



信息隐藏实验报告

实验(一)：伪随机数的生成

姓 名：_____

学 号：_____

同组成员：_____

专 业：_____

二〇二二年十月

第一部分 实验分工与完成情况

1.1 实验分工与完成情况

如下表所示，其中 1-1~3-1 为必做项，3-2~3-3 为可选项，4 为探究项。

(表 1 实验分工与完成情况)

任务点	内容	完成情况	主要贡献者	贡献率
1-1	产生符合高斯分布 $N(\mu, \sigma^2)$ 的随机数	√		50%
1-2	用参数估计法估计相应的 μ 和 σ	√		
1-3	比较高斯分布的理论 CDF 和实验 CDF	√		
2-1	利用介绍的逆函数法根据 $[0, 1]$ 的均匀分布来产生尺度参数为 β 的指数分布，并且估计参数 β 的取值	√		
2-2	比较指数分布的理论 CDF 和实验 CDF	√		
2-3	推导一下 β 的最大似然率参数估计法	√		
3-1	根据课件介绍的方法产生 GGD 分布的随机数，形状参数为 $c = 1.0$ 和 $c = 0.5$	√		50%
3-2	(可选) 用矩估计法看看产生的随机数的形状参数	√		
3-3	(可选) 比较 GGD 分布的理论 CDF 和实验 CDF	√		
4	(探究) 实现估计 GGD 形状参数 c 的其他方法	√		

第二部分 相关内容介绍-背景知识

2.1 随机数及其生成

随机数是专门的随机试验的结果。

在**统计学领域**中，许多技术都需要使用到随机数。例如：在从统计总体中抽取有代表性的样本的时候，或者在将实验动物分配到不同的试验组的过程中，或者在进行蒙特卡罗模拟法计算的时候等等。

在**密码学领域**中，保密通信中大量运用的会话密钥的生成即需要真随机数的参与。如果一个随机数生成算法是有缺陷的，那么会话密钥可以直接被推算出来。若果真发生这种事故，那么任何加密算法都失去了意义。

在**信息隐藏**中，伪随机数可以保证不可见性和保密性，保证随机嵌入选择的像素位不发生冲突。

产生随机数有多种不同的方法。这些方法被称为随机数生成器。随机数最重要的特性是它在产生时后面的那个数与前面的那个数毫无关系。

在实际应用中，往往使用**伪随机数**。这些数列是“似乎”随机的数，实际上它们是通过一个固定的、可以重复的计算方法产生的。计算机或计算器产生的随机数有很长的周期性。它们不真正地随机，因为它们实际上是可以计算出来的，但是它们具有类似于随机数的统计特征。这样的发生器叫做伪随机数发生器。

2.2 常见分布函数

(1) 高斯分布

高斯分布随机分布中最常见的一种，又称为正态分布。正态曲线呈钟型，两头低，中间高，左右对称因其曲线呈钟形，因此人们又经常称之为钟形曲线。若随机变量 X 服从一个数学期望为 μ 、方差为 σ^2 的正态分布，记为 $N(\mu, \sigma^2)$ 。其概率密度函数为正态分布的期望值 μ 决定了其位置，其标准差 σ 决定了分布的幅度。当 $\mu = 0, \sigma = 1$ 时的正态分布是标准正态分布。

(2) 指数分布

指数分布（也称为负指数分布）是描述泊松过程中的事件之间的时间的概率分布，即事件以恒定平均速率连续且独立地发生的过程。这是伽马分布的一个特殊情况。它是几何分布的连续模拟，它具有无记忆的关键性质。除了用于分析泊松过程外，还可以在其他各种环境中找到。

(3) 广义高斯分布

广义高斯分布 (GGD) 被经常使用与图像/视频信号的统计分析，其形状参数常被用为图像的特征进行分类或回归。如在图像质量评价任务中，Anish Mittal 等人提出的 BRISQUE 模型利用 GGD 拟合归一化后图像 (MSCN) 的分布，利用 GGD 参数作为一组特征。

2.3 常见参数估计方法

(1) 极大似然估计

极大似然估计方法 (MLE) 也称为最大似似估计或最大似然估计，是建立在极大似然原理的基础上的一个统计方法，是概率论在统计学中的应用。极大似然估计提供了一种给定观察数据来评估模型参数的方法，即：“模型已定，参数未知”。通过若干次试验，观察其结果，利用试验结果得到某个参数值能够使样本出现的概率为最大，则称为极大似然估计。

