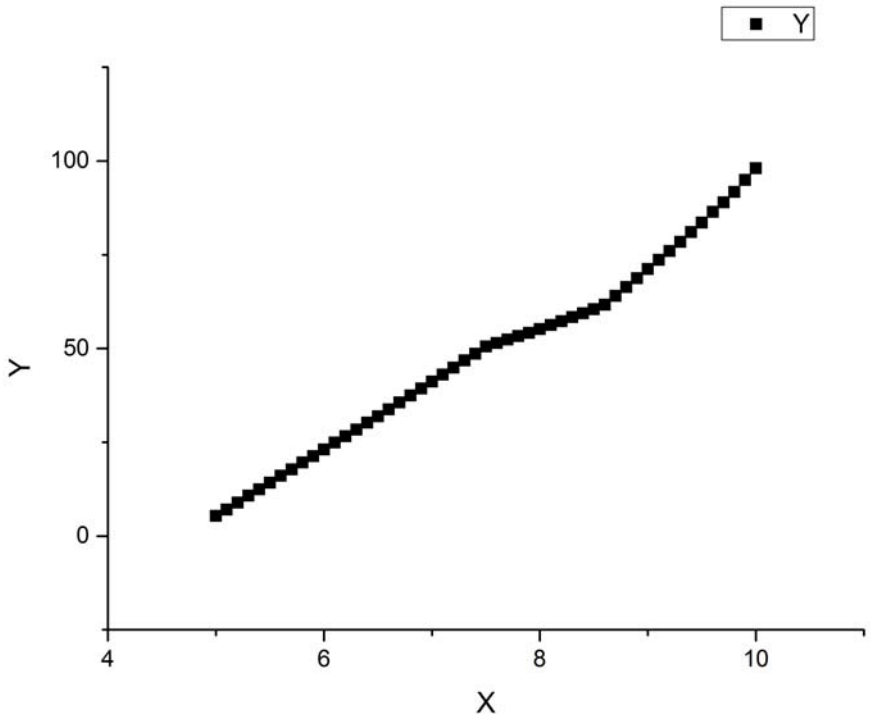

图 2-1 实验数据 X-Y 的散点拟合图

从上图中，我们看出，样本曲线大致分为三个部分，因此我们决定采用分三段三次拟合的方式，首先的工作是找出两个分界点。

2.2.2 分界点的确定

我们选择通过对每次实验选取的样本进行每两个点计算斜率然后进行比较取出两个斜率的突变点作为两个分界点，如公式 2-1 所示。

t=(y(i+1)-y(i))/(x(i+1)-x(i))-(y(i)-y(i-1))/(x(i)-x(i-1)) （2-1）

由于我们获得分界点的方法是每次实验样本都通过计算斜率来确定分界点，故增加了每次样本的计算的时间复杂度，但相比较于一次性确定分界点后连续使用的方法，我们的方法更为准确。

2.2.3 各段的拟合曲线次数的确定

区间 1：

$$U_i = \sum_{m=0}^{n_1} a_m D_i^{\,m} + \varepsilon_i \qquad \text{其中 } i=1,2,\ldots,N_1 \tag{2-2}$$

区间 2：

$$U_j = \sum_{l=0}^{n_3} b_l D_j^{\,l} + \varepsilon_j \qquad \text{其中 } j=N_1,\ldots,N_2 \tag{2-3}$$

区间 3：

$$U_k = \sum_{n=0}^{n_2} c_n D_k^{\,n} + \varepsilon_k \qquad \text{其中 } k=N_2,\ldots,51 \tag{2-4}$$

以上各式中 $\varepsilon_h \sim N(0,\sigma^2)$ 且相互独立，其中 h=1,2,3 ••• 51。

区间 1 与区间 2 衔接点两个方程连续且斜率相同：

$$\sum_{m=0}^{n_1} a_m D_{N_1}^{\,m} = \sum_{l=0}^{n_3} b_l D_{N_1}^{\,l} \tag{2-5}$$

解得：

$$a_0 = -\sum_{m=0}^{n_1} a_m D_{N_1}^{\,m} + \sum_{l=0}^{n_3} b_l D_{N_1}^{\,l} \tag{2-6}$$

同理可解得：

$$c_0 = -\sum_{n=0}^{n_2} c_n D_{N_2}^{\,n} + \sum_{l=0}^{n_3} b_l D_{N_2}^{\,l} \tag{2-7}$$

设误差平方和为：

$$\begin{aligned} Q_2 &= \sum_{h=1}^{51} \varepsilon_h^{\,2} \\ &= \sum_{i=1}^{N_1} (U_i - (\sum_{m=0}^{n_1} a_m D_i^{\,m}))^2 + \sum_{j=N_1}^{N_2} (U_j - (\sum_{l=0}^{n_3} b_l D_j^{\,l}))^2 + \sum_{k=N_2}^{51} (U_k - (\sum_{n=0}^{n_2} c_n D_k^{\,n}))^2 \end{aligned} \tag{2-8}$$

欲使 Q_2 达到最小，等价于求 $Q_2(a_1,a_2,...,a_{n_1},b_0,b_1,b_2,c_1,c_2,...,c_{n_2})$ 的最小值点，故可直接对数据进行多项式拟合。

469 组数据的计算结果见表 2-1.

