

Monte Carlo 方法及其应用

曲双石 王会娟

摘要:本文首先介绍了 Monte Carlo 方法的产生和发展。在第二部分给出了 Monte Carlo 方法对定积分进行近似计算的两种方法:随机投点法和样本平均值法,其中对样本平均值法,在借助于产生服从标准指数分布和标准正态分布随机数的基础上,将其推广到被积区间为无穷的情况分别加以讨论。上述两种方法分别给与实例,利用 Mat lab 编程进行了求解和误差分析。第三部分本文进一步用 Monte Carlo 方法解决 n 维定积分问题,进行了误差分析,并给出实例。第四部分介绍了 Monte Carlo 方法在计算概率时的应用,对两个具体的例子用此方法在 Mat lab 帮助下求解,并通过与用公式计算的精确结果,比较说明 Monte Carlo 方法在计算概率问题中的有效性。

关键词: Monte Carlo 方法;随机投点法;样本平均值法;Mat lab 编程;概率计算

Monte Carlo Method and its Applications

Qv Shuangshi Wang Huijuan

Abstract: First, this paper introduces the initiation and development of Monte Carlo method. In the second part, two methods of Monte Carlo are given to simulate computing integral: pelting randomly method and sample average method. Particularly, in terms of sample average method, with the random numbers generating from Exp (0,1) and N (0,1), we extend usual integral to the unlimited integral area. We also give some examples on using Mat lab to compute and analyze the error. In the third part, we continue to talk about how to solve the integral of n determines by Monte Carlo, and analyze the error and give some examples. In the forth part, we introduce Monte Carlo method's application in computing probability. Also to two particular examples, we use Monte Carlo method and Mat lab to compute them. Comparing to the exact results, we can see the efficiency in using Monte Carlo to solve the probability problems.

Key words: Monte Carlo method; pelting randomly method; sample average method; Mat lab program; computing probability

1 引言

Monte Carlo 方法,又名随机模拟法(stochastic simulation)或统计实验法。它是以概率统计理论为基础,依据大数定律(样本均值替代总体均值),利用电子计算机数字模拟技术,解决一些很难直接用数学运算求解或用其他方法不能解决的复杂问题的一种近似算法。

Monte Carlo 方法能够比较逼真地描述复杂事物的特点,如象物理化学等方面的实验过程等。加上飞

速发展的电子计算机技术,对于以往那些只能进行定性而很难进行定量研究的问题, Monte Carlo 方法可以大显身手,应用日趋广泛。

2 Monte Carlo 方法的产生与发展

2.1 Monte Carlo 方法的产生

Monte Carlo 方法是在二战期间产生和发展起来的。他的奠基者是美籍匈牙利人数学学家冯诺伊曼(J. Von Neumann 1903-1957)。由于通常计算量相当大而电子计算机在当时还没有出现,所有运算只能用手工

进行,故而相当长的时间里 Monte Carlo 方法难以推广。

作为 Monte Carlo 方法的最初应用,是解决蒲丰氏问题。1777 年,法国数学家 Buffon 提出利用投针实验求解的问题。

设平面上有无数多条距离为 1 的等距平行线,现向该平面随机的投掷一根长度为 $l(l \leq 1)$ 的针。随机投针是指针的中心点于最近的平行线间的距离 x 均匀分布在 $[0, 1/2]$ 上,针与平行线的夹角 φ (不管相交与否) 均匀分布在 $[0, \pi]$ 上,如图一所示,故而,故其 $X \sim U(0, \frac{1}{2})$, $\phi \sim U(0, \pi)$ 概率密度函数分别为 $p(x) = \frac{1}{\frac{1}{2}}, p(\varphi) = \frac{1}{\pi}$ 。

故我们得到针与线相交的充要条件是 $\frac{x}{\sin \varphi} \leq \frac{1}{2}$ 。

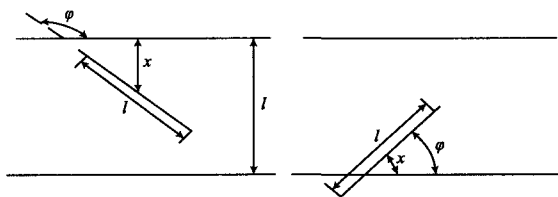


图 1 针与线相交的几种情况

则针与线相交的概率是 $P(x \leq \frac{l}{2} \sin \varphi)$

$$= \int_0^{\pi} \int_0^{\frac{l}{2} \sin \varphi} p(x) p(\varphi) dx d\varphi = \int_0^{\pi} \int_0^{\frac{l}{2} \sin \varphi} \frac{2}{\pi} dx d\varphi$$

$$= \int_0^{\pi} \frac{2}{\pi} \cdot \frac{l}{2} \sin \varphi d\varphi = \frac{2l}{\pi}$$

$$\text{所以得到圆周率 } \pi = \frac{2l}{P(x \leq \frac{l}{2} \sin \varphi)} = \frac{2l}{P}$$

假如我们能做大量的投针实验并记录下针与线的相交次数,则可以根据大数定律估计出针线相交的概率 P 。投针实验 N 次可能有 n 次使针与任意平行线相交,那么 $P \approx \frac{n}{N}$,显然,试验次数 N 越多, P 的近似程度越好。历史上曾有几位学者做过这样的投针试验,并用手工计算出 π 值,结果参见表 1。

表 1 Buffon 试验手工计算的结果

试验者	时间(年)	针长 l	投针次数	相交次数	π 估计值
Wolf	1850	0.80	5000	2532	3.15956
Smith	1855	0.60	3024	1218	3.15665
Fox	1884	0.75	1030	489	3.15951
Lazzarini	1925	0.83	3408	1808	3.14159292

应该指出,上述试验的精度一般不会很高。譬如,假设 $l=1$,则 $p = \frac{2}{\pi}$ 。由中心极限定理,如果试验次数为

N ,则 p 的估计值 \hat{p} 渐近服从 $N(p, \frac{p(1-p)}{N})$,近似为 $N(0.6366, \frac{0.2313}{N})$ 。因此,若要以 95% 的概率保证 p 精

确到三位有效数字,即 \hat{p} 与 p 的差距小于 0.001,则 N 必须满足 $N \geq 1.96^2 \times 0.2313 / 0.001^2 \approx 8.89 \times 10^5$ 。

重复进行上千次的投针实验和手工计算,要消耗大量的人力、财力。Monte Carlo 方法虽然能解决此类问题,但得不到推广应用。

2.2 Monte Carlo 方法的发展

20 世纪 40 年代以后,随着电子计算机的出现和发展,人们有可能用计算机来模拟这类实验和计算。计算机具有计算速度快和存储容量大的特点,采用数字模拟技术可以代替许多实际上非常庞大而复杂的实验,并迅速将实验结果进行运算处理,于是 Monte Carlo 方法重新被提起,引起世人重视,应用日渐广泛。实际上,采用 Monte Carlo 方法在计算机上建立模型来解决 Buffon 问题是非常简单的。

我们在计算机上进行模拟试验,给定 l ,我们可以在计算机上随机产生 x 和 φ ,然后判断 $\frac{x}{\sin \varphi} \leq \frac{l}{2}$ 是否成立。若成立,则针线相交,否则不交。假如我们在计算机上独立的产生 N 对这样的 x 和 φ ,并记录下 $\frac{x}{\sin \varphi} \leq \frac{l}{2}$ 成立的次数,记为 n 。则 π 估计值可取为 $\frac{2 \times N \times l}{n}$,这就是随机模拟的计算结果。