(2) 矩估计

矩估计，即矩估计法，也称“矩法估计”，就是利用样本矩来估计总体中相应的参数。首先推导涉及相关参数的总体矩（即所考虑的随机变量的幂的期望值）的方程。然后取出一个样本并从这个样本估计总体矩。接着使用样本矩取代（未知的）总体矩，解出感兴趣的参数。从而得到那些参数的估计。

第三部分 实验目的、内容和原理

3.1 产生符合高斯分布 $N(\mu, \sigma^2)$ 的随机数

3.1.1 实验目的

通过实验，掌握于 Box-Muller 方法，通过给定的服从均匀分布的随机数，生成符合高斯分布的随机数；了解和掌握参数估计方法，使用该方法 μ, σ 的值。同时，通过 Origin 比较理论 CDF 和实验 CDF。

3.1.2 实验内容

- (1) 利用 Box-Muller 方法，产生符合高斯分布 $N(\mu, \sigma^2)$ 的随机数；
- (2) 用参数估计法估计相应的 μ 和 σ ；
- (3) 比较高斯分布的理论 CDF 和实验 CDF。

3.1.3 实验原理

- (1) 利用 Box-Muller 方法，产生高斯分布随机数：

Box-Muller，一般是要得到服从正态分布的随机数，基本思想是先得到服从均匀分布的随机数再将服从均匀分布的随机数转变为服从正态分布。

需要选取两个服从 0-1 均匀分布的随机数，两个随机数分别为 U_1, U_2 ；通过 Box-Muller 变换公式，使得 X, Y 满足：

$$\begin{aligned} X &= \cos(2\pi U_1) \sqrt{-2\ln U_2} \\ Y &= \sin(2\pi U_1) \sqrt{-2\ln U_2} \end{aligned} \quad (1)$$

通过(1)式，得到服从高斯分布 $N(0, 1)$ 的随机数 X, Y 。

$$X' = \mu + \sigma X \quad (2)$$

通过(2)式，得到服从高斯分布 $N(\mu, \sigma^2)$ 的随机数 X', Y' 。

- (2) 用参数估计法估计相应的 μ 和 σ ：

设给定的一组样本为： X_1, X_2, \dots, X_n 。

$$\begin{aligned} \mu &= E(X) \approx \frac{\sum_{i=1}^N X_i}{N} \\ \sigma &= \sqrt{\frac{\sum_{i=1}^N (X_i - \mu)^2}{N}} \end{aligned} \quad (3)$$

对于高斯分布的似然估计，使用(3)式，求得参数估计结果。

$$\begin{aligned} \mu &= E(X) \approx \frac{X_1 + X_2 + \dots + X_n}{N} \\ \mu^2 + \sigma^2 &= E(X)^2 \approx \frac{X_1^2 + X_2^2 + \dots + X_n^2}{N} \end{aligned} \quad (4)$$

对于高斯分布的矩估计，使用(4)式，求得参数估计结果。

3.2 产生符合指数分布的随机数

3.2.1 实验目的

通过实验，通过反函数法生成尺度参数为 β 的指数分布随机数的方法；了解和掌握参数估计方法，使用该方法估计 β 的值。同时，通过 Origin 比较理论 CDF 和实验 CDF。

3.2.2 实验内容

(1) 利用介绍的逆函数法根据 $[0, 1]$ 的均匀分布来产生尺度参数为 β 的指数分布，并且估计参数 β 的取值；

(2) 比较指数分布的理论 CDF 和实验 CDF；

(3) 推导一下 β 的最大似然率参数估计法。

3.2.3 实验原理

首先，根据产生随机变量的逆变换法；根据定理：设 $F(x)$ 是任一连续的分佈函数，如果 $\mu \sim U(0,1)$ [均匀分布] 且 $\eta \sim F(x)$ 。

$$P(\eta \leq x) = P(F^{-1}(\mu) \leq x) = P(\mu \leq F(x)) = F(x) \quad (5)$$

证明：由于 $\mu \sim U(0,1)$ ，则有式子(5)；所以： $\eta \sim F(x)$ 。

$$x = -\beta \ln(\mu) \quad (6)$$

通过逆变换法产生指数分布随机数：首先产生均匀分布的随机数 $\mu \sim U(0,1)$ ；然后由逆变换公式(6)计算 $x = -\beta \ln(\mu)$ 。则 $x \sim E(\beta)$ 。

