姓名: 孙秋实 **Homework - 5#**

学号: 10185501402

Course: 概率论

布置日期：2020.04.08 截止日期：2020.04.14 提交日期：2020.04.14

Problem 1

计算问题. 此处讨论的问题是计算单位正方形中的子集 S 的面积的方法. 我们利用单位正方形上服从均匀分布的一串随机的点列. 如果第 i 个点是在集合 S 中，令 X = 1，否则为 0 现在设 X1,⋯,Xn 是这样生成的随机变量序列，记

$$S_n = \frac{X_1 + \cdots + X_n}{n}$$

(a) 证明 E[S] 等于子集 S 的面积，而 $var(S_n)$ 当 n 无限增加时趋于 0

(b) 证明为了计算 Sn 的值，我们可以利用 S_{n-1} 和 X_n 的值，而并不依赖于以前的 $S_1, S_2, \cdots S_{n-1}$ 写出一个公式

(c) 利用计算机的随机数发生器写一个计算机程序，产生数列 Sn ,n = 1, 2, ⋯, 10000 其中 S 是单位正方形的内切圆怎样利用你的程序去近似 π 的值?

(d) 利用类似的计算机程序去近似地计算单位正方形内由条件 $0 \leqslant \cos \pi x + \sin \pi y \leqslant 1$ 所确定的点集的面积

Solution:

(a) 首先每个点是否再子集 S 中都是独立的

先考虑取一点列的期望 $\mathbf{E}[S_n] = \mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}[X_i] = \mathbf{E}[X_i] = S$

再考虑子集 S 的面积，因为是单位正方形，所以点可取位置的总面积为 1，而因随机变量 X 服从均匀分布，$P(X_i = 1) = S/1 = S$

因为每次取点都是独立的，所以方差满足线性性:

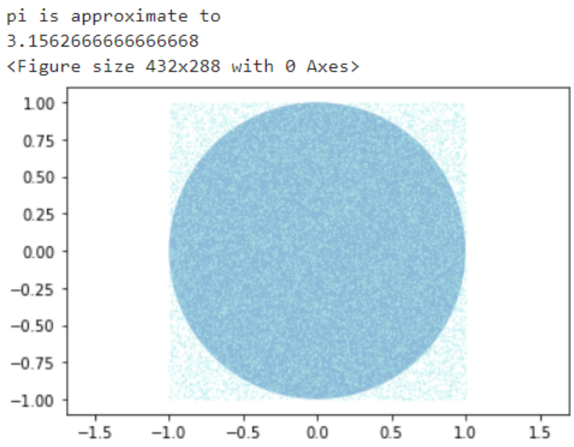
$Var(S_n) = Var(\frac{1}{n}\Sigma_{i=1}^{n} X_i) = \frac{1}{n^2} \times n \times Var(X_i) = \frac{1}{n} Var(X_i)$, 可见当 n 趋于无穷时 $Var(S_n)$ 趋于 0

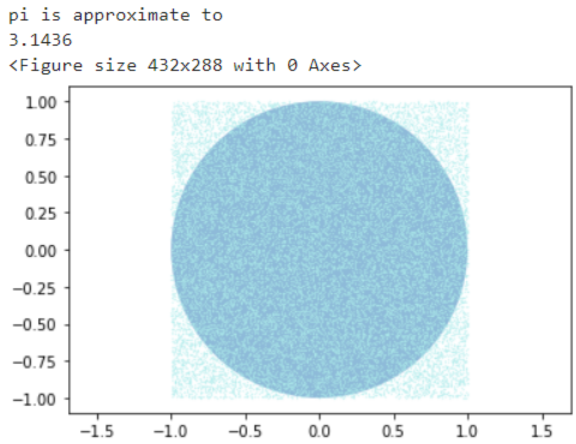
(b) 导出一个递推关系，每次取点相互独立，只与子集大小有关

$S_n = \frac{n-1}{n} \times S_{n-1} + \frac{1}{n} \times X_n$

(c) 使用 Monte Carlo 方法利用内切圆近似求解 pi，其核心思想是利用以下数学关系:

