计算物理第二次作业第 6 题

姓名：姚星宇　　学号：PB21000188　　班级：2021 级少年班学院四班

2022 年 10 月 18 日

1　题目重述

对两个函数线型（Gauss 分布和类 Lorentz 型分布），令其为 p（x）其中常数 $a \neq b \neq 0$, 用舍选法对 p(x) 抽样。将计算得到的归一化频数分布直方图与理论曲线 $p(x)$ 进行比较，讨论差异，讨论抽样效率。

$$Gaussian :\sim exp(-ax^2); Lorentzianlike :\sim \frac{1}{1+bx^4} \tag{1}$$

注：这里对于原题中“设其一为 $p(x)$，另一为 $F(x)$”不能理解，故将两个分布均看作密度分布进行解决。

2　题目分析

本作业使用 Python（Anaconda 3）进行代码编写。

对于这类边缘与中心大小相差较大的连续函数，可使用阶梯函数进行舍选法抽样。考虑到 Gaussian 函数与 Lorentzian like 函数均减小较快，故仅对 $[-5, 5]$ 间进行抽样也具有足够的精度。

在抽样过程中，先将抽样区间分为 k 份 (k 大约为十几)，然后定步长搜索找到每个区间的最大值，将这个最大值作为该区间的阶梯函数的大小。然后对于阶梯函数积分（即将每个区域的最大值求和），得到与区间编号相对应的、阶梯函数的分布的数组：

$$bmax = [0, \max_{[0,a]}(f(x)), \max_{[0,a]}(f(x)) + \max_{[a,2a]}(f(x)), \cdots, \sum_i^k \max_{[ia,ia+a]} f(x)] \tag{2}$$

对该数组进行归一化得到 $b0$。生成一个 $[0, 1]$ 之间的随机数 $ran1$，在 $b0$ 中进行二分查找，得到区间编号 i 使得 $b0[i] < ran1 < b0[i+1]$。然后在将 i 线性转换到这一区间 $i' = \frac{ran1-b[i]}{b[i+1]-b[i]} * (bmax[i+1] - bmax[i]) + bmax[i]$。

随后取得 $[0, 1]$ 之间的随机数 $ran2$，然后进行比较：如果 $ran2(b[i+1] - b[i]) < f(i')$, 则为有效点，将 x 加入抽样结果；反之，则舍去 x。

得到大量的 x 后，将 x 保留两位小数并借助 dictionary 容器进行频率统计，最后画出频数分布直方图并于理论曲线相比较。

3　代码

```python
    # -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import math as m

def sch_random(N, M = 1, a = 16807, b = 0, m = 2**31 - 1, seed = 1):#N 为生成个数，M 为生成间隔
```

```
    q, r = m // a, m % a        #得到p, r
    for i in range(N):
        for j in range(M):
            seed = a * (seed % q) -r * (seed // q) #进行schrage方法
            if seed < 0:
                seed += m
        yield seed/m#单位化schrage


def fun_sel_sam(function,f_random,a,b,k=5,N=100000,seed1 = 114514,seed2 =
    1919810):#抽样函数
    result = []
    temp = 0
    bmax = np.zeros((k+1),dtype = 'double')
    b0 = np.zeros((k),dtype = 'double')
    const = (b-a)/k/100
    for i in range(k):
        temp = 0
        for j in range(100):
            if temp < function((i*100+j+0.5)*const+a):
                temp = function((i*100+j+0.5)*const+a)
        b0[i] = temp
        bmax[i+1] = temp
        bmax[i+1] += bmax[i]
    bmax = bmax/bmax[-1]#产生阶梯分布函数


    brandom = f_random(N,seed = seed1)
    for i in f_random(N,seed = seed2):
        temp1,temp2 = 0,k
        while temp2 - temp1>1 :
            temp = (temp1+temp2)//2
            if bmax[temp]>i:
                temp2 = temp
            else:
                temp1 = temp
        #二分查找得到位置
        b_rand = next(brandom)*b0[temp1]
```

```
            i = ((i-bmax[temp1])/(bmax[temp2]-bmax[temp1])+temp1)*(b-a)/k+a
            if function(i)>b_rand:#判断是否保留
                result+=[i]
    return np.array(result)


def freq(c):#频数统计
    d = {}
    for i in c:
        if round(i,2) in d:
            d[round(i,2)] += 1
        else:
            d[round(i,2)] = 1
    x , y = np.zeros((len(d))),np.zeros((len(d)))
    j = 0
    for i in sorted(d.items(),key = lambda x:x[0]):
        x[j] = i[0]
        y[j] = i[1]
        j += 1
    return x,y,len(c)#返回长度计算抽样效率


if __name__ == '__main__':
    fig , ax = plt.subplots(1, 2, figsize = (8,2))

    x,y,eff= freq(fun_sel_sam(lambda x:m.exp(-2*x**2), sch_random, a=-5, b=5,k=10,N=100000))
    z = np.exp(-2*x**2)#产生标准曲线
    y = y/y.max()*1.1
    ax[0].plot(x, y, color = 'r', linestyle = '-', marker = 'None')
    ax[0].plot(x, z, color = 'b', linestyle = '-', marker = 'None')
    print('抽样效率:'+str(eff/100000))


    x,y,eff = freq(fun_sel_sam(lambda x:1/(2*x**4+1), sch_random, a=-5, b=5, k=10,N=100000))
    z = 1/(2*x**4+1)#产生标准曲线
    y = y/y.max()*1.1
    ax[1].plot(x, y, color = 'r', linestyle = '-', marker = 'None')
```