3.2.4 任务点：推导一下 β 的最大似然率参数估计法

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta}, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (7)$$

我们知道，指数分布的总体密度函数如式(7)所示。

$$L(\beta) = \prod_{i=1}^n f(x_i) = \begin{cases} \frac{1}{\beta^n} e^{-\frac{1}{\beta} \sum_{i=1}^n x_i}, & x_1 > 0, \dots, x_n > 0 \\ 0, & \text{其余} \end{cases} \quad (8)$$

其似然函数如式(8)所示。

$$\frac{1}{\beta^n} e^{-\frac{1}{\beta} \sum_{i=1}^n x_i} > 0 \quad (9)$$

由于式(8)，要使 $L(\beta)$ 最大，只需使得不等式左式达到最大。

$$\frac{d \ln \left(\frac{1}{\beta^n} e^{-\frac{1}{\beta} \sum_{i=1}^n x_i} \right)}{d\beta} = n\beta - \sum_{i=1}^n x_i = 0 \quad (10)$$

由式(9)，得到的最大似然估计。（式(11)）

$$\beta' = \frac{\sum_{i=1}^n x_i}{n} \quad (11)$$

3.3 产生符合 GGD 分布的随机数

3.3.1 实验目的

通过实验，通过课件介绍的方法生成符合 GGD 分布的随机数。同时，通过矩估计法看看产生的随机数的形状参数，通过 Origin 比较理论 CDF 和实验 CDF。

3.3.2 实验内容

- (1) 根据课件介绍的方法产生 GGD 分布的随机数，形状参数为 $c=1.0$ 和 $c=0.5$;
- (2) (可选) 用矩估计法看看产生的随机数的形状参数;
- (3) (可选) 比较 GGD 分布的理论 CDF 和实验 CDF。

3.3.3 实验原理

GGD 分布的估计标准偏差如式(12)：

$$\sigma^2 = E(X^2) \approx \frac{\sum_{i=1}^N X_i^2}{N} \quad (12)$$

由(13)可计算出形状参数 c 。

$$E(|X|) = \frac{2A}{c\beta^2} \Gamma(2/c) = \sigma_x \frac{\Gamma(2/c)}{\sqrt{\Gamma(2/c) * \Gamma(3/c)}} = \frac{\sum_{i=1}^N |X_i|}{N} \quad (13)$$

第四部分 实验过程与参数说明

4.1 产生符合高斯分布 $N(\mu, \sigma^2)$ 的随机数

4.1.1 程序编写思想

使用 C++ 标准库中的 `uniform_real_distribution` 类，生成 0-1 均匀分布随机数。

首先，使用 Box-Muller 变换公式（式(1)）将随机数转换为符合高斯分布 $N(0, 1)$ 的随机数。

然后，使用式(2)的变换公式，转化为服从高斯分布 $N(\mu, \sigma^2)$ 的随机数。

最后，使用式(3)和式(4)对高斯分布的参数 μ, σ 进行参数估计。

4.1.2 程序实现代码

（表 2 高斯分布程序实现代码）

```
double x, y; //均匀分布[0, 1]随机数
double g_data; //高斯分布 N(μ, σ) 随机数
double z; //高斯分布 N(0, 1) 随机数
double temp;
```

```

//file << "num_sample = " << num_sample << endl;

//生成[0,1]均匀分布随机数设置
unsigned seed =
std::chrono::system_clock::now().time_since_epoch().count();
mt19937_64 generator(seed);
uniform_real_distribution<double> dist(0.0, 1.0);

//生成高斯分布  $N(\mu, \sigma)$  随机数
for (int i = 0; i < num_sample; i++) {
    x = dist(generator);
    y = dist(generator);

    //BOX-MULLER 生成高斯分布  $N(0, 1)$  随机数
    temp = 2 * pi * y;
    z = sqrt((-2) * log(x)) * cos(temp);

    //高斯分布  $N(\mu, \sigma)$  随机数
    g_data = in_ave + z * in_var;
    file << g_data << endl;

    //参数估计  $\mu$  和  $\sigma$ 
    c_ave += g_data;
    c_var += g_data * g_data;
}

c_ave /= num_sample;
c_var = sqrt(c_var / num_sample - c_ave * c_ave);

