《Python统计计算》

（2021年秋季学期）

翟祥
北京林业大学

E-mail: zhaixbh@126.com

统计计算

第13章 随机模拟

统计计算

- 在用数学模型，包括概率统计模型处理实际应用中的问题时，我们希望建立的模型能够尽可能地符合实际情况。
- 但是，实际情况是错综复杂的，如果一味地要求模型与实际完全相符，会导致模型过于复杂，以至于不能进行严格理论分析，结果导致模型不能使用。
- 所以，实际建模时会忽略许多细节，增加一些可能很难验证的理论假设，使得模型比较简单，可以用数学理论进行分析研究。
- 这样，简化的模型就可以与实际情况有较大的差距，即使我们对模型进行了完美的理论分析，也不能保证分析结果是可信的。
- 这一困难可以用随机模拟的方法解决。

随机模拟

- 模拟是指把某一现实的或抽象的系统的某种特征或部分状态，用另一系统（称为模拟模型）来代替或模拟。
- 为了解决某问题，把它变成一个概率模型的求解问题，然后产生符合模型的大量随机数，对产生的随机数进行分析从而求解问题，这种方法叫做随机模拟方法，又称为蒙特卡洛(Monte Carlo) 方法。

随机模拟计算的基本思路

1.针对实际问题建立一个简单且便于实现的概率统计模型，使所求的量（或解）恰好是该模型某个指标的概率分布或者数字特征。

2.对模型中的随机变量建立抽样方法，在计算机上进行模拟测试，抽取足够多的随机数，对有关事件进行统计

3.对模拟试验结果加以分析，给出所求解的估计及其精度(方差)的估计

4.必要时，还应改进模型以降低估计方差和减少试验费用，提高模拟计算的效率

- 例如，一个交通路口需要找到一种最优的控制红绿灯信号的办法，使得通过路口的汽车耽搁的平均时间最短，而行人等候过路的时间不超过某一给定的心理极限值。
- 十字路口的信号共有四个方向，每个方向又分直行、左转、右转。因为汽车和行人的到来是随机的，我们要用随机过程来描述四个方向的汽车到来和路口的行人到来过程。理论建模分析很难解决这个最优化问题。
- 但是，我们可以采集汽车和行人到来的频率，用随机模拟方法模拟汽车和行人到来的过程，并模拟各种控制方案，记录不同方案造成的等待时间，通过统计比较找出最优的控制方案。

随机模拟的应用

- 应用面广，适应性强。只要问题能够清楚地描述出来，就可以编写模拟程序产生大量数据，通过分析模拟数据解决问题。
- 算法简单, 容易改变条件, 但计算量大。
- 随机模拟用在科学研究中，常常作为探索性试验来使用。
- 随机模拟在科学研究中的另一种作用是说明新的模型或技术的有效性。在公开发表的统计学论文中，已经有一半以上的文章包括随机模拟结果（也叫数值结果），用来辅助说明自己提出的模型或方法的有效性。
- 随机模拟还是许多新的统计方法的主要工具，例如，蒙特卡洛检验，bootstrap 置信区间和bootstrap 偏差修正，MCMC。利用大量计算机计算（包括随机模拟）来进行统计推断的统计学分支叫做“计算统计”（computational statistics）。

随机数

- 设随机变量 X 有分布函数 $F(x)$，$\{X_i, i = 1, 2, \ldots\}$ 独立同分布 $F(x)$，则 $\{X_i, i = 1, 2, \ldots\}$ 的一次观测的数值 $\{x_i, i = 1, 2, \ldots\}$ 叫做分布 $F(x)$ 的随机数序列，简称**随机数**。
- 随机数是统计中一个重要的计算工具“随机模拟”的基本构成元素。
- 我们可以用物理方法得到一组真实的随机数，比如，反复抛掷硬币、骰子、正二十面体骰子，抽签、摇号，等等。这些方法得到的随机数质量好，但是数量不能满足随机模拟的需要。
- 另一种办法是预先生成大量的真实随机数存储起来，进行随机模拟时读取存储的随机数。这种方法的速度较低，已经被取代了。
- 现在进行随机模拟的主流方法是在计算机上用固定的算法实时地产生随机数，严格来说是**伪随机数**。

