统计信号处理第二次大作业报告

无88 刘子源 2018010895

一、题目描述

给定特征方程

$$\lambda^3 + a\lambda^2 + b\lambda + c = 0$$

且

$$\lambda_1 \geq \lambda_2 \geq \lambda_3$$

令

$$p_i = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3}$$

试利用上述特征多项式的系数近似表示下面的表达式

$$H = -\sum_{i=1}^{3} p_i log_3 p_i$$

二、公式化简

将特征多项式进行因式分解：

$$(\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) = 0$$

可得到一元三次方程的韦达定理：

$$\lambda_1 + \lambda_2 + \lambda_3 = -a$$

$$\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 = b$$

$$\lambda_1\lambda_2\lambda_3 = -c$$

由此得到$a < 0, b > 0, c < 0$

将$p_i = \frac{\lambda_i}{\lambda_1+\lambda_2+\lambda_3}$代入原方程，得到关于p的方程：

$$p^3 - p^2 + \frac{b}{a^2}p - \frac{c}{a^3} = 0$$

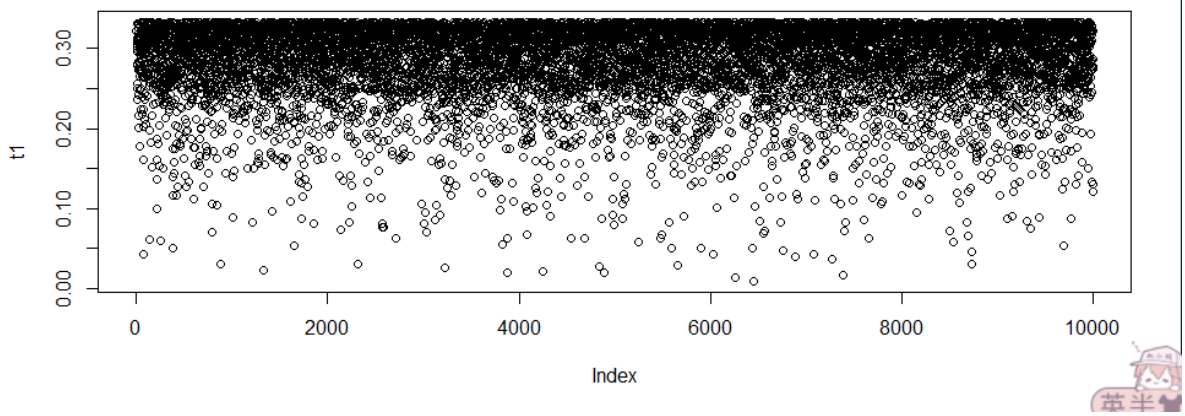
这样可以将a、b、c三个参数化简为$\frac{b}{a^2}$和$\frac{c}{a^3}$两个参数，接下来要寻找H与其的关系。

三、探索性分析

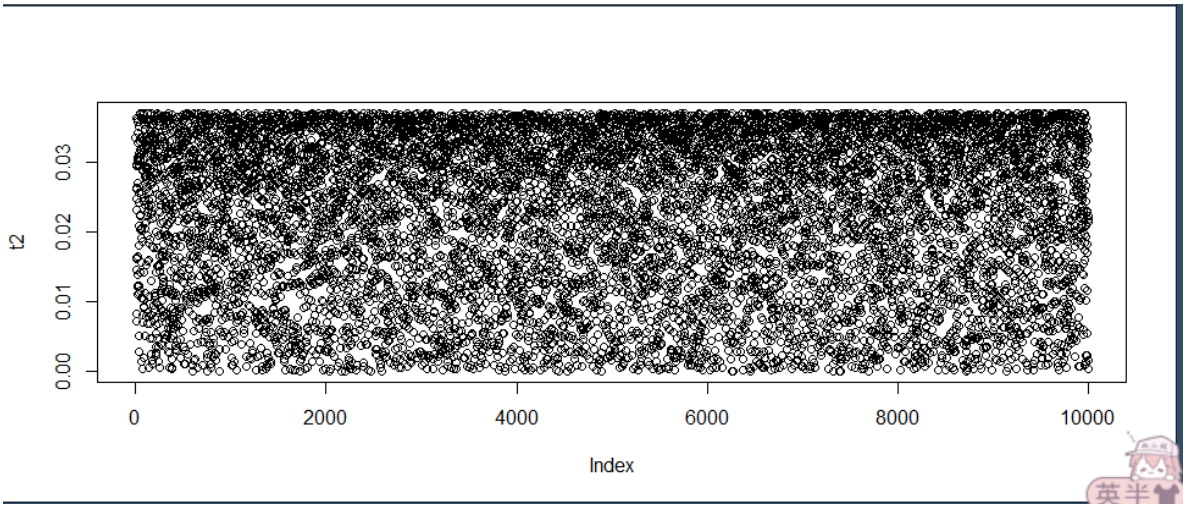
给定参数a，b，c求出λ较为困难，因为非常容易出现复数解得情况，观察到$0 < p_i < 1$，其数量级不会受λ_i的影响，故可以随机生成λ，进而求得参数a，b，c。