```

4.1.3 输入输出样例

输入样例，如图中的前三行所示；输出样例，如图中的后四行所示。此外，生成的随机数保存在“random_Gaussian.txt”文件中。

```

请输入参数  $\mu$ : 0.2201
请输入参数  $\sigma$ : 4
请输入生成随机数个数: 3000
期望:0.2201
估计期望:0.232993
方差:4
估计方差:3.96712

```

(图1 高斯分布输入输出样例)

```
random_Gaussian.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
-4.01134
-0.457254
-1.27858
4.34209
3.20677
0.962912
-0.678134
-2.95323
1.57737
-1.55423
```

(图 2 高斯分布数据样例)

4.2 产生符合指数分布的随机数

4.2.1 程序编写思想

使用 C++ 标准库中的 `uniform_real_distribution` 类，生成 0-1 均匀分布随机数。

然后，根据逆变换公式（式(6)），将 0-1 均匀分布随机数转换为符合尺度参数为 β 的指数分布的随机数。

4.2.2 程序实现代码

(表 3 指数分布程序实现代码)

```
double x;//均匀分布[0,1]随机数
double e_data;//指数分布  $\beta$  随机数

//生成[0,1]均匀分布随机数设置
unsigned seed =
std::chrono::system_clock::now().time_since_epoch().count();
mt19937_64 generator(seed);
uniform_real_distribution<double> dist(0.0, 1.0);

//生成指数分布  $\beta$  随机数
for (int i = 0; i < num_s; i++) {
    x = dist(generator);

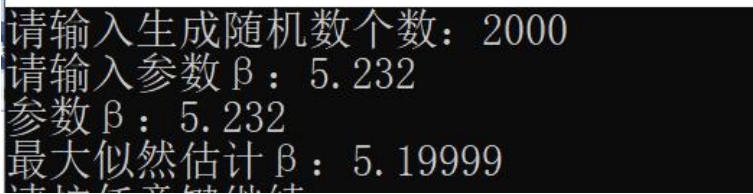
    //指数分布  $\beta$  随机数
    e_data = (-ln_beta) * log(1 - x);

    file << e_data << endl;
    c_beta += e_data;
}

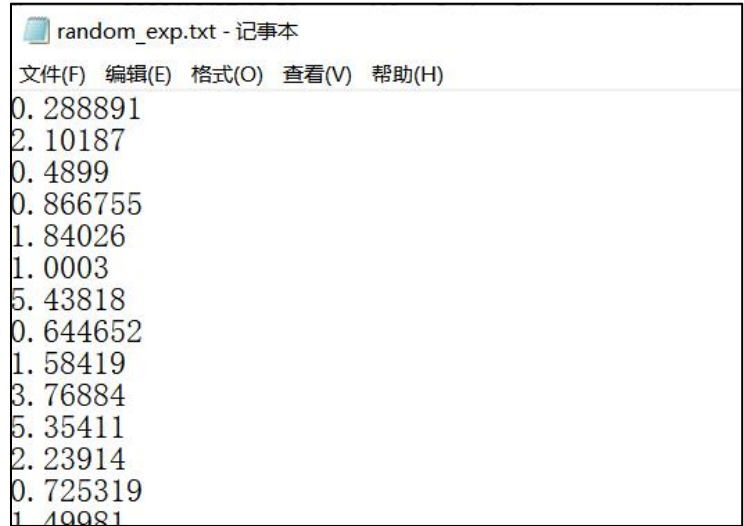
//最大似然估计参数  $\beta$ 
c_beta /= num_s;
```


4.2.3 输入输出样例

输入样例，如图中的前两行所示；输出样例，如图中的后两行所示。
此外，生成的随机数保存在“random_exp.txt”文件中。



(图 3 指数分布输入输出样例)



(图 4 指数分布数据样例)

4.3.1 程序编写思想

使用 C++ 标准库中的 `uniform_real_distribution` 类，生成 0-1 均匀分布随机数。

对于形状参数 $c=1$ ，首先生成指数分布随机数，然后，根据课件中给定的推导公式得到 $c=1$ 的 GGD 分布。

对于形状参数 $c=0.5$ ，随机生成两个服从尺度参数为 $1/\beta^{0.5}$ 的指数分布随机数，二者相加，即为所求形状参数 $c=0.5$ 的 GGD 分布随机数。