借助 Mat lab 在计算机上进行投针实验,(相应程序见附件)所取得的计算结果参见表 2。

表 2 表 1. Buffon 试验计算机模拟计算的结果

$N \backslash \pi$	$l=0.4$	$l=0.6$	$l=0.8$	$l=0.9$	$l=1.0$
50000	3.1252442	3.1583934	3.1392246	3.14652309	3.1464351
100000	3.1241457	3.1211798	3.1269543	3.1324504	3.1286664
200000	3.1414435	3.1396353	3.1373472	3.1366262	3.1371811
500000	3.1519392	3.1470787	3.1455443	3.1422826	3.1425492
800000	3.1483358	3.1465609	3.1440825	3.1423004	3.1430294
1000000	3.1435296	3.1459072	3.1412646	3.1419370	3.1423368
2000000	3.1497179	3.1462907	3.1418722	3.1410570	3.1407306
5000000	3.1393330	3.1398735	3.1410784	3.1416836	3.1412106

不难看出,利用计算机运算不仅结果精确,而且迅速。

随着电子计算机的普及, Monte Carlo 方法作为一种独到的方法得到开发,并首先应用到核武器的试验与研制中。尤其是各种可是编程方法的不断涌现,更显示出 Monte Carlo 方法的最独到的优点,即形象直观

地用数学方法在电子计算机上实现数字模拟实验。

3 Monte Carlo 方法在一重定积分计算中的应用

考虑一个简单的定积分

$$\theta = \int_a^b f(x) dx \quad (1)$$

不少统计问题,如计算概率、各阶矩等,最后都归结为定积分的近似计算问题,我们首先介绍两种求的简单的 Monte Carlo 方法,并给出求积分的几个实例。

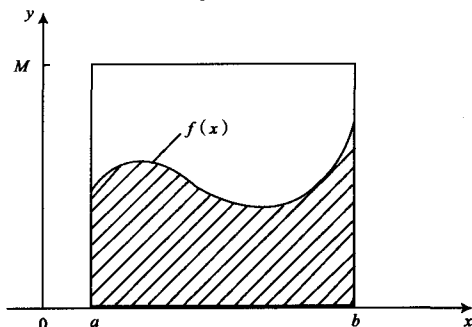
3.1 随机投点法

考虑(1)的积分。简单起见,设 a, b 有限, $y=f(x)$ 在 $[a, b]$ 上连续非负,有 $0 \leq f(x) \leq M$, 其中 $M = \max_{a \leq x \leq b} f(x)$ 。实际上,如果对于函数 $f(x)$,不是在区间 $[a, b]$ 上都有 $f(x)$

≥ 0 ,则进行计算时可以利用恒等式 $\int_a^b f(x) dx$

$= \int_a^b [f(x)+h]dx - h(b-a)$, 其中 $h>0$ 是适当选择的正数,

使对一切 $x: a \leq x \leq b$, 都有 $f(x)+h \geq 0$ 。



令 $\Omega = \{(x, y): a \leq x \leq b, 0 \leq y \leq M\}$, 并设 (X, Y) 是在 Ω 上均匀分布的二维随机变量, 联合密度函数为

$\frac{1}{M(b-a)} I_{a \leq x \leq b, 0 \leq y \leq M}$, 则易见 $\theta = \int_a^b f(x) dx$ 是 Ω 中曲线 $y=f(x)$ 下方的面积(如上图所示)。

随机投点法的思想是:向 Ω 中进行随机投点。若点落在 $y=f(x)$ 下方称为中的, 否则成为不中, 则点中的概率为 $p = \frac{\theta}{M(b-a)}$, 若进行了 N 次投点, 其中 n 次中

的, 则得到 θ 的一个估计 $\hat{\theta}_1 = M(b-a) \frac{n}{N}$ 。

其实施步骤为:

1) 独立产生 $2N$ 个 $U(0,1)$ 的随机数 $u_i, v_i, i=1, 2, \dots, N$;

2) 计算 $x_i = a + u_i(b-a), y_i = Mv_i$ 和 $f(x_i)$;

3) 统计 $f(x_i) \geq y_i$ 的个数 n ;

4) 计算 θ 估计值 $\hat{\theta}_1 = M(b-a) \frac{n}{N}$ 。

[例 1] 用 Monte Carlo 方法计算积分 $\int_0^1 e^x dx$ 。

解: 用公式(1), $a=0, b=1, M = \max_{a \leq x \leq b} e^x = e$

我们取 $N=20$, 利用 Mat lab 编程计算所求积分值, u, v 是我们分别利用均匀分布随机数产生器 unifrnd 产生的服从均匀分布 $U(0,1)$ 的两个独立的 20×1 列向量, 令 $x=u, y=Mv=e \cdot v$, 可得计算结果参见表 3, 具体程序详见附录。

表 3 随机投点法计算积分 $\int_0^1 e^x dx$

i	u_i	v_i	$x_i = u_i$	$y_i = e \cdot v_i$	$f(x_i) = e^{x_i}$
1	0.5979	0.0149	0.5979	0.0404	1.8183
2	0.9492	0.2882	0.9492	0.7834	2.5835
3	0.2888	0.8167	0.2888	2.2201	1.3348
4	0.8888	0.9855	0.8888	2.6788	2.4323
5	0.1016	0.0174	0.1016	0.0472	1.1069
6	0.0653	0.8194	0.0653	2.2273	1.0675
7	0.2343	0.6211	0.2343	1.6884	1.264
8	0.9331	0.5602	0.9331	1.5228	2.5424
9	0.0631	0.244	0.0631	0.6633	1.0652
10	0.2642	0.822	0.2642	2.2344	1.3024
11	0.9995	0.2632	0.9995	0.7155	2.717
12	0.212	0.7536	0.212	2.0486	1.2361
13	0.4984	0.6596	0.4984	1.7931	1.6461
14	0.2905	0.2141	0.2905	0.5819	1.3371
15	0.6728	0.6021	0.6728	1.6367	1.9596
16	0.958	0.6049	0.958	1.6444	2.6065
17	0.7666	0.6595	0.7666	1.7927	2.1523
18	0.6661	0.1834	0.6661	0.4984	1.9467
19	0.1309	0.6365	0.1309	1.7303	1.1399
20	0.0954	0.1703	0.0954	0.4629	1.1001

由上表可见, 满足条件 $y_i \leq f(x_i)$ 的 y_i 个数为 12, 故而得到所求积分的估计值 $\hat{\theta}_1 = M(b-a) \frac{n}{N} = \frac{12 \times e}{20} = 1.631$ 。

我们可以计算此积分的精确值 $\theta = \int_0^1 e^x dx = e - 1 \approx 1.718$, 故有此方法求得的绝对误差是 $|1.718 - 1.631| = 0.087$, 相对误差 $\delta = \frac{0.087}{1.631} \times 100\% = 5.33\%$

实际上, 因为每次投点结果服从二点分布, 即

$n \sim b(N, p)$, 其中 $p = \frac{\theta}{M(b-a)}$,

故 $E[n] = Np = \frac{N\theta}{M(b-a)}, \text{Var}(n) = Npq = Np(1-p) =$

$\frac{N\theta[M(b-a) - \theta]}{M^2(b-a)^2}$ 。

不难看出:

$$E[\hat{\theta}_1] = \frac{M(b-a)}{N} E(n) = \theta.$$

$$\begin{aligned} Var(\hat{\theta}_1) &= Var(M(b-a) \frac{n}{N}) = \frac{M^2(b-a)^2}{N^2} Var(n) \\ &= \frac{\theta}{N} [M(b-a) - \theta] \end{aligned}$$

若用估计的标准差来衡量精度, 则估计值 $\hat{\theta}_1$ 的精度阶为 $n^{-1/2}$.

