

# 统计推断在数模转换中的运用

组号：12 姓名：罗舞 学号：5130309424 姓名：王泽龙 学号：5130309427

**摘要：**本报告以上海交通大学工程实践与科技创新的实验结果为研究数据，探讨统计推断在数模模数转换中的应用。报告中采用多种拟合方法，探求满足所有数据的统一形式的函数。为了减小运算量，报告采用特征点选取的方法，我们应用了模拟遗传算法和随机数函数算法，通过 matlab 工具来求得满足评价函数的局部最优解。

**关键词：**特征点，插值，曲线拟合，matlab，模拟遗传算法

## 1. 引言：

请阅读《课程设计课题和要求》。

## 2. 寻找满足条件的拟合函数

### 2.1. 对于样本数据的观察

选择样本 1, 118, 236, 352, 469 进行描点绘图，得到图 1(a)、(b)、(c)、(d)、(e)

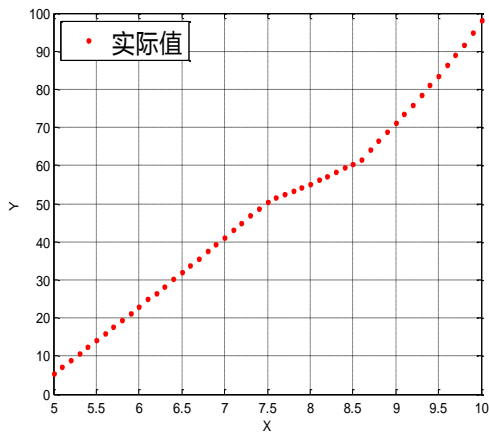


图 1 (a) 样本 1 的 Y-X 图

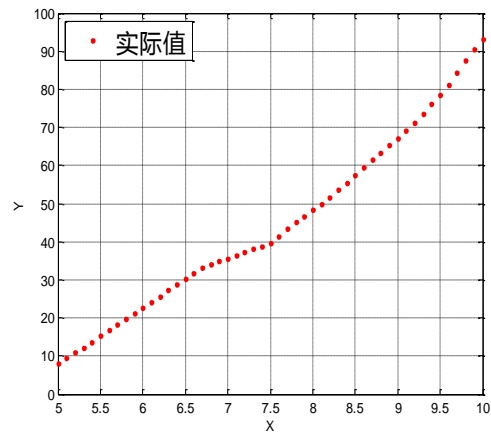


图 1 (b) 样本 118 的 Y-X 图

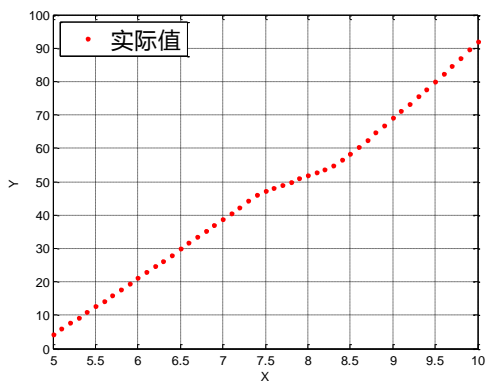


图 1 (c) 样本 236 的 Y-X 图

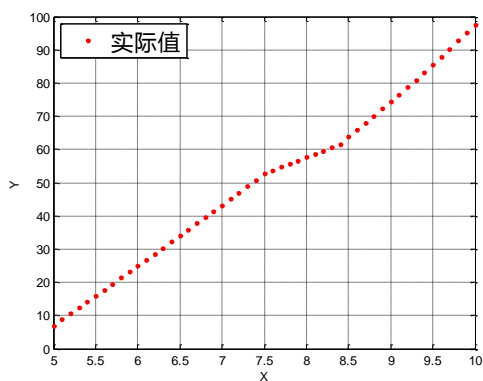


图 1 (d) 样本 352 的 Y-X 图

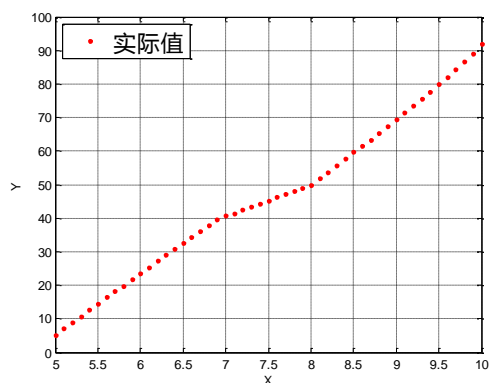


图 1 (e) 样本 469 的 Y-X 图

通过对以上 5 幅图进行观察，发现，曲线明显有两个拐点，所以宜采用三次多项式或者三次样条的方法进行拟合。

## 2.2 对数据进行三次样条插值

### 2.2.1. 三次多项式拟合插值介绍

三次多项式拟合即运用统计学相关知识，利用已知点计算出三次多项式中各项的系数，并对未知点进行插值预测。在 matlab 中，三次多项式拟合通过函数 `polyfit(x, y, 3)` 实现。

### 2.2.2. 三次样条插值介绍

三次样条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。