4.3.2 程序实现代码

注：本程序实现了 3 种矩估计方法和 1 种快速估计方法。在表 4 中以矩估计方法 2 为例。矩估计的方法原理基本相同，经过公式推导后得到了不同的表达式。

快速估计方法见本报告第七部分。

(表 4 GGD 分布程序实现代码)

```
if (fabs(in_c - 1.0) < 1e-6) { //c=1.0 时的 GGD 分布
    for (int i = 0; i < num_s; i++) {
        x = r_dist(generator);
        ber = i_dist(generator);
    }
}
```

```

//指数分布  $\beta$  随机数
e_data = (-in_beta) * log(1 - x);
if (ber == 1) {
    g_data = e_data;
}
else if (ber == 0) {
    g_data = -e_data;
}
data[i] = g_data;
file_detail << "x=" << x << "  e_data=" << e_data << "  ber=" << ber
<< "  g_data=" << data[i] << endl;
file_data << data[i] << endl;
//c_beta += e_data;
}
}

```

```

else if(fabs(in_c - 0.5) < 1e-6) { //c=0.5 时的 GGD 分布
    for (int i = 0; i < num_s; i++) {
        y = r_dist(generator);
        z = r_dist(generator);
        ber = i_dist(generator);

        //指数分布  $1/\beta^{0.5}$  随机数
        e_data = (-sqrt(in_beta)) * log(1 - y) + (-sqrt(in_beta)) * log(1
- z);

        //随机变量 Y 和 Z 相互独立且服从相同的指数分布
        if (ber == 1) {
            g_data = e_data * e_data;
        }
        else if (ber == 0) {
            g_data = -(e_data * e_data);
        }
        data[i] = g_data;
        file_detail << "y=" << y << "  z=" << z << "  e_data=" << e_data <<
"  ber=" << ber << "  g_data=" << data[i] << endl;
        file_data << data[i] << endl;
        //c_beta += e_data;
    }
}
}

```

//方法二：按课件推导

```

void random_ggd::moment_estimation(int*& data)
{
    double sig = 0; //计算 sigma_square 时的分子
    double test_c = 0; //使用查找匹配方法确定 c
}

```

```

double rho;
double dist = dist_max;
double c = 0;

double sigma;
double sum_abs = 0;
double res;

for (int i = 0; i < num_s; i++) {
    sig += data[i] * data[i];
}
sigma_square = sig / num_s;
sigma = sqrt(sigma_square);    //求得 sigma

for (int i = 0; i < num_s; i++) {
    sum_abs += fabs(data[i]);
}
res = sum_abs / num_s / sigma;    //求得应与 Gamma 函数分式相等的值

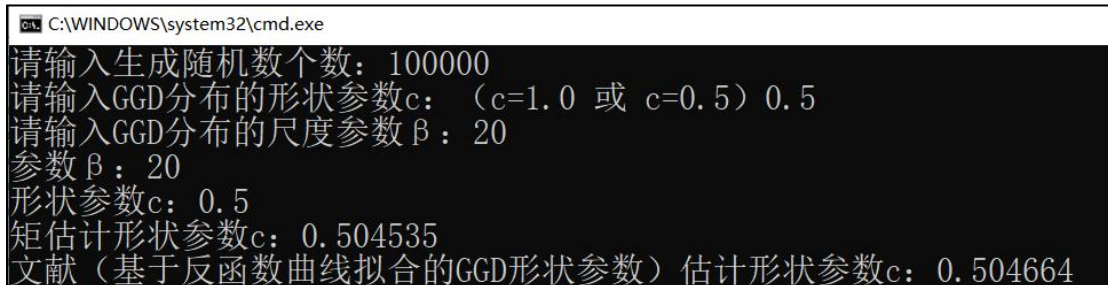
//根据等式两端相等的原则，利用查找匹配的方法确定 c
for (test_c = 0; test_c < 2.0; test_c += 0.001) {
    rho = pow(euler, gammln(2 / test_c)) / sqrt(pow(euler, gammln(1 /
test_c)) * pow(euler, gammln(3 / test_c))));
    if (fabs(rho - res) < dist) {
        dist = fabs(rho - res);
        c = test_c;
    }
}
est_c = c;
}