首先利用R语言对数据进行探索性分析，随机生成10000个特征方程的解，每个λ的取值在$0 \sim 10000$之间，观察$t_1 = \frac{b}{a^2}, t_2 = \frac{c}{a^3}$,H的分布。

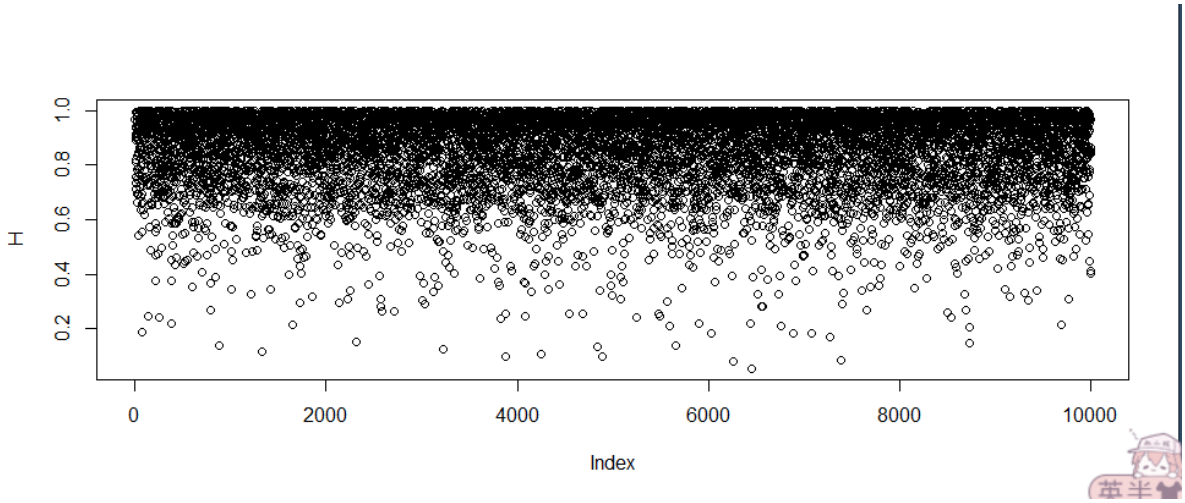
t_1分布在大约$[0, 0.35]$的区间上，且在0附近分布稀疏，在0.3附近分布密集：



