第十二周作业：蒙特卡罗方法

1. 炮弹射击的目标为一半径 100m 的圆形区域，弹着点服从以目标为中心的二元正态分布,X 和 Y 方向上的标准差分别为 80m 和 50m,相关系数为 0.4。用 Monte Carlo 方法求炮弹命中圆形区域的概率，并与利用数值积分方法得到的结果进行比较。

解 1：利用 Monte-Carlo 直接模拟

利用 matlab 函数 **mvnrnd** 生成大量二元正态分布的点(x,y)，用落在圆内的频率估计概率。

取点次数 n=1e5，最终 P=0.6997.

解 2：利用 Monte-Carlo 计算区域内的二元积分

这里需要注意，随机点(x,y)的两个坐标独立同分布，服从(-R,R)上的均匀分布。

被积概率密度函数

$$p(x,y)=\frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}}e^{-\frac{1}{2(1-\rho^2)}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}-\frac{2\rho xy}{\sigma_x\sigma_y}\right)}$$

我们所求的概率

$$P=\iint_S p(x,y)dxdy\approx\frac{S}{n}\Sigma_{\mathrm{i=1\sim n}}p(x_i,y_i)$$

其中 S 为边长为 2R 的正方形的面积，$S=4R^2$.

取 n=1e5，求得 P=0.6965.

解 3：利用数值积分方法求二元积分

利用 matlab 函数 **integral2** 求得圆内的 P=0.6979.发现此时所能达到的相对误差最小为 1e-4，不能更小，这也从侧面说明 MC 在数值积分问题上的重要作用。

2. 考察不同规模与不同稀疏度的 0-1 矩阵，对 GG 与 KKLLL 两种计算矩阵积和式的 Monte Carlo 方法进行适当的比较。

解：

注：稀疏度为 0~1 之间的小数，表示每一个矩阵元素取 0 的概率，稀疏度数值越大表示矩阵越稀疏。

以矩阵阶数=10 为例：

从计算时间的角度，Ryser-NW 法 (精确算法，计算复杂度为 $O(n^2 \cdot 2^{n-1})$)大约 1ms

左右，GG 法大约在 3ms 左右，而 KKLLL 法在 10ms 左右。明显 KKLLL 法相较于 GG 法需要更大的计算量。

相对误差的角度，除了

全 1 矩阵 GG 法相对误差 2.07%，KKLLL 法相对误差 10.57%

以外，KKLLL 法（平均相对误差 2%左右）均比 GG 法（平均相对误差 3%左右）误差更小。当稀疏度=1、0.9、0.7 时由于矩阵过于稀疏，积和式=0，此时没有相对误差数据。

以矩阵阶数=15 为例：

从计算时间的角度，Ryser-NW 法大约 30ms 左右，GG 法大约在 5ms 左右，而 KKLLL 法在 16ms 左右。在矩阵阶数较高时，明显 Ryser-NW 法相较于 KKLLL 法、GG 法需要更大的计算量，而 KKLLL 又比 GG 计算量大。

相对误差的角度，除了

全 1 矩阵 GG 法相对误差 7.83%，KKLLL 法相对误差 10.30%；

稀疏度 0.1 的矩阵 GG 相对误差 2.62%，KKLLL 相对误差 3.04%；

稀疏度 0.7 矩阵 GG 相对误差 1.99%，KKLLL 相对误差 10.39%

之外，KKLLL 法（平均相对误差 7%左右）均比 GG 法（平均相对误差 10%左右）误差更小。当稀疏度=1、0.9、0.8 时由于矩阵过于稀疏，积和式=0，此时没有相对误差数据。

以矩阵阶数=20 为例：

从计算时间的角度，Ryser-NW 法大约 1.1s 左右，GG 法大约在 10ms 左右，而 KKLLL 法在 26ms 左右。在矩阵阶数较高时，明显 Ryser-NW 法相较于 KKLLL 法、GG 法需要更大的计算量，而 KKLLL 又比 GG 计算量大。

相对误差的角度，除了

全 1 矩阵 GG 法相对误差 0.53%，KKLLL 法相对误差 5.59%；

稀疏度 0.6 的矩阵 GG 相对误差 4.74%，KKLLL 相对误差 10.08%

之外，KKLLL 法（平均相对误差 8%左右）均比 GG 法（平均相对误差 15%左右）误差更小。当稀疏度=1、0.9 时由于矩阵过于稀疏，积和式=0，此时没有相

对误差数据。

小结：

1. Ryser-NW 法虽然能够得到精确解，但是计算复杂度为 $O(n^2 \cdot 2^{n-1})$指数级，在矩阵阶数较大时计算量很大。
2. GG 法操作相对简单，计算量小，KKLLL 法计算量稍大，在矩阵不是非常密集或非常稀疏时，计算精度 KKLLL 一般优于 GG。
3. 随着矩阵阶数的上升，GG 法和 KKLLL 法的相对误差也上升，并且出现了个别情况下的 20~30%相对误差，说明算法的稳定性不是很好，对于一些特殊的矩阵计算效果不好。
4. 计算矩阵积和式的建议是，尝试多种方法，在计算时间、计算精度、算法稳定性等角度考虑，选取性能均衡的方法。