```

4.3.3 输入输出样例

输入样例，如图中的前三行所示；输出样例，如图中的后四行所示。

此外，生成的随机数保存在“random_ggd_data.txt”文件中，计算过程中的细节数据保存在“random_ggd_detail.txt”文件中。

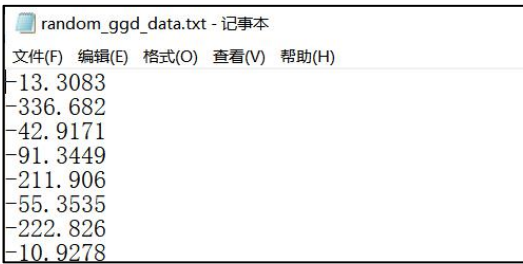


```

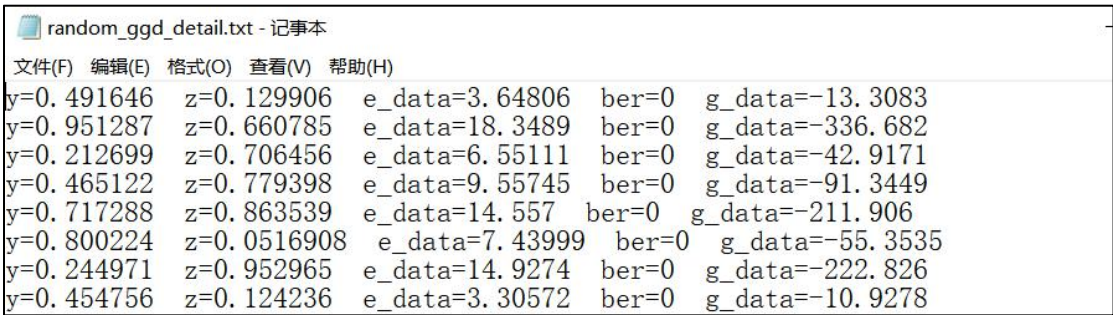
C:\WINDOWS\system32\cmd.exe
请输入生成随机数个数: 100000
请输入GGD分布的形状参数c: (c=1.0 或 c=0.5) 0.5
请输入GGD分布的尺度参数β: 20
参数β: 20
形状参数c: 0.5
矩估计形状参数c: 0.504535
文献(基于反函数曲线拟合的GGD形状参数)估计形状参数c: 0.504664

```

(图5 GGD 分布输入输出样例)



(图 6 GGD 分布数据样例)



(图 7 GGD 分布过程性数据样例)

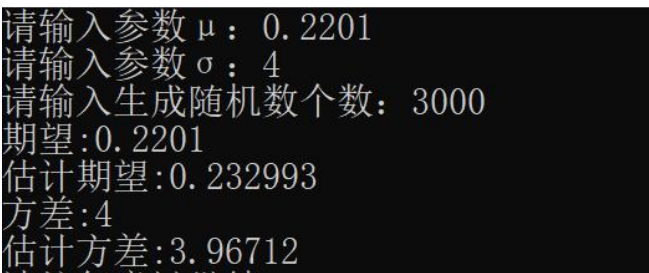
第五部分 实验结果与分析讨论

5.1 产生符合高斯分布 $N(\mu, \sigma^2)$ 的随机数

5.1.1 参数估计结果

以 $\mu=0.2201$, $\sigma=4$, 样本数 3000 为例。

在参数估计结果中, 对 μ, σ 两个参数都表现出了较好的估计结果, 呈现了较小的误差。



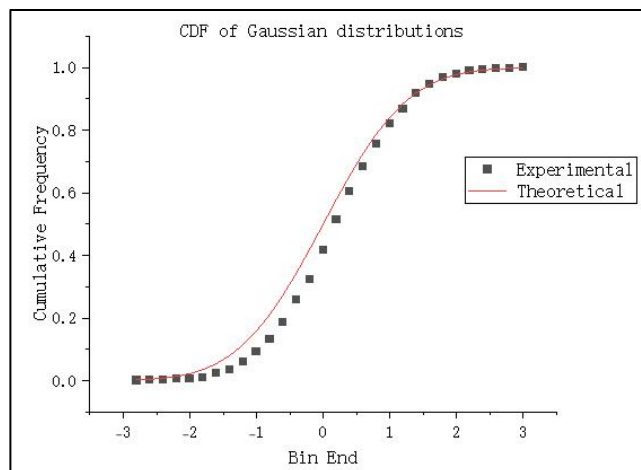
(图 8 高斯分布参数估计结果)