t_2分布在大约$[0,\ 0.04]$的区间上，且分布比较均匀：

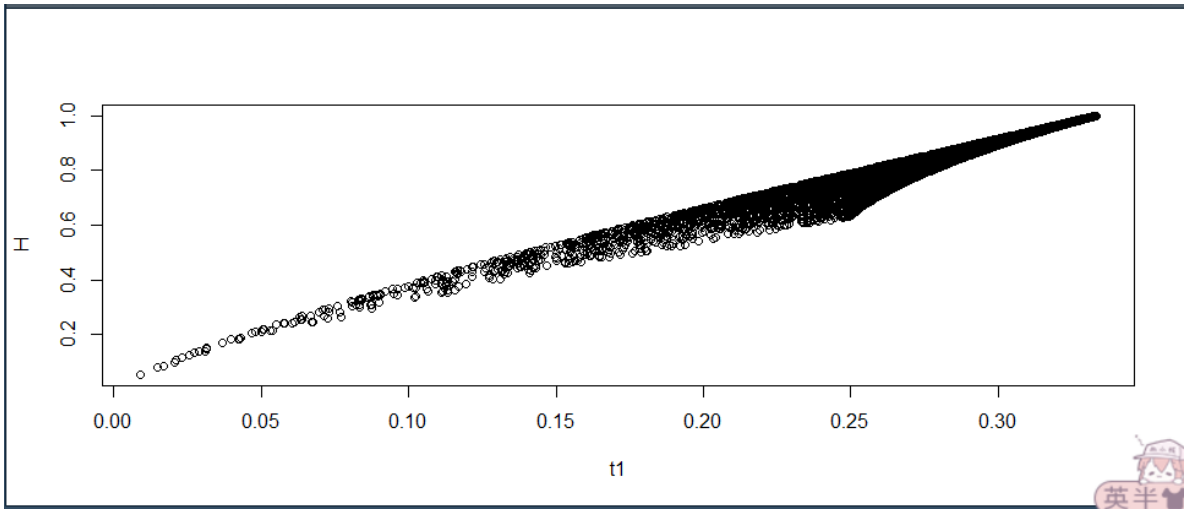


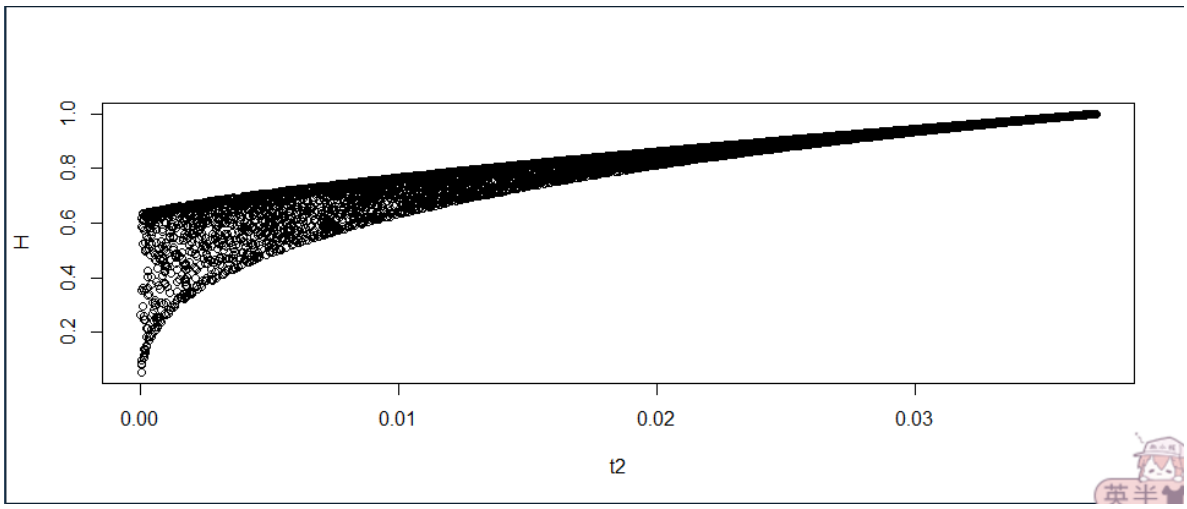
要求的H分布在区间$[0,\ 1]$之间，在靠近0的地方分布稀疏，在靠近1的地方分布密集：