表 2-1 分段拟合对于不同次数的比较

	3+3+3 次多项式	4+3+3 次多项式	3+3+4 次多项式	4+4+4 次多项式
Q_2的总和	2.9493	2.7730	2.9047	2.7286
Q_2的最大值	0.1480	0.1454	0.1473	0.1096

对结果进行比较，最终我们选择了三段都使用三次曲线拟合的方法。

2.2.4 拟合函数的实现

我们通过对多种拟合方式，以及不同次幂的拟合曲线的对比最终选择了分三段分别三次拟合的方式，首先通过对比每两点之间的斜率进而确定了两个分界点，然后我们通过掷筛子的方式随机决定每个样本选取几个点，然后在随机决定选取哪几个点，最后我们对三段曲线都进行了三次曲线拟合，并分别计算了成本然后加和，通过遗传算法最终得到了较为满意的解。

3 算法

3.1 遗传算法

3.1.1 概述

遗传算法（Genetic Algorithm）是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。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。

3.1.2 遗传算法的实现

（1）初始种群以及淘汰机制的确定

我们首先通过随机的方式决定每个样本选取几个测试点，如公式 3-1 所示，

```
w=round(rand(1,1)*50)+1;
```
（3-1）

然后我们再通过随机的方式决定哪些点作为测试点，如公式 3-2 所示，

```
t=round(rand(1,1)*50)+1;
```
（3-2）

通过这种方式，我们产生了种群数为 100 的初始种群。

种群个体适应度计算则根据给出的成本函数确定，在此不在赘述。

种群中个体的淘汰机制采用轮盘赌的方法，适应度越高（即成本越低）的个体存活概率越大。

（2）交叉及变异

我们通过将种群中的个体进行随机部分互换来进行交叉，并将变异的概率设置成了 0.01，即每个个体都有 0.01 的随机概率产生突变。算法框图如图 3-1 所示。