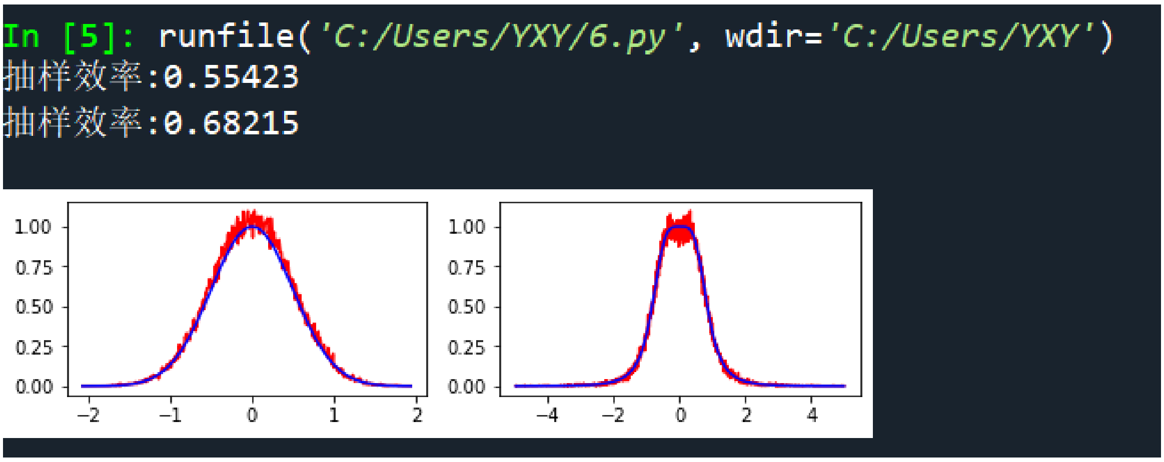

图 1: 程序运行结果截图

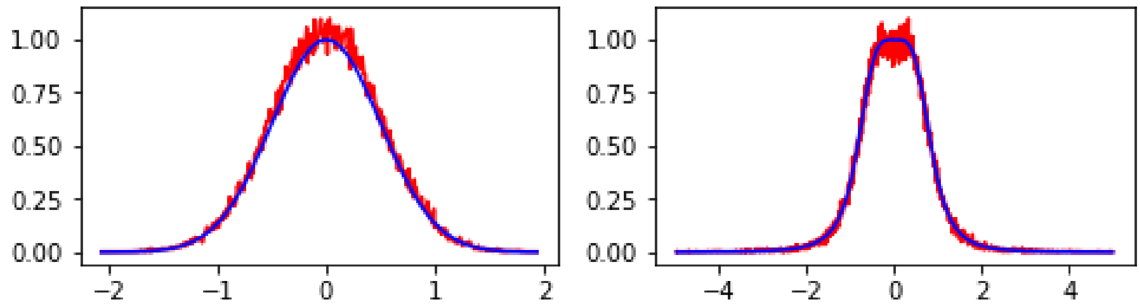

图 2: 得到的抽样图像与原始数据的对比

```
    ax[1].plot(x, z, color = 'b', linestyle = '-', marker = 'None')
    print('抽样效率:'+str(eff/100000))
```

4 结果

如图一所示，程序在 Anaconda3 环境下可以正常运行。

如图二所示，两图像基本重合，可认为抽样具有较好的符合度。

舍选法的抽样效率为 8%，较差，在这类数据量较小的情况下效率不如直接抽样。

参考文献

[1] 丁泽军. 计算物理讲义 [M]