[例 2] 用 Monte Carlo 方法求积分值 $\int_0^{\pi} \frac{\sin x}{x} dx$

解: 用公式(1), 其中 $a=0, b=\pi, M=\max_{0 \leq x \leq \pi} \frac{\sin x}{x}=1$ 取 $N=20$, 利用 Mat lab 编程可得到结果参见表 4, 具体程序详见附录。

表 4 随机投点法计算积分 $\int_0^{\pi} \frac{\sin x}{x} dx$

i	u_i	v_i	$x_i = \pi \cdot u_i$	$y_i = v_i$	$f(x_i) = \frac{\sin x_i}{x_i}$
1	0.43	0.097	1.3508	0.097	0.7225
2	0.7245	0.4621	2.2762	0.4621	0.3345
3	0.7936	0.0794	2.4931	0.0794	0.2423
4	0.7646	0.5884	2.4022	0.5884	0.2805
5	0.2437	0.2424	0.7655	0.2424	0.9052
6	0.2515	0.1901	0.79	0.1901	0.8992
7	0.432	0.9963	1.357	0.9963	0.7201
8	0.6743	0.2815	2.1184	0.2815	0.403
9	0.6793	0.7726	2.1342	0.7726	0.3961
10	0.1876	0.8845	0.5894	0.8845	0.9431
11	0.312	0.8263	0.9801	0.8263	0.8474
12	0.6445	0.3384	2.0249	0.3384	0.4438
13	0.6484	0.6551	2.0371	0.6551	0.4385
14	0.4741	0.1506	1.4893	0.1506	0.6692
15	0.2044	0.4797	0.642	0.4797	0.9327
16	0.8003	0.9587	2.5143	0.9587	0.2334
17	0.8147	0.0047	2.5594	0.0047	0.2148
18	0.1629	0.9339	0.5119	0.9339	0.9569
19	0.6488	0.699	2.0384	0.699	0.4379
20	0.5917	0.9826	1.859	0.9826	0.5157

则满足条件 $y_i \leq f(x_i)$ 的 y_i 个数 $n=12$, 利用公式(1)得到, 所求积分的估计值为 $\hat{\theta} = M(b-a) \frac{n}{N} = \pi \cdot \frac{12}{20} = 1.885$.

由数学分析的知识可知, 例 2 中的被积函数 $\frac{\sin x}{x}$

原函数是不能用初等函数表示的, 因而 $\int_0^{\pi} \frac{\sin x}{x} dx$ 不能用牛顿-莱布尼兹公式计算, 而用 Monte Carlo 方法便可求得近似值, 并且利用 Mat lab 我们可以多次

运行所编辑的求值程序, 求得积分值的多个近似值, 再取平均值以求得近似程度很高的结果。

3.2 样本平均值法

随机投点法中要求积分区间有界, 并且被积函数有界, 一般的, 这两个限制条件并不一定满足, 此时我们就要用样本平均值法。

样本平均值法的基本原理是:

考虑积分 $\theta = \int_a^b f(x) dx$, 设 $g(x)$ 是 (a, b) 上的一个

密度函数, 将积分改写为 $\theta = \int_a^b \frac{f(x)}{g(x)} g(x) dx$

$= E\left[\frac{f(x)}{g(x)}\right]$, 可见任意一个积分均可以表示为某个随机变量(函数)的期望。根据矩估计法, 若有 n 个来自 $g(x)$ 的观测值, 则可给出 θ 的一个矩估计。

(1) 被积区间上下限均为有限数

首先考虑最简单的 a, b 均为有限数时, 可取 $g(x) = \frac{1}{b-a}$ 。设 x_1, x_2, \dots, x_n 是来自 $U(a, b)$ 的随机数, 则 θ 的一个估计为

$$\hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{g(x_i)} = \frac{b-a}{n} \sum_{i=1}^n f(x_i) \quad (2)$$

具体步骤为:

- 1) 独立的产生 n 个 $U(0, 1)$ 随机数 u_1, u_2, \dots, u_n ,
- 2) 计算和 $x_i = a + (b-a)u_i$ 和 $f(x_i), i=1, 2, \dots, n$,

- 3) 计算 θ 的估计值 $\hat{\theta}_2 = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$ 。

[例 3] 再看例 1.1, 我们用样本平均值法估计 $\int_0^1 e^x dx$ 。

解: 利用公式(2), $a=0, b=1$, 令 $n=20$, 由 Mat lab 编程可得计算结果参见表 5, 其中 u 是由均匀分布随机数产生器 unifrnd 随机产生的 20×1 的列向量, 程序详见附录。

表 5 样本平均值法计算积分 $\int_0^1 e^x dx$

i	u_i	$x_i = u_i$	$f(x_i) = e^{x_i}$	i	u_i	$x_i = u_i$	$f(x_i) = e^{x_i}$
1	0.8952	0.8952	2.4478	11	0.3584	0.3584	1.4311
2	0.9424	0.9424	2.5661	12	0.2853	0.2853	1.3301
3	0.3351	0.3351	1.3981	13	0.8686	0.8686	2.3837
4	0.4374	0.4374	1.5486	14	0.6264	0.6264	1.8709
5	0.4712	0.4712	1.6018	15	0.2412	0.2412	1.2727
6	0.1493	0.1493	1.161	16	0.9781	0.9781	2.6593
7	0.1359	0.1359	1.1455	17	0.6405	0.6405	1.8974
8	0.5325	0.5325	1.7032	18	0.2298	0.2298	1.2584
9	0.7258	0.7258	2.0664	19	0.6813	0.6813	1.9765
10	0.3987	0.3987	1.4899	20	0.6658	0.6658	1.9461

由此得到所求积分值的估计为 $\hat{\theta}_2 = \frac{b-a}{n} \sum_{i=1}^n f(x_i) = 1.7577$ 。

该估计的绝对误差为 $|1.718-1.7577|=0.0397$,
相对误差 $\delta=\frac{0.0397}{1.7577}\times 100\%=2.26\%$ 可见, 样本平均值

法比随机投点法更有效, 即 $Var(\hat{\theta}_2)\leq Var(\hat{\theta}_1)$ 。

我们可以分析 $\hat{\theta}_2$ 的精度:

$$\begin{aligned} E[\hat{\theta}_2] &= E\left[\frac{b-a}{n} \sum_{i=1}^n f(x_i)\right] = \frac{b-a}{n} \sum_{i=1}^n E[f(x_i)] = \theta \\ Var(\hat{\theta}_2) &= Var\left[E\left[\frac{f(x)}{g(x)}\right]\right] = \frac{1}{n} \left\{ E\left[\frac{f(x)}{g(x)}\right]^2 - \theta^2 \right\} \\ &= \frac{1}{n} \{ E[(b-a)^2 f^2(x)] - \theta^2 \} \\ &= \frac{1}{n} \left[(b-a)^2 \cdot \int_a^b f^2(x) \frac{1}{b-a} dx - \theta^2 \right] \\ &= \frac{1}{n} \left[(b-a) \int_a^b f^2(x) dx - \theta^2 \right] \end{aligned}$$