$$\frac{Area(Circle)}{Area(Square)} = \frac{\pi r^2}{4r^2}$$



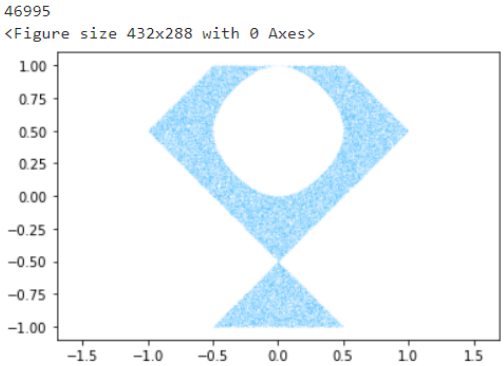


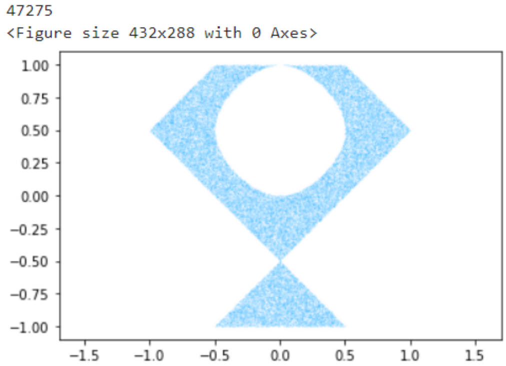
上述两张图分别是随机选取 15,000 个点和 20,000 个点的试验结果

```python
from matplotlib.patches import Circle
import numpy as np
import matplotlib.pyplot as plt
plt.figure()
n=15000 #试验次数，理论上次数越多越精确
r=1.0;a,b=(0.,0.) #取圆心和半径
left,right=a-r,a+r #约束边界条件
upper,lower=b-r,b+r
x=np.random.uniform(left,right,n) #调用均匀分布开始制作点列
y=np.random.uniform(upper,lower,n)
#计算随机点到圆心的距离
d=np.sqrt((x-a)**2+(y-b)**2)
#统计落在单位圆的随机点数目
count=sum(np.where(d<r,1,0))
pi=4*count/n #计算近似圆周率
print('pi is approximate to')
print( pi )
#往画布上添加两个图形，即点列和圆
fig=plt.figure()
axes=fig.add_subplot(1,1,1)
axes.plot(x,y,'ro',label = "Monte Carlo",color='paleturquoise',markersize=0.2) #选了个喜欢的颜色
plt.axis('equal') #防止图形在JUPYTER-LAB中变形
C1=Circle(xy=(a,b),radius=r,alpha=0.5)
axes.add_patch(C1)
plt.show()
#注意要把点的大小调的小一点，不然会挡住圆看不清效果
```

(d) 同样，使用 Monte Carlo 方法，模拟了 150,000 个随机生成的点，实验两次取均值

$Area = \frac{(46695+47275)/2}{150{,}000} \approx 0.314$



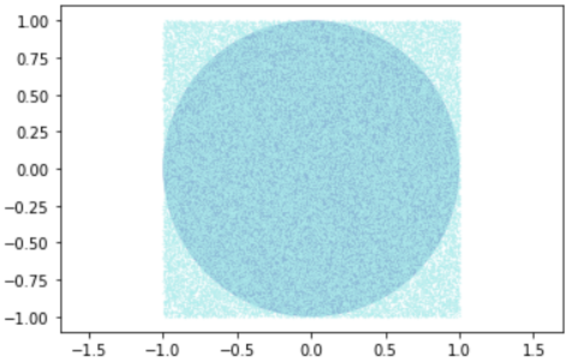


具体代码如下所示，与 (c) 小问不同的是，因为打印所有点视觉效果不好，所以这里新建了两个列表用来存储符合条件的点坐标，图中只显示了满足条件 $0 \leqslant \cos \pi x + \sin \pi y \leqslant 1$ 的点

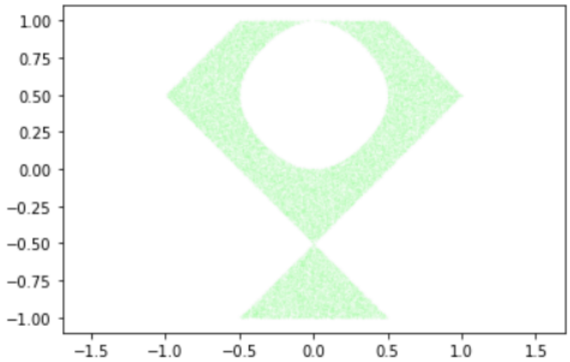
```python
import numpy as np
import matplotlib.pyplot as plt
from math import *
list1=[]
list2=[]
plt.figure()
count=0
n=150000
left,right=-1,1 #约束边界条件
upper,lower=1,-1
x=np.random.uniform(left,right,n) #调用均匀分布开始制作点列
y=np.random.uniform(upper,lower,n)
for i in range (0,n-1):
    if(0<=np.cos(pi*x[i])+np.sin(pi*y[i])<=1):
        list1.append(x[i])
        list2.append(y[i])
        count+=1
print(count)
fig=plt.figure()
axes=fig.add_subplot(1,1,1)
axes.plot(list1,list2,'ro',label = "Monte Carlo",color='deepskyblue',markersize=0.03) #又选了个喜欢的颜色
plt.axis('equal') #防止图形在JUPYTER-LAB中变形
plt.show()
```

附录: 在 Problem3(c)(d) 使用随机种子而不是直接调用均匀分布，从结果上看没有很大差异

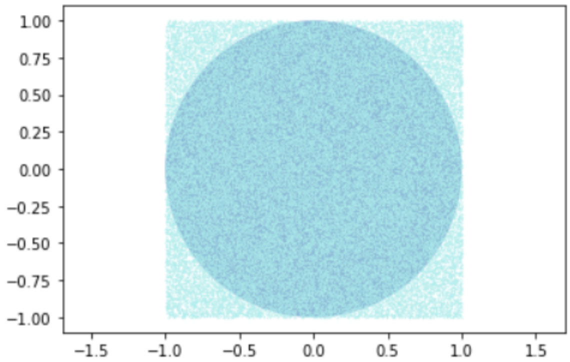
trial 50000
pi is approximate to
3.1352
<Figure size 432x288 with 0 Axes>



47342 in 150000
<Figure size 432x288 with 0 Axes>



trial 50000
pi is approximate to
3.15784
<Figure size 432x288 with 0 Axes>



47353 in 150000
<Figure size 432x288 with 0 Axes>