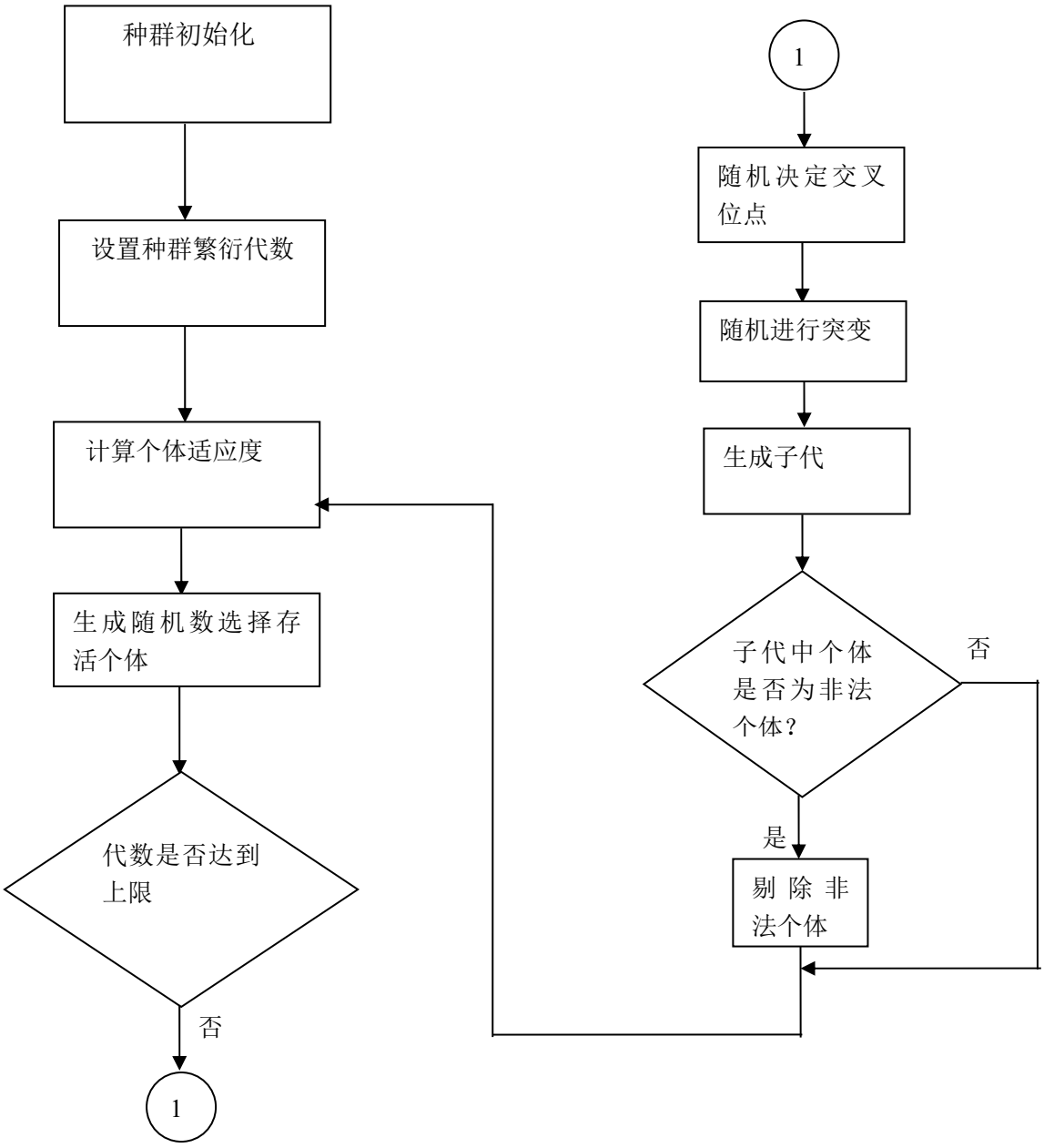

图 3-1 遗传算法的流程框图

3.2 最优解

我们组设置的参数为：

初始种群：100

遗传代数：50

变异率：0.01

在经过不断地试验之后，我们组得出的最优解为：

[5 17 27 31 35 46],最优成本为 99.1642。其中遗传算法实现代码请见附录 A

3.3 遗传算法的不足

通过使用遗传算法计算最优解的过程中，我们小组也发现了许多的遗传算法的不足：

（1）遗传算法的计算量非常的庞大，因为计算机进行分段多次拟合的过程耗时比较大，我们测算出 MATLAB 进行两次三次拟合和一次多段三次样条差值拟合后再运行评分函数得到结果大约需要 0.6 秒，然而通过我们遗传算法设置的初始参数，初始种群 100 以及交配 50 代使得电脑需要运行成千上万次上述过程，导致总运行时间非常的长。

（2）交叉用的是单点法，直接交叉两段基因，所以不利于搜索的全面性，易陷入局部极值。

（3）在选择下一代群体时，最佳个体的生存机会将显著增加，最差个体的生存机会将被剥夺，低适值个体淘汰太快容易使算法收敛于局部最优解。

4 结论

在大半个学期的学习实践过程中，我们小组遇到了许多的困难，也学习了许多的知识，同样也收获了许多。我们小组在学习 MATLAB 的用法上花费了大量的时间，之后随着对 MATLAB 的逐渐熟悉，我们小组的工作也进展的越来越顺利，我们首先使用了分三段一次拟合的方式，但结果不是很理想，最后经过面谈时老师的悉心教导，我们最终选择了分三段三次拟合的方法，并得到了显著地成效，经过 50 代的遗传，最后我们小组得到了较为满意的最优解，最优解为[5 17 27 31 35 46],最优成本为 99.1642。

最后，衷心感谢老师的悉心教导，这门课使我们受益匪浅。

5 参考文献

[1] 上海交大电子工程系. 统计推断讲座 1,2,3 ftp://202.120.39.248.

[2] 百度百科 词条“遗传算法，曲线拟合”

