随 机 算 法

第6章 蒙特卡罗算法

6.1 Monte Carlo方法---起源

蒙特卡罗(Monte Carlo)方法，也称为计算机随机模拟方法，是一种基于"随机数"的计算方法。

这一方法源于美国在第二次世界大战中研制原子弹的“曼哈顿计划”。该计划的主持人之一、数学家冯·诺伊曼用驰名世界的赌城—摩纳哥的Monte Carlo—来命名这种方法，为它蒙上了一层神秘色彩。

John Von Neumann
(1903-1957)

Monte-Carlo Monaco

Monte Carlo方法的应用

- 物理：核物理，热力学与统计物理，粒子输运问题等
- 数学：多重积分、解微分方程、非线性方程组求解等
- 工程领域：真空技术，水力学，激光技术等
- 经济学领域：期权定价、项目管理、投资风险决策等
- 其他领域：化学、医学，生物，生产管理、系统科学、公用事业等方面，随着科学技术的发展，其应用范围将更加广泛。

基本思想：

针对待求问题，根据物理现象本身的统计规律，或人为构造一合适的依赖随机变量的概率模型，使某些随机变量的统计量为待求问题的解，进行大统计量的统计实验方法或计算机随机模拟方法。

- **理论依据：**

大数定理：均匀分布的算术平均收敛于真值

中心极限定理：置信水平下的统计误差

- **两个例子：**

Buffon投针实验求 π

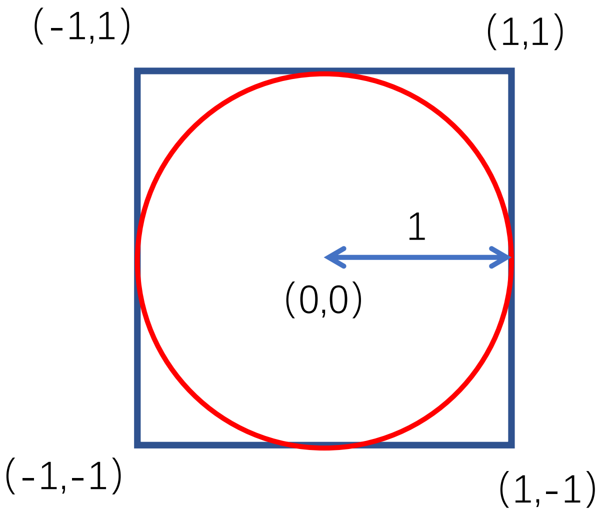
射击问题（打靶游戏）

■ 考虑估计常数值π

设(X,Y)是在平面上以原点$(0,0)$为中心的一个2×2的正方形内均匀随机选取的一点.

如果令$Z=\begin{cases}1 & \text{如果}\sqrt{X^2+Y^2}\le 1\\ 0 & \text{其他}\end{cases}$

$\Rightarrow$ $Pr(Z=1)=\dfrac{\pi}{4}$



■ 假设进行这个试验 m 次（X,Y是独立选取的）

Z_i表示第 i 次试验的 Z 值。如果$W=\sum_{i=1}^{m} Z_i$，那么

$$E[W]=E\left(\sum_{i=1}^{m} Z_i\right)=\sum_{i=1}^{m} E[Z_i]=\frac{m\pi}{4}$$

因此$W'=(4/m)W$是 π 的自然估计。利用切尔诺夫界，计算得

$$Pr(|W'-\pi|\geq \varepsilon\pi)=Pr\left(\left|W-\frac{m\pi}{4}\right|\geq\frac{\varepsilon m\pi}{4}\right)=Pr(|W-E[W]|\geq \varepsilon E[W])$$

$$\leq 2e^{-\frac{m\pi\varepsilon^2}{12}}.$$

随着m增大，估计值偏离π的概率越小.

定义 6.1

随机化算法给出 V 值的一个(ε, δ)近似，如果算法的输出 X 满足

ε: 估计精度 $$Pr(|X - V| \le \varepsilon V) \ge 1 - \delta$$

δ: 有效性 $\Leftrightarrow \mathbb{P}\{|X-V| \ge \varepsilon V\} \le \delta$

■ 我们估计 π 的方法给出一个(ε, δ)近似.

$$\mathbb{P}\{|W'-\pi| \ge \varepsilon\pi\} \le 2e^{\frac{m\pi\varepsilon^2}{12}}$$

令 $2e^{\frac{m\pi\varepsilon^2}{12}} \le \delta$ 即可, 解得 $m \ge \frac{12\ln\frac{2}{\delta}}{\pi\varepsilon^2}$

定理 6.1

设$(x_1, \dots, x_m)$是独立同分布示性随机变量，$\mu = E[X_i]$，若$m \ge$ ${(3\ln(2/\delta))}/{\varepsilon^2\mu}$，那么

$$Pr\left(\left|\underbrace{\frac{1}{m}\sum_{i=1}^{m} X_i}_{\text{算术平均值}} - \mu\right| \ge \varepsilon\mu\right) \le \delta$$

即 m 个样品提供μ的一个(ε, δ)近似.