5.1.2 比较理论 CDF 和实验 CDF

在 Origin 软件中, 比较理论 CDF 与实验 CDF 的结果如图所示。

观察图可知, 理论与实验 CDF 的值在中间部分的误差较大。

推测这可能是由于样本数过少导致的, 实验生成的随机数较为贴合高斯分布。



(图 9 高斯分布理论 CDF 与实际 CDF 比较)

5.2.1 参数估计结果

以 $\beta = 5.232$ ，样本数 2000 为例。在估计结果中，尺度参数 β 的估计值与理论值较为接近，得到了误差较小的实验结果。

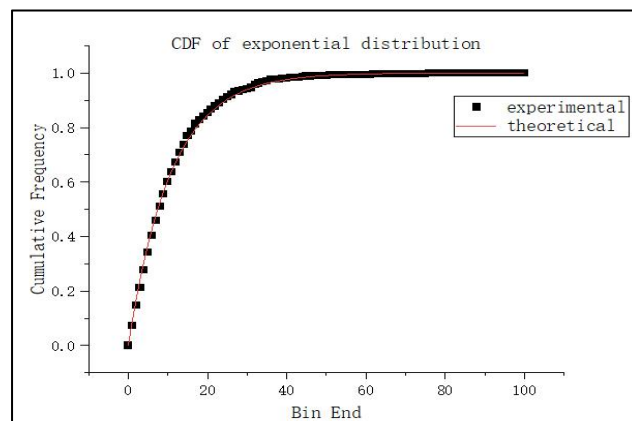
```

请输入生成随机数个数: 2000
请输入参数  $\beta$ : 5.232
参数  $\beta$ : 5.232
最大似然估计  $\beta$ : 5.19999
    
```

(图 10 指数分布参数估计结果)

5.2.2 比较理论 CDF 和实验 CDF

在 Origin 软件中，比较理论 CDF 与实验 CDF 的结果如图所示。观察图可知，理论与实验 CDF 的值整体较为相符，误差很小。



(图 11 指数分布理论 CDF 与实际 CDF 比较)

5.3.1 参数估计结果

以 $c=0.5$ ， $\beta=20$ ，样本数 100000 为例。

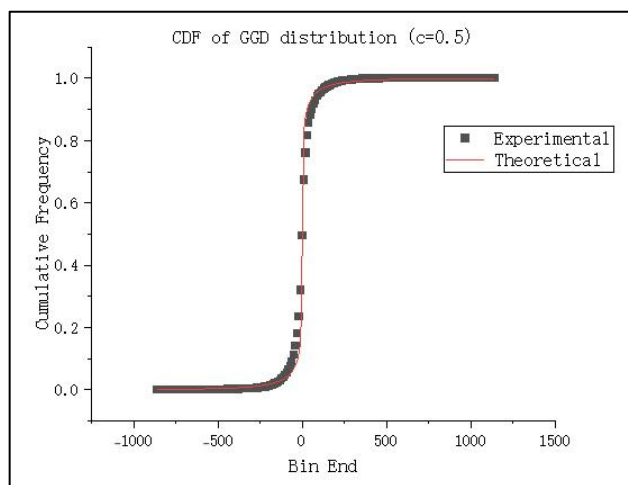
在参数矩估计结果中，对形状参数 c 表现出了较好的估计结果，呈现了较小的误差。

```
C:\WINDOWS\system32\cmd.exe
请输入生成随机数个数: 100000
请输入GGD分布的形状参数c: (c=1.0 或 c=0.5) 0.5
请输入GGD分布的尺度参数β: 20
参数β: 20
形状参数c: 0.5
矩估计形状参数c: 0.504535
文献(基于反函数曲线拟合的GGD形状参数)估计形状参数c: 0.504664
```

(图 12 GGD 分布参数估计结果)

5.3.2 比较理论 CDF 和实验 CDF

在 Origin 软件中, 比较理论 CDF 与实验 CDF 的结果如图所示。
观察图可知, 理论与实验 CDF 的值整体较为相符, 误差很小。



(图 13 GGD 分布理论 CDF 与实际 CDF 比较)