定义:函数  $S(x) \in C^2[a, b]$ ，且在每个小区间  $[x_j, x_{j+1}]$  上是三次多项式，其中  $a = x_0 < x_1 < \dots < x_n = b$  是给定节点，则称  $S(x)$  是节点  $x_0, x_1, \dots, x_n$  上的三次样条函数。

若在节点  $x_j$  上给定函数值  $Y_j = f(x_j)$  ( $j = 0, 1, \dots, n$ )，并成立  $S(x_j) = y_j$  ( $j = 0, 1, \dots, n$ )，则称  $S(x)$  为三次样条插值函数。

实际计算时还需要引入边界条件才能完成计算。边界通常有自然边界（边界点的二阶导为 0），夹持边界（边界点导数给定），非扭结边界（使两端点的三阶导与这两端点的邻近点的三阶导相等）。

在本次课题中，三次样条插值由 matlab 提供的内置函数 `inter1(xx, yx, x, 'spline')` 完成。

## 3 找出满足条件的数据

### 3.1. 模拟遗传算法介绍

遗传算法是计算数学中用于解决最佳化的搜索算法，是进化算法的一种。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），通过自然选择和突变产生新的生命种群，该种群在算法的下一次迭代中成为当前种群。

### 3.2. 模拟遗传算法的步骤

遗传算法是解决搜索问题的一种通用算法，对于各种通用问题都可以使用。搜索算法的

共同特征为：

- ① 首先组成一组候选解；
- ② 依据某些适应性条件测算这些候选解的适应度；
- ③ 根据适应度保留某些候选解，放弃其他候选解；
- ④ 对保留的候选解进行某些操作，生成新的候选解。

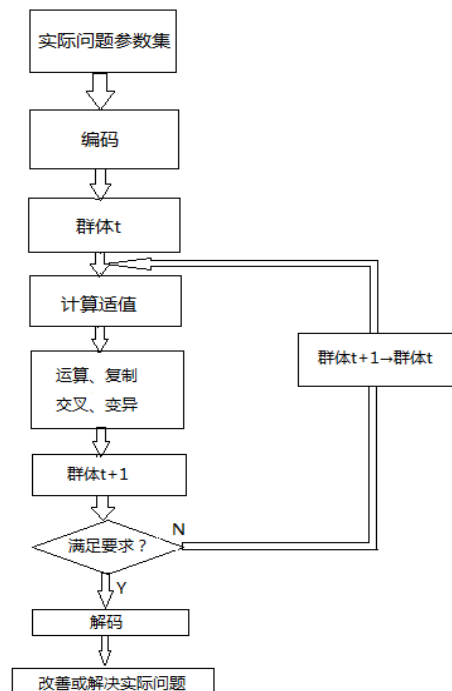


图 2 遗传算法流程示意图

在遗传算法中，上述几个特征以一种特殊的方式组合在一起：基于染色体群的并行搜索，带有猜测性质的选择操作、交换操作和突变操作。这种特殊的组合方式将遗传算法与其它搜索算法区别开来

### 3. 2. 3. 遗传算法在该问题的运用

本课题中，遗传算法用来搜索对于某种特定的插值方式（我们使用了三次多项式拟合以及三次样条插值），“最佳”的取点方式。所谓“最佳”，是指该取点方案的成本（包括插值误差成本和单次测量成本）尽可能小。具体方法如下：