定理6.1证明

令 $Z=\sum_{i=1}^{m}X_i$，$\mathbb{E}[Z]=m\mu$

$$\mathbb{P}\left\{\left|\frac{1}{m}\sum_{i=1}^{m}X_i-\mu\right|\geq\varepsilon\mu\right\}$$

$$=\mathbb{P}\left\{\left|\frac{1}{m}Z-\mu\right|\geq\varepsilon\mu\right\}$$

$$=\mathbb{P}\left\{\left|Z-m\mu\right|\geq\varepsilon m\mu\right\}$$

$$=\mathbb{P}\left\{\left|Z-\mathbb{E}[Z]\right|\geq\varepsilon\mathbb{E}[Z]\right\}$$

$$\leq 2e^{-\frac{\mathbb{E}[Z]\varepsilon^2}{3}}$$

切尔诺夫界

令 $2e^{-\frac{\mathbb{E}[Z]\varepsilon^2}{3}}\leq\delta$ 即可，

解得 $m\geq\dfrac{3\ln\frac{2}{\delta}}{\mu\varepsilon^2}$，证毕.

为什么用 $Z=\sum_{i=1}^{m}X_i$，而不是 $Z=\frac{1}{m}\sum_{i=1}^{m}X_i$？

若用 $Z=\frac{1}{m}\sum_{i=1}^{m}X_i$ 进行类似推导，最终结果为：

$$\mathbb{P}\left\{\left|Z-\mathbb{E}[Z]\right|\geq\varepsilon\mathbb{E}[Z]\right\}\leq 2e^{-\frac{\mathbb{E}[Z]\varepsilon^2}{3}}=2e^{-\frac{\mu\varepsilon^2}{3}}$$

定义 6.2

一个问题的完全多项式随机化近似方案（FPRAS）：

给定一个输入 $\boldsymbol{x}$，任意的参数 $\boldsymbol{\varepsilon}$，$\boldsymbol{\delta}$ 满足 $0<\boldsymbol{\varepsilon}, \boldsymbol{\delta}<1$，算法及时输出 $\boldsymbol{V(x)}$ 的一个 $(\boldsymbol{\varepsilon}, \boldsymbol{\delta})$ 近似，即多项式 $\mathbf{1}/\boldsymbol{\varepsilon}$、$\mathbf{ln}\boldsymbol{\delta}^{-1}$ 和输入 $\boldsymbol{x}$ 的大小。

$\ln\frac{1}{\delta}$

6.2 应用:DNF计数问题

· 使得DNF成立的赋值个数.

- 考虑析取范式（DNF）中布尔公式满足赋值个数的计数问题。
- 一个DNF公式是子句$C_1 \vee C_2 \vee \cdots \vee C_t$的析取（OR），其中每个子句是文字的合取（AND）。

  例如以下是一个DNF公式：$(x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (\bar{x}_1 \wedge x_3 \wedge x_4)$
- 对比MAXSAT问题：输入公式是一个子句集合的合取（AND），而每个子句又是文字的析取（OR），这通常称作合取范式（CNF）。

· 第五章定理5.4，最大可满足性MAXSAT问题，考虑的是CNF

· CNF：$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_2 \vee x_4)$

DNF：$(x_1 \wedge \bar{x}_2 \wedge x_3) \vee (\bar{x}_2 \wedge x_4)$

有使DNF不成立的一种赋值

n为文字个数

2^n为赋值总数

· 当$\overline{CNF}$ = DNF时，若CNF有成立的一种赋值，则使DNF成立的赋值个数 < 2^n

相反的，若使DNF成立的赋值个数为2^n，则CNF没有成立的一种赋值.

■ 朴素算法

算法 6.1 DNF计数算法I

输入： 一个有n个变量的DNF公式F.

输出： $Y = c(F)$的一个近似.