第六部分 实验结论

本次实验圆满完成了 100%的所有任务、100%的所有选做任务, 并对 GGD 形状参数快速估计算法进行了实验探究, 均呈现了较好的结果。

这次实验对课堂所学有了进一步的理解, 也有利于后续实验的顺利开展。

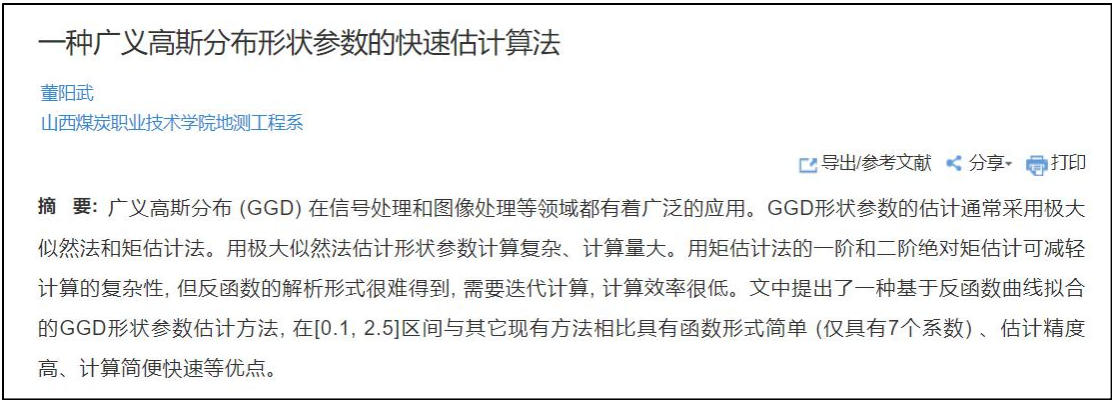
第七部分 实验探究: 复现一种 GGD 形状参数快速估计算法

本次实验复现了论文[7]中提出的一种 GGD 形状参数快速估计算法。

在实验代码中以

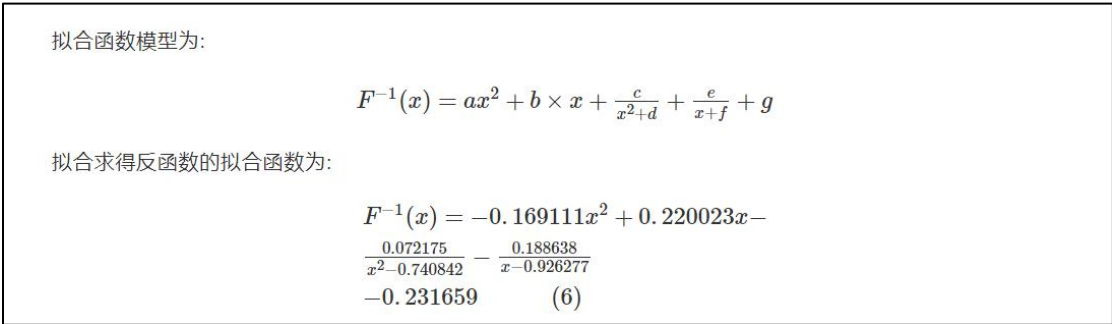
```
void moment_estimation_quick(double*& data);
double moment_estimation_quick_func(double x);
```

两个函数呈现。



(图 14 复现的 GGD 参数估计算法)

该算法使用反函数拟合的方法, 在较快的时间代价中得到了比较精准的估计结果。



(图 15 复现的 GGD 参数估计算法)

第八部分 参考文献

[1]徐柏军, 岳春国, 徐正军. 伪随机数实现及变换方法研究[J]. 科学技术与工程, 2007, 7 (11): 2472-2475. DOI:10. 3969/j. issn. 1671-1815. 2007. 11. 004.

[2]王新成, 孙宏. 高速伪随机数发生器的设计与实现. 计算机工程与应用, 2004;11:20~23

[3]赵翔, 郝林. 数字水印综述[J]. 计算机工程与设计, 2006, 27 (11): 1946-1950. DOI:10. 3969 /j. issn. 1000-7024. 2006. 11. 011.

[4]史帅.Box-Muller 变换原理详解[EB/OL]. <https://zhuanlan.zhihu.com/p/38638710>

[5]指数分布随机数[EB/OL]. <https://www.cnblogs.com/liam-ji/p/11626243.html>

[6]广义高斯分布 (GGD) 和非对称广义高斯分布 (AGGD) 的形状参数快速估计[EB/OL]. https://blog.csdn.net/sinat_36438332/article/details/88363492

[7]董阳武. 一种广义高斯分布形状参数的快速估计算法[J]. 矿山测量, 2012, (05): 45-48.