随机数

- 伪随机数是用计算机算法产生的随机数序列，其表现与真实的$F(x)$ 的独立同分布序列很难区分开来。
- 因为计算机算法的结果是固定的，所以伪随机数不是真正的随机数；
- 但是，好的伪随机数序列与真实随机数序列表现相同，很难区分，现在的计算机模拟都是使用伪随机数，我们把伪随机数也叫做随机数。
- **需要某种分布的随机数时，一般先生成均匀分布随机数，然后由均匀分布随机数再转换得到其它分布的随机数。产生均匀分布随机数的算法叫做均匀分布随机数发生器。**

如何产生伪随机数

生成随机数的逆变换方法是基于以下熟知的定理

Theorem 1 (Probability Integral Transformation). 若X为连续型随机变量, 其*cdf* 为F_X, 则 $U = F_X\ (X) \sim U(0,1)$.

Proof. 定义

$$F_X^{-1}(u) = \inf\{x : F_X(x) = u\}, \quad 0 < u < 1.$$

若随机变量$U \sim U(0,1)$, 则对所有$x \in \mathcal{R}$, 有

$$\begin{aligned} P(F_X^{-1}(U) \le x) &= P(\inf\{t : F_X(t) = U\} \le x) \\ &= P(U \le F_X(x)) = F_U(F_X(x)) = F_X(x). \end{aligned}$$

因此$F_X^{-1}(U)$和随机变量X同分布. □

迭代法产生伪随机数

从而, 若要产生X的一个随机观测x, 可以

1. 导出逆变换函数 $F_X^{-1}(u)$.
2. 从均匀分布$U(0,1)$中产生一个随机数u, 令$x = F_X^{-1}(u)$.

这种方法要求F的逆函数要容易求出.

例: 使用逆变换方法产生连续型密度$f(x) = 3x^2 I(0 < x < 1)$的随机观测值.

此处$F_X(x) = x^3, (0 < x < 1)$, 因此 $F_X^{-1}(u) = u^{1/3}$, 因此

均匀分布随机数的产生和周期性

现在常用的均匀分布随机数发生器有线性同余法、反馈位寄存器法以及随机数发生器的组合。

周期性

- 均匀分布随机数发生器生成的是 $\{0, 1, \ldots, M\}$ 或 $\{1, 2, \ldots, M\}$ 上离散取值的离散均匀分布，然后除以 M 或 $M+1$ 变成 $[0, 1]$ 内的值当作连续型均匀分布随机数，实际上是只取了有限个值。
- 因为取值个数有限，根据算法 $x_n = g(x_{n-1}, x_{n-2}, \ldots, x_{n-p})$ 可知序列一定在某个时间发生重复，使得序列发生重复的间隔 T 叫做随机数发生器的**周期**。好的随机数发生器可以保证 M 很大而且周期很长。

线性同余发生器

- 线性同余发生器，Linear congruential generator（LCG），是利用求余运算的随机数发生器。其递推公式为：

$$x_n = (ax_{n-1} + c) \pmod{M},\ n = 1, 2, \ldots$$

  - 这里等式右边的 $(ax_{n-1} + c) \pmod{M}$ 表示 $ax_{n-1} + c$ 除以 M 的余数，正整数 M 为除数，正整数 a 为乘数，非负整数 c 为增量。
  - 取某个非负整数 x_0 为初值可以向前递推，递推只需要序列中前一项。
  - 得到的序列 $\{x_n\}$ 为非负整数，$0 \leq x_n < M$。
  - 最后，令 $R_n = x_n/M$，则 $R_n \in [0, 1)$，把 $\{R_n\}$ 作为均匀分布随机数序列。
  - 这样的算法的基本思想是因为很大的整数前面的位数是重要的有效位数而后面若干位有一定随机性。
  - 如果取 $c = 0$，线性同余发生器称为**乘同余发生器**，
  - 如果取 $c > 0$，线性同余发生器称为**混合同余发生器**。