我们初步发现，H与变量t_1, t_2有着相近的数量级，因此在后续拟合过程中无须做方差的归一化处理。

接下来，首先观察H与t_1, t_2是否存在着最简单的线性关系，画出H vs t_1图像和H vs t_2图像：





可以明显的看到，H与t_1, t_2确实存在着线性关系，我们可以以线性回归模型为基础，通过增添高次项、对数项等的方法来完善模型。

四、初步建模及诊断

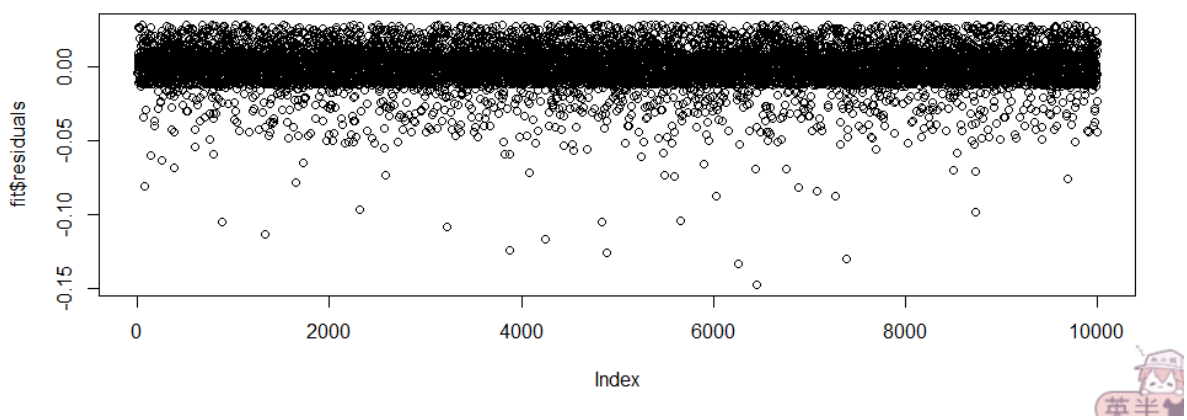
建立线性回归模型

$$H = \beta_0 + \beta_1 t_1 + \beta_2 t_2$$

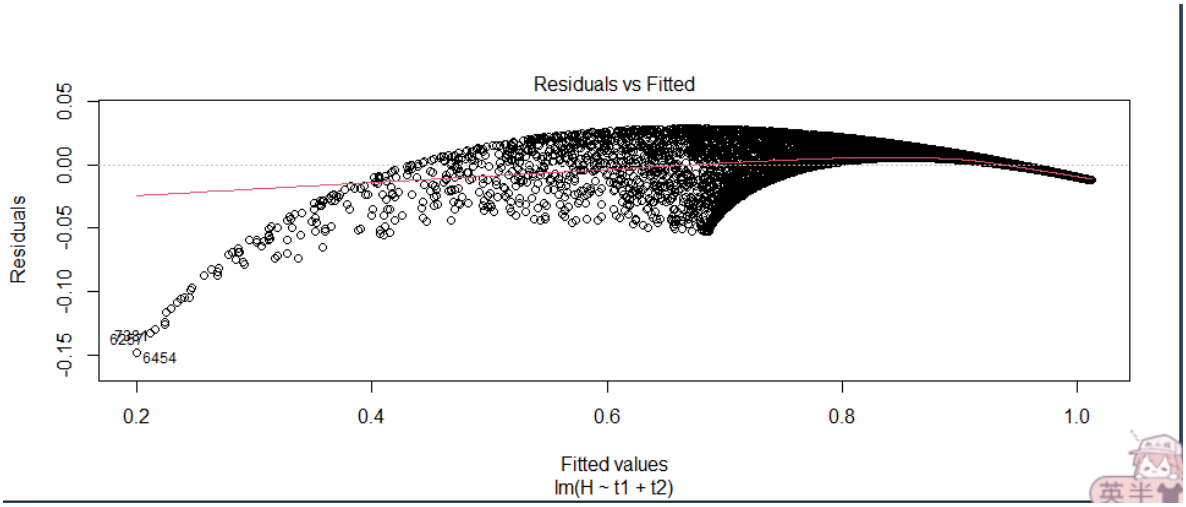
利用最小二乘得到如下估计：

$$\hat{H} = b_0 + b_1 t_1 + b_2 t_2$$

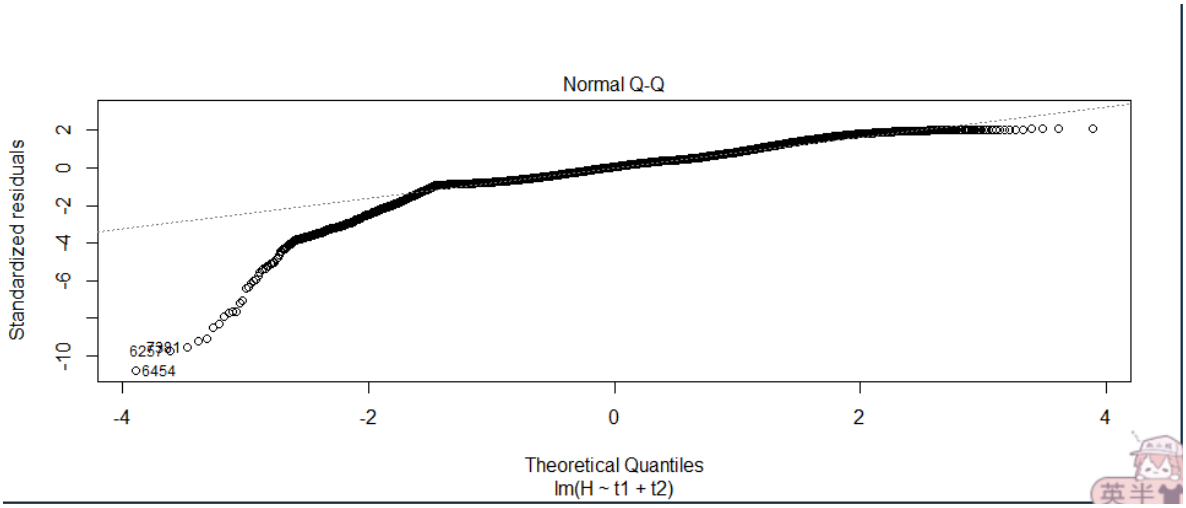
平均误差达到了0.0098，达到了0.01的标准，但是需要进一步检验残差，画出残差图，可以发现存在较多的离群值：