[3] 贺才兴等 概率论与数理统计 科学出版社 2007

[4] 薛定宇, 陈阳泉 高等应用数学问题的 MATLAB 求解 北京: 清华大学出版社, 2004

[5]陈长征 王楠。《遗传算法中交叉和变异概率选择的自适应方法及作用机理》

附录 A：遗传算法代码实现

```
function f=s2(x,y)
s=0;
for i=1:51
    if (0.5<abs(x(i)-y(i))&&abs(x(i)-y(i))<=1)
        s=s+0.5;
    end
    if (1<abs(x(i)-y(i))&&abs(x(i)-y(i))<=2)
        s=s+1.5;
    end
    if (2<abs(x(i)-y(i))&&abs(x(i)-y(i))<=3)
        s=s+6;
    end
    if (3<abs(x(i)-y(i))&&abs(x(i)-y(i))<=5)
        s=s+12;
    end
    if (abs(x(i)-y(i))>5)
        s=s+25;
    end
end
f=s;
end


function f=cb(x,m)
   n=0;
   for i=1:51
       n=n+x(i);
   end
   if (n>=3)
   x1=zeros(n,1);x2=zeros(n,1);
   y1=zeros(n,1);
   t=0;
   for i=1:51
       if (x(i)==1)
           t=t+1;
           x1(t)=i/10+5;
           x2(t)=i;
       end
   end
   s=0;
   for i=1:469
       for j=1:n
```

```
            y1(j)=m(i*2,x2(j));
        end
        y2=nihe(x1,y1,n);
        t=s2(y2,m(i*2,1:51));
        s=s+t;
    end
    s=s/469;
     s=s+n*12;
    else
      s=999;
    end
f=s;
end


function f=nihe(x,y,n)
  min=999999;mi=2;max=0;ma=n-1;
  for i=2:n-1
      t=(y(i+1)-y(i))/(x(i+1)-x(i))-(y(i)-y(i-1))/(x(i)-x(i-1));
      if t<min
          min=t;mi=i;
      end
      if t>max
          max=t;ma=i;
      end
  end
  x1=zeros(mi,1);
  y1=zeros(mi,1);
  for i=1:mi
      x1(i)=x(i);y1(i)=y(i);
  end
  f1=polyfit(((x1-1)/10+5),y1,3);
  x2=zeros(ma-mi+1,1);
  y2=zeros(ma-mi+1,1);
  for i=mi:ma
      x2(i-mi+1)=x(i);y2(i-mi+1)=y(i);
  end
  f2=polyfit(((x2-1)/10+5),y2,3);
   x3=zeros(n-ma+1,1);
  y3=zeros(n-ma+1,1);
  for i=ma:n
      x3(i-ma+1)=x(i);y3(i-ma+1)=y(i);
  end
  f3=polyfit(((x3-1)/10+5),y3,3);
```

```
  y4=zeros(51,1);
  for i=1:51
      xt=i;xt=xt/10+5;
      if (xt<x(mi))

yt=f1(1)*((xt-1)/10+5)*((xt-1)/10+5)*((xt-1)/10+5)+f1(2)*((xt-1)/10+5
)*((xt-1)/10+5)+f1(3)*((xt-1)/10+5)+f1(4);
      end
      if (xt>=x(mi)&&xt<x(ma))

yt=f2(1)*((xt-1)/10+5)*((xt-1)/10+5)*((xt-1)/10+5)+f2(2)*((xt-1)/10+5
)*((xt-1)/10+5)+f2(3)*((xt-1)/10+5)+f2(4);
      end
      if (xt>x(ma))

yt=f3(1)*((xt-1)/10+5)*((xt-1)/10+5)*((xt-1)/10+5)+f3(2)*((xt-1)/10+5
)*((xt-1)/10+5)+f3(3)*((xt-1)/10+5)+f3(4);
      end
      y4(i)=yt;
  end
  f=y4;
end


function f=yichuan(m)
  x=zeros(100,51);
  x1=zeros(100,51);
  for i=1:100
      w=round(rand(1,1)*50)+1;
      for j=1:w
         t=round(rand(1,1)*50)+1;
         x(i,t)=1;
      end
  end
  y=zeros(101);
  for p=1:50
      st=0;
       for i=1:100
          y(i)=1.0/cb(x(i,1:51),m);
          st=st+y(i);

       end
      y(100)=99999999;
      for q=1:100
```

```
      t1=rand;t1k=1;
      while (t1>y(t1k)/st)
          t1=t1-y(t1k)/st;
          t1k=t1k+1;
      end
       t2=rand;t2k=1;
      while (t1>y(t2k)/st)
          t2=t2-y(t2k)/st;
          t2k=t2k+1;
      end
      t3=rand;t3=t3*51;
      t3k=1;
      while (t3k<t3)
          x1(q,t3k)=x(t1k,t3k);
          t3k=t3k+1;
      end
      while (t3k<=t1)
          x1(q,t3k)=x(t2k,t3k);
          t3k=t3k+1;
      end
      t4=rand;
      if (t4<0.01)
          t5=rand*50+1;
          t5=floor(t5);
          x1(q,t5)=1-x1(q,t5);
      end
      end
      x=x1;
      p
  end
  min=9999999;mi=0;
 for i=1:100
     t=cb(x(i,1:51),m);
     if (t<min)
         min=t;
         mi=i;
     end
 end
 min
 x(mi,1:51)
end
```