在假设 $0 \leq f(x) \leq M$ 下, 可以证明, 当 $N=n$ 时,

$$\begin{aligned} Var(\hat{\theta}_1) - Var(\hat{\theta}_2) &= \frac{\theta}{M} [M(b-a) - \theta] \\ &- \frac{1}{n} \left[(b-a) \int_a^b f^2(x) dx - \theta^2 \right] = \frac{M(b-a)}{n} \left[\theta - \int_a^b \frac{f^2(x)}{M} dx \right] \\ &\geq \frac{M(b-a)}{n} \left[\theta - \int_a^b f^2(x) dx \right] = 0 \end{aligned}$$

而且, 只要 $\{x: f(x) < M\}$ 不是零测度集, 就有 $Var(\hat{\theta}_2) \leq Var(\hat{\theta}_1)$ 。由此, 样本平均值法比随机投点法更有效。

[例4]用 Monte Carlo 方法计算 $\int_0^{0.2} e^{-x^2} dx$

解: 利用公式(2-2), $a=0, b=0.2$, 取 $n=20$, 由 Matlab 编程可得到结果参见表6, 程序详见附录。

表6 样本平均值法计算 $\int_0^{0.2} e^{-x^2} dx$

i	u_i	$x_i=u_i$	$f(x_i)=e^{-x_i^2}$	i	u_i	$x_i=u_i$	$f(x_i)=e^{-x_i^2}$
1	0.0483	0.0097	0.9999	11	0.9898	0.198	0.9616
2	0.4604	0.0921	0.9916	12	0.1524	0.0305	0.9991
3	0.8	0.16	0.9747	13	0.2033	0.0407	0.9983
4	0.2894	0.0579	0.9967	14	0.8193	0.1639	0.9735
5	0.6951	0.139	0.9809	15	0.0584	0.0117	0.9999
6	0.2593	0.0519	0.9973	16	0.5385	0.1077	0.9885
7	0.7132	0.1426	0.9799	17	0.1902	0.038	0.9986
8	0.7204	0.1441	0.9795	18	0.5995	0.1199	0.9857
9	0.7333	0.1467	0.9787	19	0.2923	0.0585	0.9966
10	0.6223	0.1245	0.9846	20	0.0913	0.0183	0.9997

由此得到所求积分值的估计为 $\hat{\theta}_2 = \frac{b-a}{n} \sum_{i=1}^n e^{-x_i^2} = 0.1977$ 。

注意到被积函数 e^{-x^2} 的原函数仍然不能用初等函

数表示, 因而 $\int_0^{0.2} e^{-x^2} dx$ 不能用牛顿-莱布尼兹公式计算, 用样本平均值法也求得了其较为精确的近似值。

(2)被积区间上下限有一个为 ∞

考虑 a, b 中有一个为 ∞ 时, 不妨考虑区间 $(0, \infty)$, 即所求积分为 $\theta = \int_0^{\infty} f(x) dx = \int_0^{\infty} \frac{f(x)}{g(x)} g(x) dx = E\left[\frac{f(x)}{g(x)}\right]$ 。

此时 $g(x)$ 就无法取为均匀分布的密度函数, 我们考虑定义在 $(0, \infty)$ 标准指数分布的密度函数。即取 g

$$g(x) = \begin{cases} e^{-x}, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

设 x_1, x_2, \dots, x_n 是 $Exp(1)$ 的随机数, 则由样本平均值法可得到:

$$\hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{g(x_i)} = \frac{1}{n} \sum_{i=1}^n f(x_i) e^{x_i} \quad (3)$$

如果我们能够获得来自 $Exp(1)$ 的 n 个随机数, 则利用(3)就可成功估计所求的积分值了。

因为 $U(0,1)$ 分布随机数的产生在大多数计算语言(软件)中都有现成的程序可调用, 下面我们考虑如何基于均匀分布 $U(0,1)$ 产生的 $Exp(1)$ 的随机数。

方法一:(逆变换法)

设 $X \sim p(x) = e^{-x} I_{x \geq 0}$ 则其分布函数是 $F(x) = 1 - e^{-x}$, 由逆变换法可得 $F^{-1}(y) = -\ln(1-y)$, $0 \leq y \leq 1$ 。

根据定理“随机变量 U 服从 $(0,1)$ 上的均匀分布, 则 $X = F^{-1}(u)$ 的分布函数为 $F(x)$ ”, 我们得到若有服从均匀分布的随机变量 u , 则 $x = -\ln(1-u)$ 就服从标准指数分布。由于 $1-u$ 与 u 同分布, 于是得到指数分布的抽样方法:

- 1) 由 $U(0,1)$ 抽取 u ,
- 2) 计算 $x = -\ln u$ 。

上述方法看上去简单, 但因为在计算机上计算自然对数需用到级数展开, 每得到一个指数分布随机数都要用到一次级数展开, 是较费时的, 因而效率不高。

方法二:

标准指数变量与 $U(0,1)$ 变量有紧密的联系, 我们可在下述引理的基础上产生多个独立的 $Exp(1)$ 变量。

引理: 设 $X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_{n-1}$ 是来自 $U(0,1)$ 的随机样本, $Y_{(1)} < Y_{(2)} < \dots < Y_{(n-1)}$ 是 Y_1, Y_2, \dots, Y_{n-1} 的次序统计量。记 $Y_{(0)} = 0, Y_{(n)} = 1$, 令 $Z_k = (Y_{(k-1)} - Y_{(k)}) \cdot \ln\left(\prod_{i=1}^n X_i\right)$,

$k=1, 2, \dots, n$, 则 Z_1, Z_1, \dots, Z_n 独立同分布, 服从 Exp (1)。由此我们得到下述抽样法:

1) 由 $U(0, 1)$ 独立抽取 $2n-1$ 个随机数 $u_1, u_2, \dots, u_n, u_{n+1}, \dots, u_{2n-1}$;

2) 计算 $r = \ln(\prod_{i=1}^n u_i)$;

3) 将 u_{n+1}, \dots, u_{2n-1} 按从小到大排序, 记为 $Y_{(1)} < \dots < Y_{(n-1)}$,

4) 计算 $x_k = (Y_{(k-1)} - Y_{(k)}) \cdot r$, 其中 $Y_{(0)} = 0, Y_{(n)} = 1$ 。

则 x_1, x_2, \dots, x_n 是来自 $Exp(1)$ 的随机数。

我们可将方法一、方法二进行比较。若我们要抽取 n 个 $Exp(1)$ 变量, 用方法一只要抽取 n 个 $U(0, 1)$ 变量, 但要进行 n 次自然对数的计算, 方法二, 虽然抽取 $2n-1$ 个 $U(0, 1)$ 变量, 但只要进行一次自然对数的计算, 然而, 方法二还要进行一次 $n-1$ 个数据的排序。在计算机上的实际运行表明, 方法二一般要比方法一快。

这样我们通过 $U(0, 1)$ 获得了 n 个服从 $Exp(1)$ 的随机数 x_1, x_2, \dots, x_n , 通过公式(3), 就可以计算积分区间为 $(0, \infty)$ 的定积分了。

[例 5] 用 Monte Carlo 方法计算积分 $\int_0^{\infty} e^{-x^2} dx$

解: 为了说明产生 n 个服从 $Exp(1)$ 的随机数的方法一与方法二的效率的不同, 我们分别用这两种方法估计所求积分值, 都取 $n=20$ 。

方法一:

由 Mat lab 编程可得到计算结果参见表 7, 其中 u 是由均匀分布随机数产生器 `unifrnd` 随机产生的 20×1 的列向量。

表 7 样本平均值法计算积分 $\int_0^{\infty} e^{-x^2} dx$ (方法一)

i	u_i	$x_i = -\ln u_i$	$f(x_i) = e^{-x_i^2}$	$g(x_i) = e^{x_i}$	i	u_i	$x_i = -\ln u_i$	$f(x_i) = e^{-x_i^2}$	$g(x_i) = e^{x_i}$
1	0.0972	2.331	0.0044	10.2886	11	0.3315	1.1043	0.2954	3.017
2	0.293	1.2274	0.2217	3.4124	12	0.5752	0.553	0.7365	1.7384
3	0.6787	0.3876	0.8605	1.4734	13	0.1368	1.9891	0.0191	7.3089
4	0.9798	0.0204	0.9996	1.0206	14	0.8678	0.1418	0.9801	1.1523
5	0.3994	0.9177	0.4308	2.5035	15	0.3293	1.1109	0.2911	3.0371
6	0.8209	0.1974	0.9618	1.2182	16	0.488	0.7174	0.5977	2.0492
7	0.1316	2.0278	0.0164	7.5977	17	0.4315	0.8404	0.4935	2.3173
8	0.8449	0.1685	0.972	1.1836	18	0.2605	1.345	0.1638	3.8382
9	0.2327	1.4579	0.1194	4.2969	19	0.983	0.0171	0.9997	1.0173
10	0.0672	2.7008	0.0007	14.891	20	0.9657	0.0349	0.9988	1.0355

由上表, 利用公式(3)得到 $\int_0^{\infty} e^{-x^2} dx$ 的估计值为

$$\hat{\theta}_z = \frac{1}{20} \sum_{i=1}^{20} f(x_i) g(x_i) = \frac{1}{20} \sum_{i=1}^{20} e^{-x_i^2} e^{x_i} = 0.8255.$$

方法二:

同样利用 Mat lab 编程, 此时应用 `unifrnd` 产生一个 39×1 的随机数列向量 u 。前 20 个用来计算 $r = \ln(\prod_{i=1}^{20} u_i)$, 后 19 个重新按从小到大排序后加上 0, 1 组成向量 y , 具体结果如下:

u 的前 20 个分量是

0.6967 0.1471 0.0576 0.0634 0.7228 0.0337
0.6957 0.2212 0.6254 0.5133 0.5757 0.2872
0.8927 0.0926 0.2925 0.8219 0.994 0.0678
0.9731 0.9228

计算可得 $r = \ln(\prod_{i=1}^{20} u_i) = -23.1366$ 。

其它数据参见表 8:

表 8 样本平均值法计算积分 $\int_0^{\infty} e^{-x^2} dx$ (方法二)

i	u_{i+1}	y_i	$x_i = (y_i - 1 - y_i) \cdot r$	$f(x_i) = e^{-x_i^2}$	$f(x_i) = e^{x_i}$
0		0			
1	0.9587	0.0038	0.0874	0.9924	1.0913
2	0.5632	0.0341	0.7019	0.611	2.0175
3	0.9505	0.0611	0.624	0.6774	1.8664
4	0.3182	0.1291	1.5747	0.0838	4.8294
5	0.2659	0.2196	2.0933	0.0125	8.1115
6	0.0611	0.255	0.8182	0.512	2.2665
7	0.3328	0.2659	0.2524	0.9383	1.2872
8	0.2924	0.2875	0.4991	0.7795	1.6472
9	0.4025	0.2924	0.1146	0.9869	1.1214
10	0.0341	0.3182	0.597	0.7002	1.8166
11	0.2875	0.3328	0.3365	0.8929	1.4
12	0.2196	0.4025	1.613	0.0741	5.0179
13	0.5659	0.5227	2.7812	0.0004	16.1384
14	0.8701	0.5632	0.9364	0.4161	2.5507
15	0.255	0.5659	0.064	0.9959	1.066
16	0.5918	0.5918	0.5988	0.6987	1.82
17	0.5227	0.8701	6.4385	0	625.467
18	0.1291	0.9505	1.8609	0.0313	6.4297
19	0.0038	0.9587	0.1899	0.9646	1.2091
20		1	0.9548	0.4018	2.5982

由上表, 利用公式(3)得到 $\int_0^{\infty} e^{-x^2} dx$ 的估计值为 $\hat{\theta}_z = \frac{1}{20} \sum_{i=1}^{20} f(x_i) g(x_i) = \frac{1}{20} \sum_{i=1}^{20} e^{-x_i^2} e^{x_i} = 0.8776$ 。

(3) 被积区间上下限均为 ∞

当 a, b 为 ∞ 时, 即考虑区间 $(-\infty, \infty)$, 所求积分为

$$\theta = \int_0^\infty f(x) dx = \int_0^\infty \frac{f(x)}{g(x)} g(x) dx = E \left[\frac{f(x)}{g(x)} \right].$$

取 $g(x)$ 为定义在整个实数 R 上的标准正态分布

$$N(0,1) \text{ 的密度函数, 即取 } g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, x \in (-\infty, \infty).$$

设 x_1, x_2, \dots, x_n 是来自 $N(0,1)$ 的 n 个随机数, 则所求积分的估计值为

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{g(x_i)} = \frac{1}{n\sqrt{2\pi}} \sum_{i=1}^n f(x_i) e^{\frac{x_i^2}{2}} \quad (4)$$

如果我们能够利用 $U(0,1)$ 产生服从 $N(0,1)$ 的随机数, 则利用公式(4)就可估计得到积分区间为无穷的定积分了。

首先我们有下面一个重要的事实:

命题: 如果 ξ, η 是相互独立的随机变量, 并且都服从 $N(0,1)$ 分布, 则化成极坐标 $\rho\sqrt{\xi^2+\eta^2}$ 和

$\varphi = \arctan(\frac{\eta}{\xi})$, 它们是相互独立的。

证明: 采用极坐标 $x=r \cos\theta, y=r \sin\theta$

则 $r=\sqrt{x^2+y^2}, \theta=\arctan(\frac{y}{x})$, 其中 $r \geq 0, \theta \in (0, 2\pi)$ 。

因为 (ξ, η) 的联合密度函数是 $p(x, y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}$,

并且变换的 Jacobi 行列式为 $J = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix}$

$$= \begin{vmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r,$$

于是 (ρ, φ) 的联合密度函数为 $P(r, \theta) = \frac{r}{2\pi} e^{-\frac{r^2}{2}}$ 。

由此可知 (ρ, φ) 是相互独立的随机变量, 并且

$r=\sqrt{x^2+y^2}$ 的密度函数是 $R(r) = \begin{cases} re^{-\frac{r^2}{2}}, & r \geq 0 \\ 0, & r < 0 \end{cases}$, 而

$\theta=\arctan(\frac{y}{x})$ 的密度函数是: $U(\theta) = \begin{cases} \frac{1}{2\pi}, & 0 \leq \theta \leq 2\pi \\ 0, & \text{其它} \end{cases}$ 。

即 θ 服从 $[0, 2\pi]$ 上的均匀分布。证毕。

利用上述命题我们就可以产生标准正态分布的随机数, 有如下引理:

引理: 设 U_1, U_2 是独立同分布的 $U(0,1)$ 变量, 令

$$X_1 = (-2\ln U_1)^{1/2} \cos(2\pi U_2)$$

$$X_2 = (-2\ln U_1)^{1/2} \sin(2\pi U_2) \quad (5)$$

则 X_1 与 X_2 独立, 均服从标准正态分布。

证明: 我们考虑此变换的 Jacobi 行列式 J

首先我们知道

$$\begin{vmatrix} \frac{\partial X_1}{\partial U_1} & \frac{\partial X_1}{\partial U_2} \\ \frac{\partial X_2}{\partial U_1} & \frac{\partial X_2}{\partial U_2} \end{vmatrix} = \begin{vmatrix} -\frac{1}{U_1\sqrt{-2\ln U_1}} \cos(2\pi U_2) & -2\pi(-2\ln U_1)^{1/2} \sin(2\pi U_2) \\ -\frac{1}{U_1\sqrt{-2\ln U_1}} \sin(2\pi U_2) & 2\pi(-2\ln U_1)^{1/2} \cos(2\pi U_2) \end{vmatrix} = -\frac{2\pi}{U_1},$$