进一步画出residuals vs $\hat{H}$图像，可以明显观察到模型存在严重的异方差性，在H较小的时候误差很大，模型的鲁棒性较差。



可以通过正态性检验来分析模型出现异方差性的原因，画出残差的Q-Q图，可以发现模型存在较为严重的左偏和拖尾性，这时可以通过引入对数项来改善。



五、改进模型及诊断

经过多次用R语言对数据进行拟合后，决定模型中使用如下7个解释变量：

$t_1 = \frac{b}{a^2}$， $t_2 = \frac{c}{a^3}$

对数项$log(t_1), log(t_2)$

交叉项$t_1t_2, t_1log(t_1), t_2log(t_2)$

建立回归模型

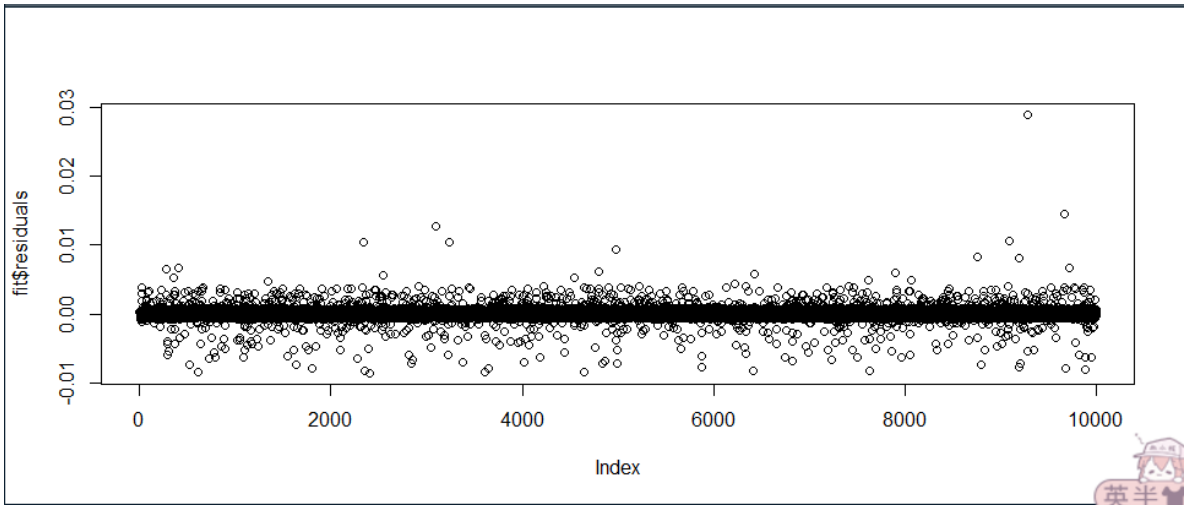
$$H = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \beta_3 log(t_1) + \beta_4 log(t_2) + \beta_5 t_1 log(t1) + \beta_6 t_2 log(t_2) + \beta_7 t_1 t_2$$

利用最小二乘得到如下估计：

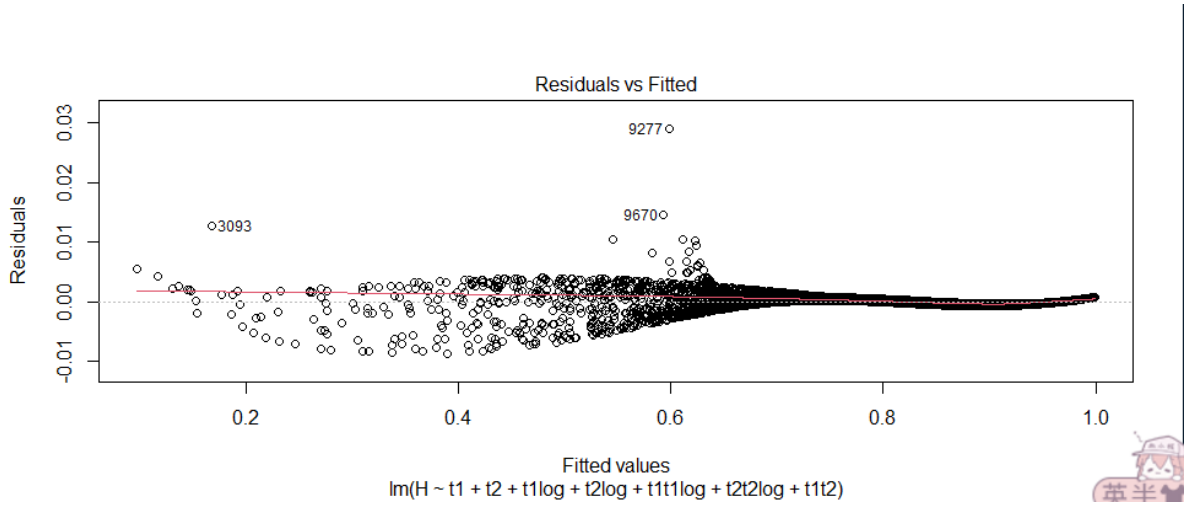
$$\hat{H} = b_0 + b_1 t_1 + b_2 t_2 + b_3 log(t_1) + b_4 log(t_2) + b_5 t_1 log(t1) + b_6 t_2 log(t_2) + b_7 t_1 t_2$$