LCG的周期

- 因为线性同余法的递推算法仅依赖于前一项，序列元素取值只有 M 个可能取值，所以产生的序列 $x_0, x_1, x_2, \ldots$ 一定会重复。
- 若存在正整数 n 和 m 使得 $x_n = x_m (m < n)$，则必有 $x_{n+k} = x_{m+k}, k = 0, 1, 2, \ldots$，即 $x_n, x_{n+1}, x_{n+2}, \ldots$ 重复了 $x_m, x_{m+1}, x_{m+2}, \ldots$，称这样的 $n - m$ 的最小值 T 为此随机数发生器在初值 x_0 下的**周期**。
- 由序列取值的有限性可见 $T \leq M$。
    - 考虑线性同余发生器

$$x_n = 7x_{n-1} + 7 \pmod{10}.$$

    - 取初值 $x_0 = 7$,
    - 数列为: $(7, 6, 9, 0, 7, 6, 9, 0, 7, \ldots)$,
    - 周期为 $T = 4 < M = 10$。

LCG的周期

- 考虑线性同余发生器

$$x_n = 5x_{n-1} + 1 \pmod{10}$$

- 取初值 $x_0 = 1$。
- 数列为: $(1, 6, 1, 6, 1, \ldots)$,
- 显然周期 $T = 2$。

- 考虑线性同余发生器

$$x_n = 5x_{n-1} + 1 \pmod{8}$$

- 取初值 $x_0 = 1$。
- 数列为 $(1, 6, 7, 4, 5, 2, 3, 0, 1, 6, 7, \ldots)$,
- $T = 8 = M$ 达最大周期。

- 从例子发现 $T \leq M$ 且有的线性同余发生器可以达到最大周期 M，称为满周期。
- 满周期时，初值 x_0 取为 $0 \sim M-1$ 间的任意数都是一样的，$X_M = x_0$，序列从 X_M 开始重复。
- 如果发生器从某个初值不是满周期的，那么它从任何初值出发都不是满周期的，不同初值有可能得到不同序列。
- 比如随机数 $x_n = 7x_{n-1} + 7 \pmod{10}$, 从不同初值出发的序列可能为如 $7, 6, 9, 0, 7, \cdots$; $1, 4, 5, 2, 1, \cdots$; $3, 8, 3$。
- 不同的 M, a, c 选取方法得到不同的周期，适当选取 M, a, c 才能使得产生的随机数序列和真正的 U$[0, 1]$ 随机数表现接近。

其他随机数发生器--FSR

线性同余法的周期不可能超过2^L（L 为整数型尾数的位数），而且作为多维随机数相关性大，分布不均匀。

基于 Tausworthe(1965) 文章的 FSR 方法是一种全新的做法，对这些方面有改善。

FSR(反馈位移寄存器法) 按照某种递推法则生成一系列二进制数 $\alpha_1, \alpha_2, \ldots, \alpha_k, \ldots$，其中 α_k 由前面的若干个 $\{\alpha_i\}$ 线性组合并求除以 2 的余数产生:

递推公式为

$$\alpha_k = (c_p\alpha_{k-p} + c_{p-1}\alpha_{k-p+1} + \cdots + c_1\alpha_{k-1}) \pmod 2,\ k = 1, 2, \ldots \tag{2.1}$$

线性组合系数 $\{c_i\}$ 只取 0, 1，这样的递推可以利用程序语言中的整数二进制运算快速实现。

其他随机数发生器—组合发生器

随机数设计中比较困难的是独立性和多维的分布。可以考虑把两个或若干个发生器组合利用，可以比单个发生器由更长的周期和更好的随机性。

- 利用三个 16 位运算的素数模乘同余发生器:

$$\begin{aligned} U_n &= 171U_{n-1} \pmod{30269} \\ V_n &= 172V_{n-1} \pmod{30307} \\ W_n &= 170W_{n-1} \pmod{30323} \end{aligned}$$