于是有: $J^{-1} = \frac{2\pi}{U_1}$, 即 $J = \frac{U_1}{2\pi}$ 。

所以, X_1, X_2 的联合密度函数是

$$\begin{aligned} p(x, y) &= p(u_1, u_2) J = \frac{U_1}{2\pi} = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} \\ &= \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \right) \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} \right) \end{aligned}$$

所以, X_1 与 X_2 相互独立, 且均服从标准正态分布。证毕。

由此引理, 我们可以给出同时产生两个 $N(0,1)$ 的随机数的方法:

1) 由 $U(0,1)$ 独立产生两个随机数 u_1, u_2 ,

2) 用(5)式计算 x_1, x_2 。

则 x_1, x_2 就是服从标准正态分布的随机数。

注意到, 利用(5)计算时, 用到了自然对数, 同指数分布的方法一, 计算自然对数需要用到级数展开, 每产生一个随机数就要用到一次级数展开, 效率不高。

其实由于标准正态分布的在概率统计中有着十分重要的地位, 许多人致力于研究 $N(0,1)$ 随机数的产生, 因而 $N(0,1)$ 变量的产生有许多方法。

我们考虑用中心极限定理为理论基础的一个著名的近似方法。即用 n 个 $U(0,1)$ 变量产生一个 $N(0,1)$ 变量, 具体的实施步骤为:

1) 由 $U(0,1)$ 独立产生 n 个随机数 u_1, u_2, \dots, u_n ,

2) 计算 $x = \sqrt{12n} (\bar{u} - \frac{1}{2})$, 其中 $\bar{u} = \frac{1}{n} \sum_{i=1}^n u_i$ 。

则 x 就是服从标准正态分布的随机数。

在实用中常取 $n=6$ 或 $n=12$ 。

由上述方法, 我们就可得到来自 $N(0,1)$ 的 n 个随

机数 x_1, x_2, \dots, x_n , 则利用公式(4) $\hat{\theta}_2 = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{g(x_i)} =$

$\frac{1}{n\sqrt{2\pi}} \sum_{i=1}^n f(x_i) e^{\frac{x_i^2}{2}}$, 我们就可计算积分区间为无穷的

定积分 $\theta = \int_{-\infty}^{\infty} f(x) dx$ 了。

[例 6] 用 Monte Carlo 方法计算积分值 $\int_{-\infty}^{\infty} e^{-x^2} dx$

表 9 样本平均值法计算积分 $\int_{-\infty}^{\infty} e^{-x^2} dx$ (方法一)

i	u_i	x_i	$g(x_i)$	$f(x_i)=e^{-x_i^2}$	i	u_i	x_i	$g(x_i)$	$f(x_i)=e^{-x_i^2}$
1	0.2323	-0.7069	0.6067	0.3108	11	0.4377	-1.0775	0.3132	0.2233
2	0.3179	1.5556	0.0889	0.119	12	0.5918	-0.701	0.6118	0.312
3	0.7888	-0.4573	0.8113	0.3593	13	0.1145	-0.8765	0.4638	0.2717
4	0.6344	-0.5151	0.7669	0.3494	14	0.3192	1.8884	0.0283	0.0671
5	0.6598	-0.8867	0.4556	0.2693	15	0.6219	0.7685	0.554	0.2969
6	0.5376	-0.2134	0.9555	0.39	16	0.8946	-0.5996	0.698	0.3333
7	0.9192	0.0782	0.9939	0.3977	17	0.9636	0.2702	0.9296	0.3846
8	0.7805	-0.4029	0.8501	0.3678	18	0.0201	0.0342	0.9988	0.3987
9	0.3267	-0.1057	0.9889	0.3967	19	0.1107	-1.7206	0.0518	0.0908
10	0.7387	-1.492	0.1079	0.1311	20	0.403	1.2009	0.2364	0.194

其中, $g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}}$ 。

由上表, 利用公式(4)计算 $\int_{-\infty}^{\infty} e^{-x^2} dx$ 的估计值为

$$\hat{\theta}_2 = \frac{1}{20} \sum_{i=1}^{20} \frac{f(x_i)}{g(x_i)} = 1.7792$$

表 10 样本平均值法计算积分 $\int_{-\infty}^{\infty} e^{-x^2} dx$ (方法二)

i	x_i	$g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}}$	$f(x_i) = e^{-x_i^2}$	i	x_i	$g(x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x_i^2}{2}}$	$f(x_i) = e^{-x_i^2}$
1	0.178	0.3927	0.9688	11	0.9698	0.2493	0.3904
2	0.9348	0.2577	0.4173	12	-0.3488	0.3754	0.8854
3	0.5444	0.344	0.7435	13	0.7205	0.3077	0.5951
4	1.4279	0.1439	0.1302	14	-0.0331	0.3987	0.9989
5	0.7869	0.2927	0.5384	15	0.4946	0.353	0.783
6	-0.4116	0.3665	0.8441	16	-0.1268	0.3957	0.984
7	1.558	0.1185	0.0883	17	0.9931	0.2436	0.373
8	-2.1613	0.0386	0.0094	18	-0.8443	0.2793	0.4902
9	-1.1468	0.2067	0.2684	19	-0.0128	0.3989	0.9998
10	-1.2671	0.1788	0.2008	20	-0.6177	0.3296	0.6828

由上表, 利用公式(4)计算 $\int_{-\infty}^{\infty} e^{-x^2} dx$ 的估计值为

$$\hat{\theta}_2 = \frac{1}{20} \sum_{i=1}^{20} \frac{f(x_i)}{g(x_i)} = 1.7818。$$

解:

方法一.

利用公式(5)来产生服从 $N(0,1)$ 分布的随机数, 首先利用均匀分布随机数产生器 `unifrnd` 产生一个 20×1 的随机数列向量 u , 然后分别由 u_i, u_{i+1} 产生服从 $N(0,1)$ 分布的随机数 $x_i, x_{i+1}, i=1, 3, 5, \dots, 19$ 。

利用 Mat lab 编程得到计算结果参见表 9, 具体程序详见附录。

方法二:

利用近似法, 不妨取 $n=12$, 即由 12 个独立的服从 $U(0,1)$ 分布的随机数产生一个 $N(0,1)$ 随机数 x_i , 利用 Mat lab 编程, 我们采用一个循环产生服从 $N(0,1)$ 分布的 20×1 随机数列向量 x 。

具体计算结果参见表 10, 具体程序详见附录。

4 Monte Carlo 方法在多重定积分计算中的应用

考虑 n 维空间 R^n 中函数 $f(X)$ 在有界闭域 D 内的积分, 这里 $X=(x_1, x_2, \dots, x_n)$ 表示 n 维空间中的点, 假设函数 $f(X)$ 在 D 可积, 即 $I = \int_D f(X) dX$ 存在。

现在用 Monte Carlo 方法来求解此积分并估计其误差, 最后给出欲达到指定精度所需的数据量。

4.1 积分方法

设 Ω 是 R^n 中的可度量区域, $|D|$ 是 Ω 的度量值, 在 Ω 上定义了一个可积函数 $f(X)$, $X \in \Omega$, 现在考虑积分 $I = \int_D f(X) dX$ 。