此时平均误差可以降低至10^{-4}量级，满足要求。

做出残差图，相比于最初建立的模型，离群现象已大大改善。



画出residuals vs $\hat{H}$图像，观察到异方差性已得到很大的改善，在H较小时误差也会稳定在0.01以内，模型的稳定性很好。



所以我选择此模型为最终模型。

六、测试结果

用matlab对模型进行测试验证。

step1

首先随机生成50000个数据对模型参数$\beta=[\beta_0,\beta_1,\beta_2,\beta_3,\ \beta_4,\ \beta_5,\ \beta_6,\ \beta_7]$进行估计，得到$\beta$的估计值

$$\hat{\beta}=[0.0701,1.3264,0.3147,0.0034,0.0047,-0.7773,-2.0438,-3.2013]$$

即H的估计为：

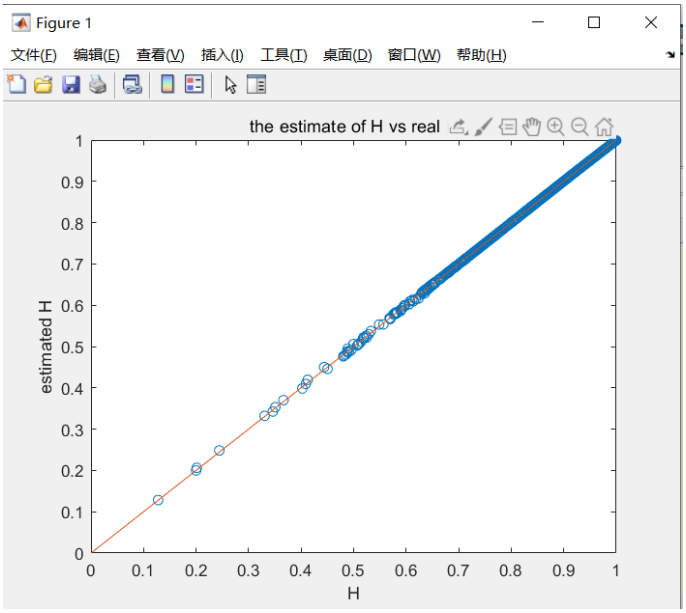
$$\hat{H}=0.0701+1.3264t_1+0.3147t_2+0.0034log(t_1)+0.0047log(t_2)-0.7773t_1log(t_1)-2.0438t_2log(t_2)-3.2013t_1t_2$$

step2

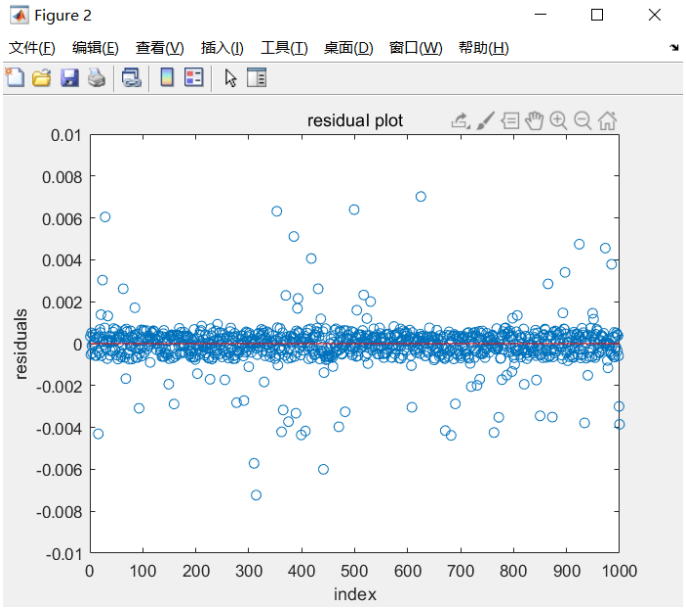
随机生成1000个测试点，得到平均误差$\overline{error}=5.496\times10^{-4}$

```
>> error

error =

   5.4958e-04
```

做出H的估计值 vs H真实值图像，点几乎都分布在直线$y=x$附近。



画出残差图，残差已很好地近似为稳定的白噪声。



step 3

作业要求，输入系数a,b,c，输出H的估计量，但由于a,b,c有较多的限制，容易出现复数解的情况，所以改为输入特征方程的解$\lambda_1, \lambda_2, \lambda_3$的解，由方程的解倒推出参数a,b,c，再由参数a,b,c得到H的估计量$\hat{H}$，并与H的真实值进行比较：

```
You should have input coefficient a, b, c
but in this problem, a,b,c have strict conditions,
your input may lead to complex solutions, which is not allowed
instead, please input lambda1,2,3. I will calculate a,b,c according to your lambdas
and I will give the estimated H calculated from a,b,c, and the real H, and the error between them
please input lambda1 which is positive:21
please input lambda2 which is positive:32
please input lambda3 which is positive:43
a =
  -96.0000

b =
   2.9510e+03

c =
  -2.8896e+04

the real H equals
    0.9634

the estimated H equals
    0.9635

the error is
    8.1409e-05
```