1. $X \leftarrow 0$.
2. 对$k = 1$到m;

随机穷举.

a) 产生一个从所有2^n个可能赋值中均匀随机地选取的n个变量的随机赋值;

b) 如果随机赋值满足F，则$X \leftarrow X + 1$

3. 返回$Y \leftarrow (X/m)2^n$.

证明：$Y=(X/m)2^n$ 是赋值计数 $c(F)$ 的一个近似

令随机变量 $X_k=\begin{cases}1, & \text{满足DNF公式} \\ 0, & \text{其它}\end{cases}$，$k=1\sim m$

$$X=\sum_k X_k$$

$$E[X]=\sum_k E[X_k]=m\cdot\frac{c(F)}{2^n}$$

$$E[Y]=(E[X]/m)\cdot 2^n=c(F)$$

由定理 6.1，$\frac{1}{m}\sum_{k=1}^{m}X_i=\frac{X}{m}$，$\mu=E[X_i]=\frac{c(F)}{2^n}$

若 $m\geq\frac{3\ln\frac{2}{\delta}}{\varepsilon^2\mu}=\frac{3\cdot 2^n\cdot\ln\frac{2}{\delta}}{\varepsilon^2\cdot c(F)}$，则 $\Pr\{\left|\frac{X}{m}-\mu\right|\geq\varepsilon\mu)\leq\delta$

定理 6.1

设$(x_1,\dots,x_m)$是独立同分布示性随机变量，$\mu=E[X_i]$，若$m\geq$ $(3\ln(2/\delta))/_{\varepsilon^2\mu}$，那么

$$Pr\left(\left|\underbrace{\frac{1}{m}\sum_{i=1}^{m}X_i}_{\text{算术平均值}}-\mu\right|\geq\varepsilon\mu\right)\leq\delta$$

即 m 个样品提供μ的一个(ε,δ)近似.

缺点：样本抽取数据集有效赋值不够稠密，m要足够大近似的 $c(F)$ 才有效.

■ DNF计数问题的完全多项式随机化近似方案（FPRAS）

算法 6.2 DNF计数算法II

输入： 一个有n个变量的DNF公式F.

输出： $\boldsymbol{Y = c(F)}$的一个近似.

1. $X \leftarrow 0$.
2. 对$k = 1$到m；

a) 以概率$|SC_i| / \sum_{i=1}^{t} |SC_i|$均匀随机地选取一个赋值$a \in SC_i$；

b) 如果a不在任一SC_j 中，$j < i$，则$X \leftarrow X + 1$（不在前面已有的赋值中）

3. 返回$\mathrm{Y} \leftarrow (X/m) \sum_{i=1}^{t} |SC_i|$.

- SC_i：满足子句 i 的赋值集合
- 假设DNF公式包含 t 个有效子句（存在赋值使子句成立）
- $|SC_i|$ 容易计算
  - 例：$(x_1 \wedge \bar{x}_2 \wedge x_4) \vee (x_1 \wedge x_5) \vee (x_2 \wedge \bar{x}_5)$

    $|SC_1| = 2^2 \qquad |SC_2| = 2^3 \qquad |SC_3| = 2^3$

- $|U| = \sum_{i=1}^{t} |SC_i|$，$c(F) = |\bigcup_{i=1}^{t} SC_i|$，则 $c(F) \leq |U|$
- 2.(a)步保证了均匀抽样：

$$\frac{|SC_i|}{\sum_{i=1}^{t}|SC_i|} \cdot \frac{1}{|SC_i|} = \frac{1}{|U|}$$

- 记 $S = \{(i,a) \mid a \in SC_i, \forall j < i, a \notin SC_j\}$，则 $|S| = |\bigcup_{i=1}^{t} SC_i| = c(F)$

  $|S|$ 表示2.(b)步 X 的大小
- $\frac{X}{m} = \frac{|S|}{|U|} \Rightarrow |S| = \frac{X}{m}|U|$（即输出）
- 性质：$\frac{X}{m} = \frac{|S|}{|U|} \geq \frac{1}{t}$

定理 6.2

当$m=\left\lceil(3t/\varepsilon^2)\ln(2/\delta)\right\rceil$时，DNF计数算法II是DNF计数问题的一个完全多项式随机化近似方案（FPRAS）.