取一维区间 (a, b) , 使 $\Omega \in (a, b)^n$, $(a, b)^n$ 是 (a, b) 的 n 次笛卡尔积。取 n 个相互独立的在 (a, b) 上均匀分布的随机变量 $\xi_1, \xi_2, \dots, \xi_n$, 令 $\xi = (\xi_1, \xi_2, \dots, \xi_n)$, 则 ξ 在 Ω 中是均匀分布的。即有 $P(\xi \in V | \xi \in \Omega) = \frac{|V|}{|D|}$, 这里 $|V|$ 表示 V 的度量, $V \subseteq \Omega$ 。

由积分中值定理, 得

$$E[f(\xi) | \xi \in \Omega] = \frac{\int_D f(X) dX}{|D|} = \frac{I}{|D|}$$

由此得计算方法如下:

(1) 产生一个 n 维随机向量 $\xi = (\xi_1, \xi_2, \dots, \xi_n)$, ξ 在 Ω 中是均匀分布, 若某个向量值 $\xi^{(i)} = (\xi_1^{(i)}, \xi_2^{(i)}, \dots, \xi_n^{(i)}) \notin \Omega$, 则舍去, 设总共有 m 个向量属于 Ω , 记为 $\xi^{(1)}, \xi^{(2)}, \dots, \xi^{(m)}$ 。

(2) 记 $I_m = \frac{|D|}{m} \sum_{i=1}^m f(\xi^{(i)})$, 其中 $|D|$ 是 Ω 的度量值。

由中心极限定理 $\lim_{m \rightarrow \infty} I_m = I$, 这样, 只要取适当大小的 m 值, 即可求得满足精度的积分值。

4.2 误差分析

首先求出 I_m 的期望和方差:

$$E(I_m) = E\left[\frac{|D|}{m} \sum_{i=1}^m f(\xi^{(i)})\right] = \frac{|D|}{m} \sum_{i=1}^m E[f(\xi^{(i)})] = |D| E[f(\xi^{(1)})] = I$$

$$\begin{aligned} D(I_m) &= D\left[\frac{|D|}{m} \sum_{i=1}^m f(\xi^{(i)})\right] = \frac{|D|^2}{m^2} \sum_{i=1}^m D[f(\xi^{(i)})] \\ &= \frac{|D|^2}{m} D[f(\xi^{(1)})] \end{aligned}$$

由中心极限定理知: $\frac{I_m - E(I_m)}{\sqrt{D(I_m)}} \rightarrow N(0, 1)$ 问题化

为: 对 $\forall \varepsilon > 0$ 及给定的显著性水平 α , 要使

$P(|I_m - I| < \varepsilon) > 1 - \alpha$ 或 $P(|I_m - I| \geq \varepsilon) \leq \alpha$ 成立, 则 m 应取多大?

$$\text{根据 } P(|I_m - I| < \varepsilon) = P\left(\left|\frac{I_m - I}{\sqrt{D(I_m)}}\right| < \frac{\varepsilon}{\sqrt{D(I_m)}}\right) >$$

$(1 - \alpha)$, 查正态分布表, 得临界值 $u_{1-\frac{\alpha}{2}}$, 即有 P

$$\left(\left|\frac{I_m - I}{\sqrt{D(I_m)}}\right| < u_{1-\frac{\alpha}{2}}\right) = 1 - \alpha, \text{ 所以, 只要 } \frac{\varepsilon}{\sqrt{D(I_m)}} \geq$$

$u_{1-\frac{\alpha}{2}}$ 即 $m \geq \frac{|D| u_{1-\frac{\alpha}{2}}^2}{\varepsilon^2} D[f(\xi^{(1)})]$, 就能使 I_m 与 I 之差小于 ε 得概率大于 $1 - \alpha$ 。

若方差 $D[f(\xi)]$ 不易求出, 则可用频率法公式, 此

时 $m = \frac{h^2 |D| u_{1-\frac{\alpha}{2}}^2}{\varepsilon^2} p \cdot q$, 其中 h 是 $f(\xi)$ 在 Ω 中的最大

值, p, q 分别是 $\frac{I}{|D|h}$ 和 $1 - \frac{I}{|D|h}$ 。

4.3 实例考察

例 1. 求二重积分

$$I = \iint_D (x^2 + y^2 - x) d\sigma, D = \{(x, y) | y \leq 2, y \geq x, y \leq 2x\}$$

解: 产生 $U(0, 2)$ 分布的随机数序列 (x_i, y_i) , 设有 m

对 (x_i, y_i) 落在 D 中, 则 $I_m = \frac{|D|}{m} \sum_{i=1}^m f(x_i, y_i)$, 其中 $|D|$ 是积分区域 D 的面积。下表是 m 取不同值时的计算结果和误差:

m	计算值	理论值	绝对误差	相对误差
20	2.271	2.166	0.105	0.048
40	2.459	2.166	0.292	0.135
100	2.206	2.166	0.039	0.018
500	2.114	2.166	0.053	0.025
1000	2.193	2.166	0.026	0.012

误差分析:

若给定 $\alpha = 0.05, \varepsilon = 0.01$, 即要求以 95% 的概率保证误差不超过 1%。由此可计算得:

$$m = \frac{1.96^2}{0.01^2} D[f(\xi^{(1)})] \approx 88126$$

例 2. 计算二重积分 $I = \iint_D e^{-(x^2+y^2)} d\sigma, D = x^2 + y^2 \leq 1$

解: 产生 m 对单位圆上均匀分布的随机数对 $(\xi_i,$

$\eta_i)$, 则由公式可得积分值为 $I_m = \frac{|D|}{m} \sum_{i=1}^m f(x_i, y_i)$, 其中

$|D|$ 是积分区域 D 的面积。下表是 m 取不同值时的计算结果和误差:

m	计算值	理论值	绝对误差	相对误差
20	1.912	1.985	0.074	0.037
40	1.991	1.985	0.006	0.003
100	2.03	1.985	0.044	0.022
500	2.009	1.985	0.023	0.012
1000	1.997	1.985	0.011	0.005

误差分析:

若给定 $\alpha=0.05, \varepsilon=0.01$, 即要求以 95% 的概率保证误差不超过 1%。由此可计算得:

$$m = \frac{1.96^2 |D|^2}{0.01^2} D[f(\xi, \eta)] \approx 12419$$

本计算方法使用方便, 计算精度高, 且能根据用户的要求调整计算量, 从而达到预期的计算目的, 经大量例子测试, 实际数据量只需 200-300 即可达到一般的精度要求。

5 Monte Carlo 方法在概率计算中的应用

概率是用于描述不确定性事件发生的可能性的, 对于一些比较简单的随机事件的概率计算问题, 我们可以通过一些比较常用的概率计算公式进行准确的计算, 但是随着研究问题的深入和事件本身的复杂性, 导致了概率计算的困难, 甚至是根本无法计算。然而随着计算机技术的发展, 近代发展起来的 Monte Carlo 方法在复杂事件的概率计算汇总却起了十分重要的作用。

下面我们具体的事例的来说明一下 Monte Carlo 方法在计算概率时的应用。

5.1 会面问题

甲乙两艘轮船驶向一个不能同时停泊两艘轮船的码头停泊, 它们在一昼夜到达的时刻是等可能的, 如果甲船停泊的时间是 1 小时, 乙船停泊的时间是 2 小时, 求它们的任何一艘都不需要等待码头空出的概率。