- 作线性组合并求余:

$$R_n = (U_n/30269 + V_n/30307 + W_n/30323) \pmod 1$$

- 这个组合发生器的周期约有 7×10^{12} 长, 超过 $2^{32} \approx 4 \times 10^9$。

- MacLaren 和 Marsaglia(1965) 设计了组合同余法，组合两个同余发生器，一个用来“搅乱”次序。
- 设有两个同余发生器 A 和 B。用 A 产生 m 个随机数 (如 $m = 128$)，存放在数组 $T = (t_1, t_2, \ldots, t_m)$. 需要产生 x_n 时，从 B 中生成一个随机下标 $j \in \{1, 2, \ldots, m\}$，取 $x_n = t_j$，但从 A 再生成一个新随机数 y 代替 T 中的 t_j, 如此重复。
- 这样组合可以增强随机性，加大周期 (可超过 2^L)。也可以只使用一个发生器，用 x_{n-1} 来选择下标。

随机数的检验

我们要解决一个自己的模拟问题时，还是要反复确认所用的随机数在自己的问题中是好的，没有明显缺陷的。

- 比如，一个经典的称为 RANDU 的随机数发生器用在一维积分时效果很好，但连续三个一组作为三维均匀分布则很不均匀。
- 为了验证随机数的效果，最好是找一个和自己的问题很接近但是有已知答案的问题，用随机模拟计算并考察其精度。
- 对均匀分布随机数发生器产生的序列 $\{R_n, n = 1, 2, \ldots, N\}$ 可以进行各种各样的检验以确认其均匀性与独立性。

K-s检验

- 对随机数列 $\{R_n, n = 1, 2, \ldots, N\}$ 计算

$$\bar{R} = \frac{1}{N}\sum_{n=1}^{N} R_n,\ \overline{R^2} = \frac{1}{N}\sum_{n=1}^{N} R_n^2,\ S^2 = \frac{1}{N}\sum_{n=1}^{N}(R_n - \frac{1}{2})^2$$

则 $N \to \infty$ 时 $\bar{R}$、$\overline{R^2}$ 和 S^2 均渐近服从正态分布，可以用 Z 检验法检验这三个统计量与理论期望值的偏离程度。

- 把 $[0, 1]$ 等分成 k 段，用拟合优度卡方检验法检验 $\{R_n, n = 1, 2, \ldots, N\}$ 落在每一段的取值概率是否近似为 $1/k$。

- 用 Kolmogorov-Smirnov 检验法进行拟合优度检验，看 $\{R_n, n = 1, 2, \ldots, N\}$ 是否与 U[0,1] 分布相符。

- 把 $\{R_n, n = 1, 2, \ldots, N\}$ 每 d 个组合在一起成为 $\mathbb{R}^d$ 向量，把超立方体 $[0, 1]^d$ 每一维均分为 k 份，得到 k^d 个子集，用卡方检验法检验组合得到的 $\mathbb{R}^d$ 向量落在每个子集的概率是否近似为 k^{-d}。

- 把 $\{R_n, n = 1, 2, \ldots, N\}$ 看作时间序列样本，计算其样本自相关函数列 $\{\hat{\rho}_j, j = 1, 2, \ldots\}$，在 $\{R_n, n = 1, 2, \ldots\}$ 独立同分布情况下 $\{\sqrt{N}\hat{\rho}_j, j = 1, 2, \ldots, N\}$ 应该渐近服从独立的标准正态分布，可以据此进行白噪声检验。

- 把 $\{R_n\}$ 离散化为 $y_n = \text{floor}(kR_n), n = 1, 2, \ldots, N$，令 $\xi_n = y_n$，$\eta_n = y_{n+b}$(b 为正整数)，$n = 1, 2, \ldots, N - b$，用列联表检验法检验 ξ_n 和 η_n 的独立性。

其他检验

- 游程检验。把序列 $\{R_n, n = 1, 2, \ldots, N\}$ 按顺序切分为不同段，每一段中的数值都是递增的，这样的一段叫做一个**上升游程**，如 (0.855), $(0.108, 0.226)$, $(0.032, 0.123)$, $(0.055, 0.545, 0.642, 0.870)$, $\ldots$。在相邻的两个上升游程中前一个游程的最后一个值大于后一个游程的第一个值。在 $\{R_n, n = 1, 2, \ldots, \}$ 独立同分布条件下游程长度的理论分布可以得知，然后可以比较实际游程长度与独立同分布条件下期望长度。