(1) 初始化种群：我们将某种取点方案视作一个生物个体，数个生物个体构成种群。程序中的参数size\_of\_population（第10行）为种群中个体数，我们将其设为200；参数number\_of\_generation（第11行）为种群发生世代交替的代数，我们将其设为30。变量populaion是个size\_of\_population\*52的矩阵，第i行第j（j>0, j<52）个元素为1表示第i中取点方案中，第j个点被选中；第i行第j（j>0, j<52）个元素为0表示第i中取点方案中，第j个点没有被选中。第i行第52个数据用来记录该行表示的选点方案成本。我们通过随机函数产生出初始种群

(2) 计算适应度：

我们先采用的拟合的方式为三次多项式：运用三次多项式对数据每一个样本逐一进行拟合，并计算每一种选点方式的成本。取适应度为成本的倒数，然后所有个体的适应度除

以适应度最高个体（最优解）的适应度得到该个体产生下一代的概率，最后通过随机数确定下一代具体的“基因型”（即选点方式）

### (3) 交叉互换并变异

在产生的下一代种群中交换“基因”的片段，并模拟编译过程，得到的新一代再进行成本以及适应度的运算，如此往复。

三次样条拟合的计算过程与上面三次多项式所述完全一致，只是将拟合方式从polyfit改为spline就可以

注意：附录已提供源代码，改代码采用的是三次样条插值，如果要得到三次多项式拟合的结果，只需将57行代码改为

```
y1(j,:) = polyval(polyfit(x_optimal,y0_optimal(j,:),3),x);
```