七、复杂度分析

如果不做任何近似，用解析解来表达H的话，复杂度非常高：

```
In[19]:= Simplify[H]
         [化简]
Out[19]= -((-2 (2 2^(1/3) a^2 - 6 2^(1/3) b - 2 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) + (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) Log[(-2 2^(1/3) a^2 + 6 2^(1/3) b + 2 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) - (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) /
           (6 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3))] + (2 2^(1/3) (1 + i Sqrt[3]) a^2 + 2^(1/3) (-6 - 6 i Sqrt[3]) b +
           4 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) + (1 - i Sqrt[3]) (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) Log[(2 2^(1/3) (1 + i Sqrt[3]) a^2 + 2^(1/3) (-6 - 6 i Sqrt[3]) b + 4 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) +
             (1 - i Sqrt[3]) (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) / (12 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3))] +
         (2 2^(1/3) (1 - i Sqrt[3]) a^2 + 6 i 2^(1/3) (i + Sqrt[3]) b + 4 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) +
           (1 + i Sqrt[3]) (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) Log[(2 2^(1/3) (1 - i Sqrt[3]) a^2 + 6 i 2^(1/3) (i + Sqrt[3]) b +
           4 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) + (1 + i Sqrt[3]) (-4 a^3 + 18 a b - 54 c + 6 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(2/3)) /
           (12 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3))]) / (12 a (-2 a^3 + 9 a b - 27 c + 3 Sqrt[3] Sqrt[-a^2 b^2 + 4 b^3 + 4 a^3 c - 18 a b c + 27 c^2])^(1/3) Log[3])

         消去  分子  分母  梯度  更多...
```

用我的模型拟合的话，复杂度非常低，只需计算出：$t_1, t_2, log(t_1), log(t_2)$

就可以进一步得到$t_1^2, t_2^2, t_1t_2, t_1log(t_1), t_2log(t_2), t_1log(t_2), log(t_1)log(t_2)$.....各种高次项和交叉项，在上述模型中选择了其中解释能力强的7个变量，实际上可以继续增添变量来提高预测能力，且复杂度不会变大。

八、总结

通过本次大作业，我体会到了“观察”的重要性。

最开始做的时候，我钻进了牛角尖，试图用a,b,c具体地表达H并用泰勒展开进行级数化简，结果做了很久都没有成功。这时我突然想到，我应该先观察一下数据的特征，当我观察到t_1, t_2, H分布范围均在$[0, 1]$之间，并通过画图发现的明显的线性相关性之后，一切都迎刃而解了，只需要利用简单的统计推断和线性回归的知识选择解释能力强的项优化模型即可。所以面对问题的时候一定要先观察，预处理和探索性分析非常重要！

我发现，学科之间是有相通性的。后来我之所以想到线性拟合，很大程度上是因为我学了统辅的课。大作业中我用了很多在统计课程中学到的知识，包括R语言分析数据、模型的方差分析、异方差处理、对数变换、正态性检验等等。这些知识帮助我系统地建立模型分析问题，考虑的方面更加全面。发现不同的课程之间的相通性时，我真的非常高兴。