- 扑克检验。把 $\{R_n\}$ 离散化为 $y_n = \text{floor}(8R_n), n = 1, 2, \ldots, N$, 然后每连续的 8 个作为一组，计数每组内不同数字的个数（$1 \sim 8$ 个）。在 $\{R_n\}$ 独立同均匀分布条件下每组内不同数字个数的理论分布概率可以计算出来，然后用卡方检验法检验实际观测与理论概率是否相符。

- 配套检验。$\{R_n\}$ 离散化为 $y_n = \text{floor}(kR_n)$(k 为正整数), $n = 1, 2, \ldots, N$, 然后顺序抽取 $\{y_n, n = 1, 2, \ldots, N\}$ 直到 $\{y_n\}$ 的可能取值 $\{0, 1, \ldots, k-1\}$ 都出现过为止，记录需要抽取的 $\{y_n\}$ 的个数 L，反复抽取并记录配齐数字需要抽取的值的个数 $l_j, j = 1, 2, \ldots$。在 $\{R_n\}$ 独立同 U(0,1) 分布条件下这样的 L 分布可以得到，可以计算 $\{l_j\}$ 的平均值并用渐近正态分布检验观测均值与理论均值的差异大小，或直接用卡方检验法比较 $\{l_j\}$ 的样本频数与理论期望值。

- 正负连检验。令 $y_n = R_n - \frac{1}{2}$，把连续的 $\{y_n\}$ 的正值分为一段，把连续的 $\{y_n\}$ 的负值分为一段，每段叫做一个“连”，连长 L 的分布概率为 $P(L = k) = 2^{-k}, k = 1, 2, \ldots$，可以用卡方检验法检验 L 的分布；总连数 T 满足 $ET = \frac{n+1}{2}$, $\text{Var}(T) = \frac{n-1}{4}$, 可以用 Z 检验法检验 T 的值与理论期望的差距。

- 升降连检验。计算 $y_n = R_n - R_{n-1}, n = 2, 3, \ldots, N$，把连续的正值的 $\{y_n\}$ 分为一段叫做一个上升连，把连续的负值的 $\{y_n\}$ 分为一段叫做一个下降连，可以用卡方检验法比较连的长度与 $\{R_n\}$ 独立同分布假设下的理论分布，或用 Z 检验法比较总连数与理论期望值的差距。

从U(0,1)到其它概率分布的随机数

U(0,1)的均匀分布的随机数，是生成其他概率分布随机数的基础，下面我们主要介绍两种将U(0,1)随机数转换为其他分布的随机数的方法。

1.逆变换方法
(Inverse Transform Method)
2.舍取方法
(Acceptance-Rejection Method)

逆变换方法

❑离散型场合

设X为一离散型随机变量, 其可能取值记为

$$\cdots < x_{i-1} < x_i < x_{i+1} < \cdots$$

定义

$$F_X^{-1}(u) = \inf\{x : F_X(x) \geq u\}$$

则逆变换为$F_X^{-1}(u) = x_i$, 其中$F_X(x_{i-1}) < u \leq F_X(x_i)$. 从而产生一个随机数$x$的方式为

1. 从均匀分布$U(0,1)$中产生一个随机数u
2. 取$x = x_i$, 若$F_X(x_{i-1}) < u \leq F_X(x_i)$.

从而, 若要产生X的一个随机观测x, 可以

1. 导出逆变换函数 $F_X^{-1}(u)$.
2. 从均匀分布$U(0,1)$中产生一个随机数u, 令$x = F_X^{-1}(u)$.

这种方法要求F的逆函数要容易求出.

例: 使用逆变换方法产生连续型密度$f(x) = 3x^2 I(0 < x < 1)$的随机观测值.