- 当 $m=\left\lceil\frac{3t\ln\frac{2}{\delta}}{\varepsilon^2}\right\rceil$时，算法II输出的 $\frac{X}{m}\sum_{i=1}^{t}|SC_i|$ 是关于 $c(F)$ 的一个 (ε,δ) 近似
- 证明(定理6.1)

$$\mu=\mathbb{E}[X_i]=\mathbb{P}\{X_i=1\}$$
$$=\mathbb{P}\{X_i\in S\mid X_i\in U\}=\frac{|S|}{|U|}\geq\frac{1}{t}$$

则 $\frac{3\ln\frac{2}{\delta}}{\varepsilon^2\mu}\leq\frac{3t\ln\frac{2}{\delta}}{\varepsilon^2}$，

令 $m=\left\lceil\frac{3t\ln\frac{2}{\delta}}{\varepsilon^2}\right\rceil \Leftrightarrow m\geq\frac{3\ln\frac{2}{\delta}}{\varepsilon^2\mu}$

定理 6.1

设$(x_1,\dots,x_m)$是独立同分布示性随机变量，$\mu=E[X_i]$，若$m\geq$ $(3\ln(2/\delta))/_{\varepsilon^2\mu}$，那么

$$Pr\left(\left|\underbrace{\frac{1}{m}\sum_{i=1}^{m}X_i}_{\text{算术平均值}}-\mu\right|\geq\varepsilon\mu\right)\leq\delta$$

即 m 个样品提供μ的一个$(\varepsilon,\ \delta)$近似.

投针法

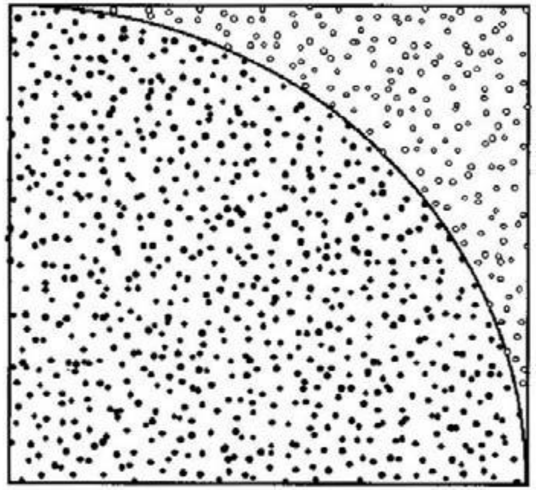
计算定积分的基本公式

$$S=\int_a^b f(x)dx=S_0\frac{n}{N}$$

S_0 为方形区域的面积，N是总点数，n 是掷入 $f(x)$ 下面积区域中的点数。

例如，用 Monte Carlo 方法求 π 值。产生一对在 $[0,1]$ 区间中均匀分布的随机数作为点的坐标 (x,y) 值，判断条件 $x^2+y^2\leq 1$ 是否成立，成立则计数 n 值，当总点数 N 足够大时：

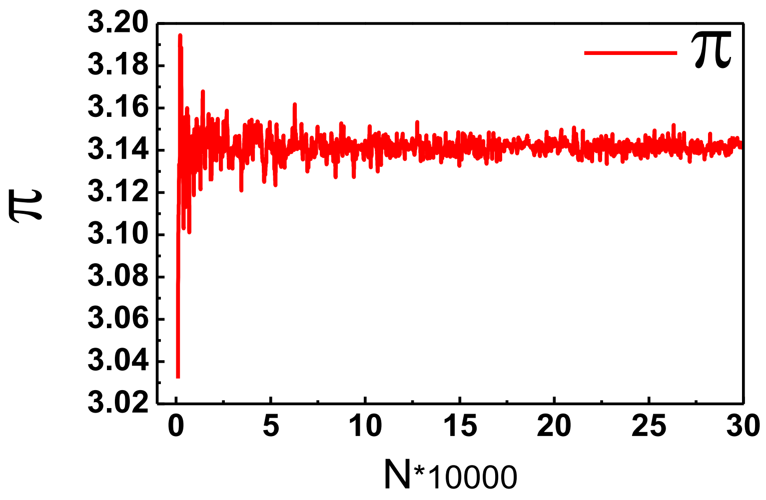
$$\frac{\pi}{4}=\lim_{N\to\infty}\frac{n}{N}$$