总之，通过这次大作业，我对统计信号处理有了更加深刻的认识，对具体问题的分析过程得到了更多的锻炼，感谢老师和助教的耐心指导！

九、附录：源代码

matlab代码：

main.m

```
%%
clear all, close all, clc;

%% fit
%   using coefficient to fit H
%   t1 : b / a^2 (t1 > 0)
%   t2 : c / a^2 (t2 > 0)
%   t1t2 : cross term of t1 and t2
%   t1log : log(t1)
%   t2log : log(t2)
%   t1t1log : t1*log(t1)
%   t2t2log : t2*log(t2)
%   etc. Some terms may not be used.

train_num = 50000;
X = zeros(train_num, 8);
Y = zeros(train_num, 1);
for i = 1 : train_num
    lambda = rand(3, 1) * 10000;
    A = [lambda.^2, lambda, ones(3, 1)];
    B = -lambda.^3;
    t = (inv(A) * B)';
    t = [t(2)/t(1)^2, t(3)/t(1)^3];
    X(i, :) = [1, t, log(t), t.*log(t), prod(t)];
    p = lambda / sum(lambda);
    Y(i) = -p' * log(p) / log(3);
end

beta = inv(X'*X) * X' * Y;

%% test
n = 1000;
H = zeros(1, n);
Hhat = zeros(1, n);
%AX = B
%A : matrix of lambda_i lambda_i^2 1
%X : vector of a b c
%B : vector of lambda^3
for i = 1 : n
    lambda = rand(3, 1) * 10000;
    A = [lambda.^2, lambda, ones(3, 1)];
    B = -lambda.^3;
    t = (inv(A) * B)';
    t = [t(2)/t(1)^2, t(3)/t(1)^3];
    X = [1, t, log(t), t.*log(t), prod(t)];
    p = lambda / sum(lambda);
    H(i) = -p' * log(p) / log(3);
    Hhat(i) = X * beta;
end

%% plot
%the estimate of H vs real H
error = mean(abs(Hhat - H));
plot(H, Hhat, 'o');
axis([0 1 0 1]);
xlabel('H');
ylabel('estimated H');
title('the estimate of H vs real H')
hold on;
x = 0 : 0.01 : 1;
y = 0 : 0.01 : 1;
plot(x, y);

%residual plot
figure();
index = 1:n;
plot(index, Hhat-H, 'o');
axis([0 n -0.01 0.01]);
xlabel('index');
ylabel('residuals');
title('residual plot')
hold on;
plot(index, index*0, 'r');

%% input
%input : lambda1,2,3
%output1 : a b c
```

```
%output2 : real H
%output3 : estimated H
%output4 : error
disp("You should have input coefficient a, b, c");
disp("but in this problem, a,b,c have strict conditions,");
disp("your input may lead to complex solutions, which is not allowed");
disp("instead, please input lambda1,2,3. I will calculate a,b,c according to your lambdas");
disp("and I will give the estimated H calculated from a,b,c, and the real H, and the error
between them")
lambda1 = input("please input lambda1 which is positive:");
lambda2 = input("please input lambda2 which is positive:");
lambda3 = input("please input lambda3 which is positive:");
lambda = [lambda1, lambda2, lambda3]';
A = [lambda.^2, lambda, ones(3, 1)];
B = -lambda.^3;
t = (inv(A) * B)';
disp("a = "), disp(t(1)), disp("b = "), disp(t(2)), disp("c = "), disp(t(3));
t = [t(2)/t(1)^2, t(3)/t(1)^3];
X = [1, t, log(t), t.*log(t), prod(t)];
p = lambda / sum(lambda);
H = -p' * log(p) / log(3);
Hhat = X * beta;
disp("the real H equals "), disp(H);
disp("the estimated H equals "), disp(Hhat);
disp("the error is "), disp(Hhat-H);
```

R语言代码：

try.R

```
n = 10000
range = 10000
lambda1 = runif(n, 0, range)
lambda2 = runif(n, 0, range)
lambda3 = runif(n, 0, range)
A = array(0, dim = c(n, 3, 3))
X = array(0, dim = c(n, 3, 1))
B = array(0, dim = c(n, 3, 1))
Lambda = array(0, dim = c(n, 3, 1))
P = array(0, dim = c(n, 3))
H = array(0, dim = n)
for (i in 1:n) {
  A[i, , ] = c(lambda1[i]^2, lambda2[i]^2, lambda3[i]^2,
               lambda1[i], lambda2[i], lambda3[i],
               1, 1, 1)
  B[i, , ] = c(-lambda1[i]^3, -lambda2[i]^3, -lambda3[i]^3)
  X[i, , ] = solve(A[i, , ])%*%B[i, , ]
  Lambda[i, , ] = c(lambda1[i], lambda2[i], lambda3[i])
  P[i, ] = Lambda[i, , ] / (lambda1[i] + lambda2[i] + lambda3[i])
  H[i] = -(P[i, 1]*logb(P[i, 1], 3) +
           P[i, 2]*logb(P[i, 2], 3) +
           P[i, 3]*logb(P[i, 3], 3))
}

a = X[, 1, ]
b = X[, 2, ]
c = X[, 3, ]
t1 = b/a^2
t2 = c/a^3
t1t2 = t1*t2
t1log = log(t1)
t2log = log(t2)
t1t1log = t1*t1log
t2t2log = t2*t2log
t1t2log = t1*t2log
t2t1log = t2*t1log
t1logt1log = t1log*t1log
t2logt2log = t2log*t2log
t1logt2log = t1log*t2log
fit = lm(H~t1+t2+t1log+t2log+t1t1log+t2t2log+t1t2)

error = mean(abs(fit$residuals))

plot(fit)
```

十、附录：文件清单

统计信号处理第二次大作业报告.pdf	大作业报告
main.m	matlab入口程序