此处$F_X(x) = x^3, (0 < x < 1)$, 因此 $F_X^{-1}(u) = u^{1/3}$, 因此

- **定理 2.2.1** 设 X 为连续型随机变量，取值于区间 (a,b)(可包括 $\pm\infty$ 和端点), X 的密度在 (a,b) 上取正值，X 的分布函数为 $F(x)$，$U \sim \mathrm{U}(0,1)$，则 $Y = F^{-1}(U) \sim F(\cdot)$。
- **证明:**

$$P(Y \leq y) = P(F^{-1}(U) \leq y) = P(U \leq F(y)) = F(y), \ \forall y \in (a,b)$$

逆变换法

- **例 2.2.7：泊松分布随机数** 设离散型随机变量 X 分布为

$$p_k = P(X = k) = e^{-\lambda}\frac{\lambda^k}{k!},\ k = 0, 1, 2, \ldots (\lambda > 0)$$

称 X 服从泊松分布，记为 $X \sim \text{Poisson}(\lambda)$。

- 易见

$$p_{k+1} = \frac{\lambda}{k+1}p_k,\ k = 0, 1, 2, \ldots$$

- 所以在利用定理 2.2.2 构造泊松随机数时，可以递推计算

$$F(k+1) = F(k) + p_{k+1} = F(k) + \frac{\lambda}{k+1}p_k,\ k = 0, 1, 2, \ldots \qquad (2.3)$$

- 算法如下：

生成 $U \sim \text{U}(0, 1)$
$k \leftarrow 0,\ p \leftarrow e^{-\lambda},\ F \leftarrow p$
while$(U > F)$ {
  $p \leftarrow \frac{\lambda}{k+1}p,\ \ F \leftarrow F + p$
  $k \leftarrow k + 1$
}
$X \leftarrow k$

泊松随机数的改进

- 这样的算法先判断是否令 X 取 0，不成立则再判断是否令 X 取 1，如此重复，判断的次数为 X 的值加 1，所以平均判断次数需要 $EX+1=\lambda+1$ 次。
- 当 λ 较小时这种方法效率不错；
- 如果 λ 比较大，这时 $p(k)=e^{-\lambda}\frac{\lambda^k}{k!}$ 在最接近于 λ 的两个整数之一上最大，按照例 2.2.2 的讨论，应该先判断这两个整数及其邻近的点。
- 令 $K=\mathrm{floor}(\lambda)$, 用式 (2.3) 先递推计算出 $F(K)$ 和 $F(K+1)$，然后判断 $F(K)<U\le F(K+1)$ 是否成立，如果成立就令 $X=K+1$;
- 否则，如果 $U\le F(K)$ 则依次判断是否应取 X 为 $K,K-1,\ldots,0$, 在这个过程中可以反向递推计算各 $F(k)$;
- 如果 $U>F(K+1)$ 则依次判断是否应取 X 为 $K+2,K+3,\ldots$，这个过程中仍然用式 (2.3) 递推计算各 $F(k)$。
- 这样的改进的算法需要的判断次数约为 $|X-\lambda|+1$，所以平均判断次数为 $E|X-\lambda|+1$。
- 因为 λ 较大时泊松分布渐近正态分布 $\mathrm{N}(\lambda,\lambda)$, 所以平均判断次数约为

$$\begin{aligned}1+E|X-\lambda| =& 1+\sqrt{\lambda}E\left|\frac{X-\lambda}{\sqrt{\lambda}}\right| \\ \approx& 1+\sqrt{\lambda}E|Z| \quad (\text{其中 } Z\sim \mathrm{N}(0,1)) \\ =& 1+0.798\sqrt{\lambda}\end{aligned}$$

- 当 λ 很大时平均判断次数由 $O(\lambda)$ 降低到了 $O(\sqrt{\lambda})$。

逆变换法—正态分布

- **定理 2.2.5** 设 U_1, U_2 独立且都服从 U(0,1)，

$$\begin{cases} X = \sqrt{-2\ln U_1}\cos(2\pi U_2), \\ Y = \sqrt{-2\ln U_1}\sin(2\pi U_2) \end{cases} \tag{2.6}$$

则 X, Y 独立且都服从标准正态分布。式 (2.6) 叫做 Box-Muller 变换。

证明： 从 (U_1, U_2) 到 (X, Y) 的变换的逆变换为

$$\begin{cases} u_1 = \exp\left\{-\frac{1}{2}(x^2+y^2)\right\} \\ u_2 = \frac{1}{2\pi}\arctan\frac{y}{x} \end{cases}$$