即可。

## 5. 最终结论

### 5.1. 最佳的数据

以三次多项式拟合得到的最优选点

1、14、27、38、51，其成本为 122.79。

以三次样条插值得到的最优选点

1、10、21、31、43、51，其成本为 95.41。

对比之下，我们选择以三次样条插值为插值方法，点1, 9, 19, 26, 33, 44, 51，作为选点的定标方案。

### 5.2. 对所得方案进行统计分析

图3为误差关于X的分布直方图，图4为各样本平均误差频数分布直方图，图5为各样本最大误差频数分布直方图

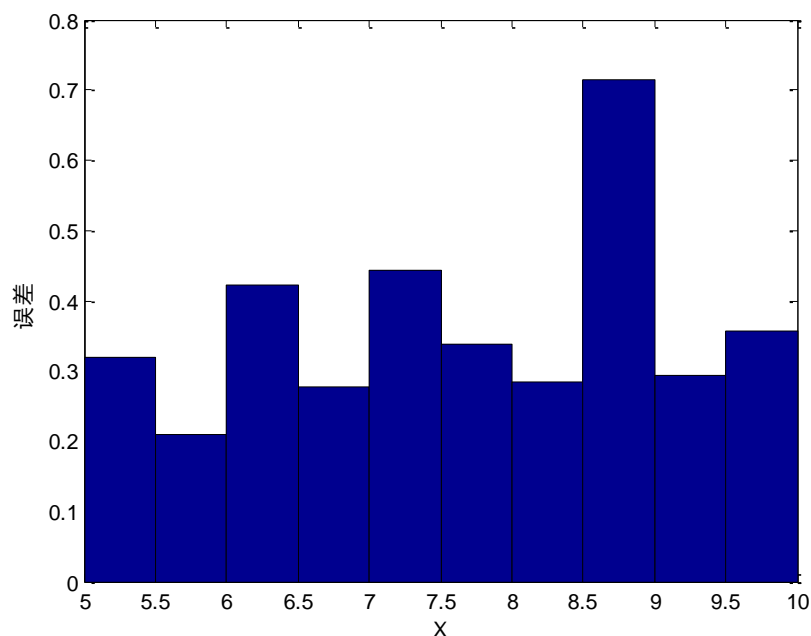


图 3 为误差关于 X 的分布直方图

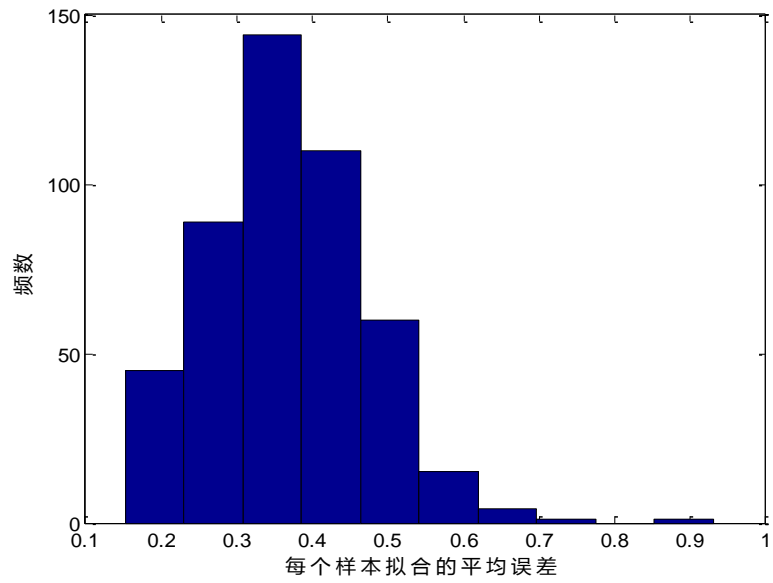


图 4 为各样本平均误差频数分布直方图

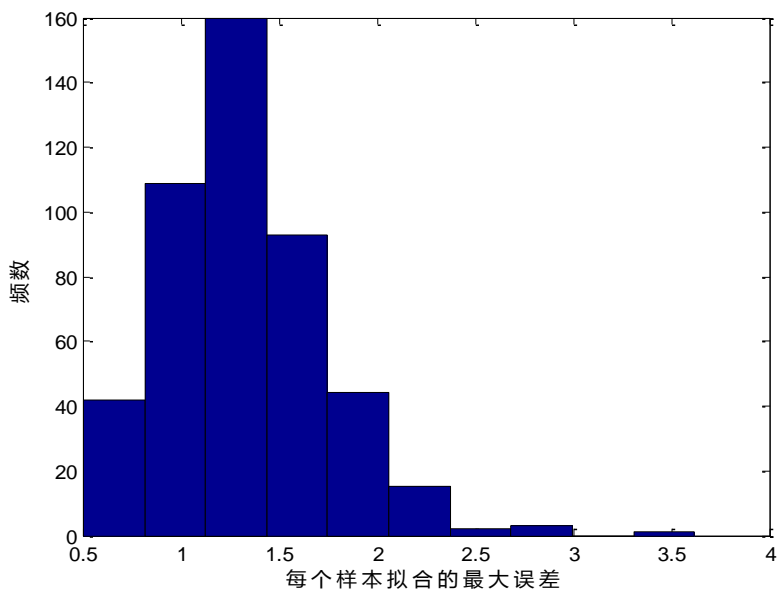


图 5 为各样本最大误差频数分布直方图

误差 区间	0~ 0.1	0.1~ 0.2	0.2~ 0.3	0.3~ 0.4	0.4~ 0.5	0.5~ 0.6	0.6~ 0.7	0.7~ 0.8	0.8~ 0.9	0.9~ 1
平均 误差	45	89	114	110	60	15	4	1	0	1
误差 区间	0.5~ 0.85	0.85~ 1.2	1.2~ 1.55	1.55~ 1.9	1.9~ 2.25	2.25~ 2.6	2.6~ 2.95	2.95~ 3.3	3.3~ 3.65	3.65~ 4
最大 误差	42	109	160	93	44	15	2	3	0	1

表 1 样本平均误差与最大误差频数分布

从图3看出曲线的两端拟合效果较好，中间稍差。从图4图5表1分析得出各个样本的平均误差最大误差都比较小，平均误差分布在0.15~0.5之间，最大误差分布在0.5~2之间，误差在允许的范围。图6为样本实际值与拟合值的对比。