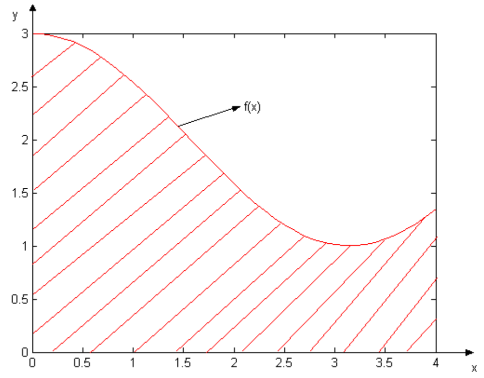
6.3 蒙特卡罗方法应用实例---数值积分

- 具体*FORTRAN*程序如下：

```
program main
    implicit none
    real(kind=8),parameter::m=1.0e8
    integer::n,counter=0
    real(kind=8)::x(1:m),y(1:m),Pi,r
    open(unit=10,file='random_num.txt')
    call random_seed()
    CALL random_number(x)
    call random_number(y)
    do n=1,m
        r=sqrt(x(n)**2+y(n)**2)
        if (r<=1) then
            counter=counter+1
        end if
    end do
    pi=(counter*1.0)*4.0/(m*1.0)
    write(10,*) 'The Pi is',pi
   end program main
```



例 计算定积分 $\theta=\int_0^4(\cos x+2.0)dx$



事实上，其精确解为 $\theta=8.0+\sin 4.0=7.2432$

用随机投点法求解：

liti27(0,4,4,1000000) result = 7.2336

注： 增加样本数目，可提高计算精度，但计算时间也会提高。

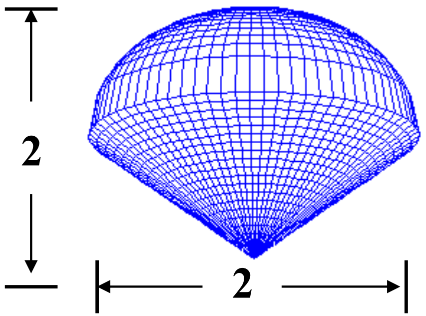
6.3 蒙特卡罗方法应用实例---数值积分

```
function result=liti27(a,b,m,mm)
%a是积分的下限
%b是积分的上限
%m是函数的上界
%mm 是随机实验次数
frq=0;
xrandnum = unifrnd(a,b,1,mm);
yrandnum = unifrnd(0,m,1,mm);

for ii=1:mm
    if (cos(xrandnum(1,ii))+2>=yrandnum(1,ii))
        frq=frq+1;
    end
end
result=frq*m*(b-a)/mm
```

例：冰淇淋锥的体积计算

$$z \ge \sqrt{x^2+y^2} \quad \& \quad z \le 1+\sqrt{1-x^2-y^2}$$



冰淇淋锥含于体积 = 8 的正方体

$$\Omega = \{(x,y,z) \mid -1 \le x \le 1, -1 \le y \le 1, 0 \le z \le 2\}$$

N个点均匀分布于正方体中,锥体中占有m个,则锥体与正方体体积之比近似为 m : N

$$\frac{V}{8} \approx \frac{m}{N}$$

蒙特卡罗方法计算冰淇淋锥的体积

```
function data=icecream(L)
if nargin==0,L=7;end
N=10000;
for k=1:L
   P=rand(N,3);
   x=2*P(:,1)-1;
   y=2*P(:,2)-1;
   z=2*P(:,3);
   R2=x.^2+y.^2;R=sqrt(R2);
   II=find(z>=R&z<=1+sqrt(1-R2));
   m=length(II);   q(k)=8*m/N;
end
data=[q; q-pi];
```

MC方法真正的优势在于计算多重积分。

设想一个物理系统由相互作用的多个粒子组成，如凝聚态物质中的原子、原子中的电子，而每个粒子都可以有几个自由度。

如对具有相互作用的 m 个原子组成的气体，经典配分函数：

$$Z = \int dr_1 \cdots dr_m \exp[-\beta \sum_{i<j} U(r_{ij})] \qquad \beta = \frac{1}{k_B T}$$

这个式子只有当 m 相当小时才可能用数值积分方法计算出来。

假设每个坐标取 10 个点，总共将有 10^{3m} 个网格点，对于 $m = 20$，如用万亿次计算机进行计算，需要10^{48} 秒（约为宇宙年龄的10^{29} 倍）。

6.3 蒙特卡罗方法应用实例---数值积分

若采用简单抽样的MC方法，则高维积分式可写为：

$$\int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \cdots \int_{a_n}^{b_n} dx_n \, f(x_1, x_2, \cdots x_n)$$

$$= \frac{1}{N} [\prod_{j=1}^{n} (b_j - a_j)] \sum_{i=1}^{N} f(x_{1i}, x_{2i}, \cdots x_{ni})$$

其中对每个坐标的抽样值是在相应的区间范围内均匀抽取的。

Thank You