逆变换的 Jacobi 行列式为

$$J = \begin{vmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} \end{vmatrix} = -\frac{1}{2\pi}\exp\left\{-\frac{1}{2}(x^2+y^2)\right\}$$

所以 (X, Y) 的联合密度为

$$f(x,y) = 1\cdot\frac{1}{2\pi}\exp\left\{-\frac{1}{2}(x^2+y^2)\right\} = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}\cdot\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2}$$

即 X, Y 独立且分别服从标准正态分布。

舍选法(Acceptance-Rejection method)

假设X和Y是随机变量, 其概率函数分别为f和g. 满足

$$\frac{f(t)}{g(t)} \leq c, \quad \forall \ t \ \ s.t. \ \ f(t) > 0$$

则舍选法(Acceptance-Rejection Method)可以用来生成X的随机数.

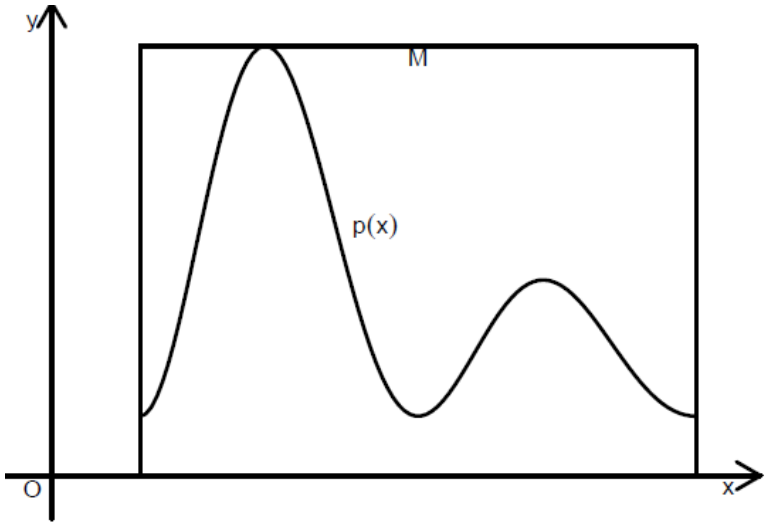
1. 找一个可以方便生成随机数的随机变量Y, 其概率函数g满足$f(t)/g(t) \leq c, \forall \ t \ s.t. \ f(t) > 0$.
2. 从g中产生一个随机数y.
3. 从均匀分布$U(0,1)$中产生一个随机数u.
4. 若$u < f(y)/(cg(y))$, 则接受$x = y$, 否则拒绝y. 重复2-4, 直至产生给定个数的x.

舍选法

- 设随机变量 Z 的分布函数很难求反函数 $F^{-1}(u)$。
- 如果 Z 取值于有限区间 $[a,b]$ 且密度 $p(x)$ 有上界 M:

$$p(x) \le M,\ \forall x \in [a,b]$$

- 则可以用如下算法生成 Z 的随机数:

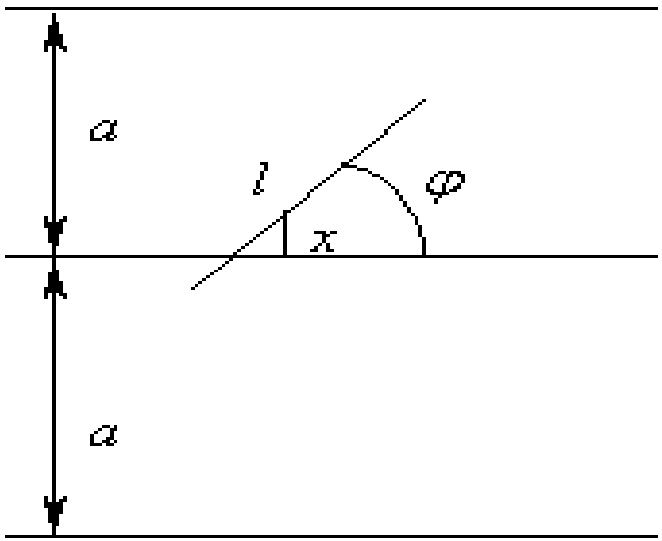


until $(Y \le p(X))$ {
  生成 $U_1, U_2 \sim \mathrm{U}(0,1)$
  取 $X \leftarrow a + (b-a)U_1$, $Y \leftarrow MU_2$
}
输出 $Z \leftarrow X$