解: 该问题是一个几何概率。

设甲到达时刻为 x , 乙到达时刻为 y 。由于甲、乙轮船在一昼夜到达的时刻是等可能的, 则 $0 \leq x \leq 24, 0 \leq y \leq 24$ 且 x, y 都是服从 $[0, 24]$ 上的均匀分布。

由条件可知, 当甲先到时, 则 $y > x$ 。若乙不需要等待, 则 $y - x > 1$ 。当乙先到时, 则 $x > y$ 。若甲不需要等待, 则 $x - y > 2$ 。则不需要等待的概率为图中阴影部分面积与

总面积的比值。即

$$P = \frac{0.5 \times 23^2 + 0.5 \times 22^2}{24 \times 24} = 0.8793。$$

我们欲利用 Monte Carlo 方法来求解此概率, 首先应产生 $[0, 1]$ 中的均匀分布随机数, 利用简单的一个变换得到 $[0, 24]$ 中均匀分布的随机数 x, y , 然后判断条件, 具体 Mat lab 程序如下:

```
>>N=5000; % 定义模拟次数
>>n=0;
>>for i=1:N
    u=unifrnd(0,1,2,1);
    x=24*u(1); %模拟甲到达时间
    y=24*u(2); %模拟乙到达时间
    if (y-x>1 | x-y>2)
        n=n+1; % 计算不需要等待的次数
    end
end
>>prob=n/N %所求概率的模拟值
```

某次模拟的结果为: prob=0.8786, 即甲、乙任何一艘都不需要等待码头空出的概率是 0.8786。这一结果与前面用公式精确计算得到的结果十分接近, 绝对误差为 $|0.8786 - 0.8793| = 0.0007$, 相对误差 $\sigma = \frac{0.0007}{0.8786} \times 100\% = 0.08\%$ 。

5.2 掷球入盒问题

把 10 个球掷进 4 个盒子, 设每个球落进每个盒子的可能性相等, 求落在第一个盒子的球数的数学期望。

解: 记随机变量 ξ 为落在第一个盒子的球数。

由于每个球落进每个盒子的可能性相等, 则每个球落进第一个盒子的概率为 $P=1/4$, 不落进第一个盒子的概率为 $q=1-p=3/4$, 再由各个球是否落进第一个盒子是独立的。因此 ξ 服从 $B(10, 1/4)$ 的二项分布, 则数学期望 $E\xi = N \cdot P = 10 \times \frac{1}{4} = 2.5$ 。

计算机模拟思想:

每次模拟产生 10 个 1 到 4 之间的随机数, 统计 1 的个数, 即为落进第一个盒子的球数。如三次模拟数值为 2.54, 2.502, 2.492。

掷球入盒问题的 Matlab 程序:

```
>>N=5000; %定义模拟的次数
>>for L=0:N-1
    num=0;
```

```

for i=0:9 %模拟 10 个球落进的盒子序号
    k=rand ()%4+1; %计算随机产生的球序号
    (1,2,3,4)
    if (k==1) num++; %统计每次模拟落在第一个
    盒子的球数
    end
end
s=s+num;
end
>>prob=1.0*s/N; %计算模拟期望值

```

上面我们仅用两个比较简单的例子来说明了 Monte Carlo 方法在概率计算中的应用,并可看出,用 Monte Carlo 方法模拟估计的结果与用公式精确计算得到的结果十分接近,故而用 Monte Carlo 方法计算概率问题是一个行之有效的办法。

6 结论

本文通过具体的实例,对 Monte Carlo 方法在计算定积分和概率方的应用作了简单示范,在计算机高速发展的今天,借助与数学软件对各种随机数的产生加以应用。其实 Monte Carlo 方法在贷款组和、信用风险、房地产业谨防投资风险、随机游动、投标报价等各个方面都有着很多应用。

参 考 文 献

[1] 茆诗松,王静龙,濮晓龙. 高等数理统计. 北京:高等教育出版社,2006年5月.

- [2] 田铮,肖华勇. 随机数学基础. 北京:高等教育出版社,2005年7月.
- [3] 石博强,赵金编. MATLAB 数学计算与工程分析范例教程. 北京:中国铁道出版社,2005年4月.
- [4] 王岩,隋思涟,王爱青. 数理统计与 MATLAB 工程数据分析. 北京:清华大学出版社,2006年9月.
- [5] 刘顺忠. 数理统计理论、方法、应用和软件计算. 上海:华中科技大学出版社,2005年9月.
- [6] 邓勃编. 数理统计方法在分析测试中的应用. 北京:化学工业出版社,1984年11月.
- [7] 庄楚强,吴亚森. 应用数理统计基础. 广州:华南理工大学出版社,1992年11月.
- [8] 高文森,潘伟. 大学数学 随机数学. 北京:高等教育出版社,2004年10月.
- [9] 林元烈,梁宗霞. 随机数学引论. 北京:清华大学出版社,2003年6月.
- [10] 萧树铁. 大学数学 数学实验. 北京:高等教育出版社,1999年7月.
- [11] 王兵团,桂文豪. 数学实验基础. 北京:北方交通大学出版社,2003年9月.
- [12] 贾晓峰. 微积分与数学模型 上. 北京:高等教育出版社,1999年8月.
- [13] 郭大钧. 数学分析. 济南:山东科技出版社,1985年11月.
- [14] 裴礼文. 数学分析中的典型问题与方法. 北京:高等教育出版社,2006年5月.
- [15] 邓乐斌. 数学分析的理论、方法与技巧. 北京:华中科技大学出版社,2005年7月.

(责任编辑: 冯沛)

上接第44页

流向,着力向农村地区倾斜,以改变农村教育落后的状况,逐步实现资源配置在城乡、区域间的公平、合理。同时,我们要进一步规范各级学校招生录取的制度和程序,同时要加强对政府主管部门和社会舆论的监督,以消除腐败,最大程度的实现教育公平。

(3) 政府应充分发挥建设和谐社会的主导作用

政府是建设和谐社会的主导力量,政府的执政理念和服务能力是建设和谐社会的关键。从调查的结果来看,一个务实高效、以人为本的服务型政府是广大民众的迫切期望。教育、医疗、住房贵等涉及人民切身利益的问题,政府必须要像关注“菜篮子”、“米袋子”建设一样,要充分发挥政府的扶持、监管和调控作用,而不能简单的完全推给市场。当前我国的最低生活保障、医疗保险、养老保险、失业保险覆盖率和社会保障水平较低,民生存在严重的后顾之忧,这不仅加大了城乡居民的生活压力,也影响了居民消费的信心,进

而影响了国内消费需求的扩大,并将由此引发一系列矛盾。为此,我们必须加快覆盖城乡的社会保障体系建设,加大财政投入力度,提高广大城乡居民的社会保障水平,特别是要显著提高农村和城镇贫困阶层、农民工和落后地区农村居民的社会保障水平,解除其生活后顾之忧。

(4) 公平正义、安居乐业、生活富裕和诚信友爱应成为和谐社会建设的重要目标

从调查结果看,公平正义、安居乐业、生活富裕和诚信友爱是城乡广大民众对和谐社会建设的主要期望,这种看来近似“桃花源”式的期望,其实正是现实生活中所存在不和谐因素的集中折射,它与现实社会经济生活中存在的突出矛盾和急需解决的问题是完全吻合的。因此,这些期望应该成为我们在科学发展观的引领下,强调以人为本,建设社会主义和谐社会的重要目标。

(责任编辑: 王锋)