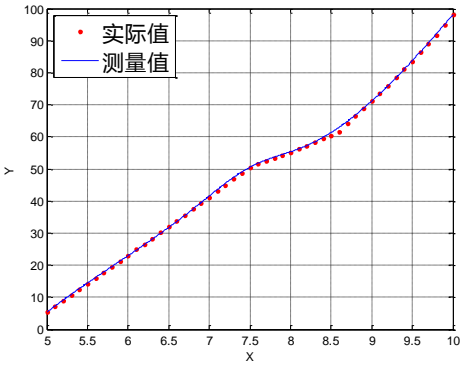


图 6 (e) 样本 1 的 Y-X 图

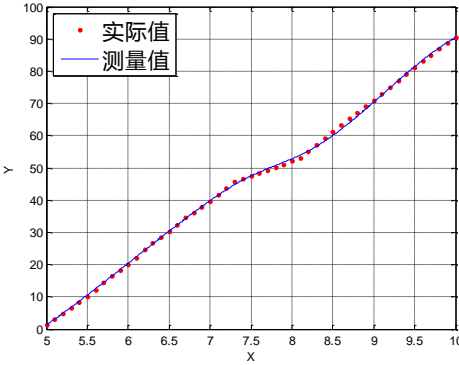


图 6 (e) 样本 118 的 Y-X 图

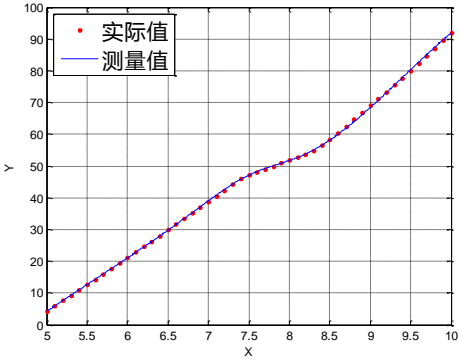


图 6 (e) 样本 236 的 Y-X 图

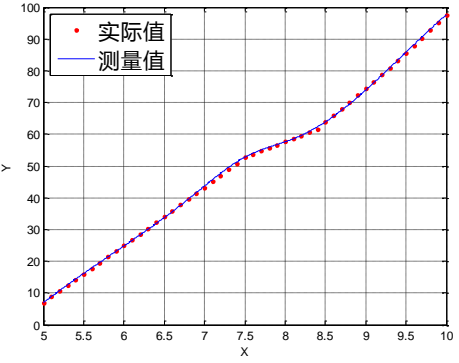


图 6 (e) 样本 352 的 Y-X 图

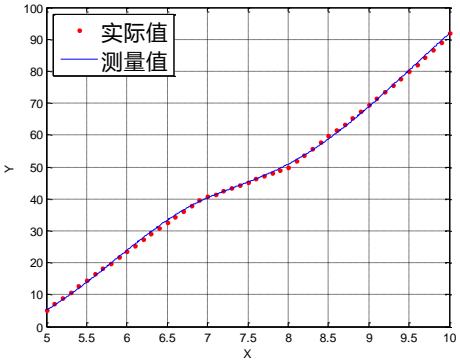


图 6 (e) 样本 469 的 Y-X 图

## 6. 参考文献

- 1、袁炎《上海交大电子系课程“统计推断在ADC、CDA转换中的运用”讲义》；

- 2、张志涌《精通Matlab》；
- 3、百度百科词条“三次样条”。

## 7. 附录：代码

注：本代码在 matlab2012b 测试通过

```
1 function go
2 %准备工作：产生初始种群
3 global y0;
4 global population;
5 global fitness;
6 global newPopulation;
7 global best; %记录最优解
8 best = zeros(1,52);
9 best(52) = inf;
10 size_of_population = 200;
11 number_of_generation = 30;
12 data = xlsread('20141010dataform.csv'); %读取数据
13 [row,col] = size(data);
14 sum_of_sample = row/2;
15 population = zeros(size_of_population,col+1); %种群储存在变量
16 %population中，population每行代表一个个体，第i行第j（j>0,j<52）个元素为1
17 %表示第i中取点方案中，第j个点被选中；第i行第j（j>0,j<52）个元素为0表示第i中
18 %取点方案中，第j个点没有被选中。第i行第52个数据用来记录该行表示的选点方案成本
19 population(:,1) = 1;
20 population(:,51) = 1;
21 newPopulation = zeros(size_of_population,col);
22 y0 = zeros(sum_of_sample,col); %记录课题提供数据中变量y的值
23 fitness = zeros(1,size_of_population); %记录个体适应度

24 for i = 1:sum_of_sample
25     y0(i,:) = data(2*i,:);
26 end
27
28 %产生初始种群
29 for i = 1:size_of_population
30     num = randi([4,col-2]); %
31     population(i,randperm(49,num)+1) = 1;
32 end
33
34 %模拟种群世代交替
35 for j = 1:number_of_generation
36     index_m =
37     calculatefitness(size_of_population,sum_of_sample,col,j);
38     nextGeneration(size_of_population,index_m);
```

```

39 end
40 fprintf('\n\n运算产生的最优解是: \n');
41 disp(find(best(1,1:51) ~= 0));
42 fprintf('成本为%5.2f\n',best(52));
43
44
45 function index_m =
46 calculatefitness(size_of_population,sum_of_sample,col,num)
47 global y0;
48 global population;
49 global fitness;
50 global best;
51 x = 5:0.1:10;
52 y1 = zeros(sum_of_sample,col);
53 for i = 1:size_of_population
54     %定标
55     index = logical(population(i,1:51));
56     for j = 1:sum_of_sample
57         y1(j,:) = interp1(x(index),y0(j,index),x,'spline');
58     end
59     %计算成本
60     Q = 12;
61     errabs = abs(y0-y1);
62     le0_5 = (errabs<=0.5);
63     le1_0 = (errabs<=1);
64     le2_0 = (errabs<=2);
65     le3_0 = (errabs<=3);
66     le5_0 = (errabs<=5);
67     g5_0 = (errabs>5);
68     sij =
69     0.5*(le1_0-le0_5)+1.5*(le2_0-le1_0)+6*(le3_0-le2_0)+12*
70     (le5_0-le3_0)+25*g5_0;
71     si = sum(sij,2)+Q*sum(population(i,1:col));
72     population(i,52) = sum(si)/sum_of_sample;
73 end
74 %适应度取成本的倒数
76 mean_of_cost = sum(population(:,52))/size_of_population;
77 fitness = 1./population(:,52);
78 [ma,index_m] = max(fitness);
79 if population(index_m,52) < best(52)
80     best = population(index_m,:);
81 end
82 fitness = fitness ./ ma;
83 fprintf('\n\n第%d代种群中, 最优解为',num);

```



```

84 disp(find(population(index_m,1:51) ~= 0));
85 fprintf('最小成本为%5.2f',population(index_m,col+1));
86 fprintf('    平均成本为%5.2f',mean_of_cost);
87 function nextGeneration(size_of_population,index_m)
88 global population;
89 global newPopulation;
90 global fitness;
91 newPopulation(1,:) = population(index_m,1:51); %保护最优解
92 survival = zeros(1,size_of_population);
93 survival(1) = fitness(1);
94 for i = 2:size_of_population
95     survival(i) = survival(i-1) + fitness(i);
96 end
97 %产生下一代
98 for j = 2:size_of_population
99     r = rand;
100     for k = 1:size_of_population
101         if r < survival(k)
102             newPopulation(j,:) = population(k,1:51);
103             break;
104         end
105     end
106 end
107 population = newPopulation(:,1:51);
108
109 %交叉互换并且变异
110 tmp = population(2,1:25);
111 for i = 2:size_of_population-1
112     population(i,1:25) = population(i+1,1:25); %交叉互换
113     population(i,randi([2,50])) = randi([0,1]); %变异
114     population(i,randi([2,50])) = randi([0,1]);
115     if sum(population(i,1:51)) < 4 %当取点过少时，无法进行拟合，这时将原
116                                     %来的
117         num = randi([3,49]); %点去除，再通过随机数产生一个新个体
118         population(i,randperm(49,num)+1) = 1;
119     end
120 end
121 population(size_of_population,1:25) = tmp;
122 population(i,randi([2,50])) = randi([0,1]);
123 population(i,randi([2,50])) = randi([0,1]);
124 if sum(population(i,1:51)) < 4
125     num = randi([3,49]);
126     population(i,randperm(49,num)+1) = 1;
127 end

```