- 这种方法叫做舍选法 I。
- 如图 2.1 所示，每次循环生成的 (X,Y) 实际上构成了矩形 $[a,b]\times[0,M]$ 上的二维均匀分布，循环退出条件为 $Y \le p(X)$，循环退出时 (X,Y) 的值落入了 $p(x)$ 曲线下方，这时 (X,Y) 服从曲边梯形 $\{(x,y) : a \le x \le b, 0 \le y \le p(x)\}$ 上的二维均匀分布，所以循环退出时 $Z = X$ 在 x 附近取值的概率是与 $p(x)$ 成正比的。

从Buffon 投针问题谈起

Buffon 投针问题:平面上画很多平行线，间距为 a．向此平面投掷长为l($l<$ a) 的针，求此针与任一平行线相交的概率 p。



随机投针可以理解成针的中心点与最近的平行线的距离X是均匀地分布在区间 $[0, a/2]$ 上的r. v.，针与平行线的夹角Φ是均匀地分布在区间 $[0, \pi]$ 上的r. v.，且X与Φ相互独立，于是针与平行线相交的充要条件为 $X \le \frac{l}{2}\sin\Phi$，即相交 $A=\{\omega: X \le \frac{l}{2}\sin\Phi\}$.

Buffon 投针问题

于是有：

$$p = P(X \le \frac{l}{2}\sin\Phi) = \int_0^{\pi} \int_0^{\frac{l}{2}\sin\varphi} \frac{2}{\pi a} dx d\varphi = \frac{2l}{a\pi} \Rightarrow \pi = \frac{2l}{ap}$$

若我们独立重复地作 n 次投针试验，记 $\mu_n(A)$ 为 A 发生的次数。 $f_n(A)$ 为 A 在 n 次中出现的频率。假如我们取 $f_n(A)$ 作为 $p = P(A)$ 的估计，即 $\hat{p} = f_n(A)$。

然后取 $\hat{\pi} = \frac{2l}{af_n(A)}$ 作为 π 的估计。根据大数定律，当 $n \to \infty$ 时， $\hat{p} = f_n(A) \xrightarrow{a.s.} p.$

从而有 $\hat{\pi} = \frac{2l}{af_n(A)} \xrightarrow{P} \pi$。这样可以用随机试验的方法求得 π 的估计。历史上有如下的试验结果。

试验者	时间(年)	针长	投针次数	相交次数	π 的估计值
Wolf	1850	0.80	5000	2532	3.15956
Smith	1855	0.60	3204	1218	3.15665
Fox	1884	0.75	1030	489	3.15951
Lazzarini	1925	0.83	3408	1808	3.14159292

数值积分问题

计算积分 $\alpha = \int_0^1 f(x)dx = Ef(X)$

我们可以将此积分看成 $f(x)$ 的数学期望。其中 $X \sim U[0,1]$ (均匀分布)。于是可以将上式积分看作是f(X)的数学期望. 若$\{U_k, 1 \le k \le n\}$为$i.i.d.U_k \sim U[0,1]$.

则可以取$\alpha_n = \frac{1}{n}\sum_{i=1}^{n} f(U_i)$作为$\alpha$的估计，

由大数定律，可以保证收敛性，即：

$$\alpha_n \to \alpha \quad with\ probability\ as\ n \to \infty$$

这表明可以用随机模拟的方法计算积分。

贝叶斯公式

$$P(B_i \mid A) = \frac{P(A \mid B_i)P(A)}{\sum P(A \mid B_i)P(B_i)}$$

$$f(x_i \mid y) = \frac{f(y \mid x_i) f(x_i)}{\int_{\infty} f(y \mid x_i) f(x_i)}$$

➢ 收敛性

- 由前面介绍可知，蒙特卡罗方法是由随机变量X的简单子样X_1，X_2，…，X_N的算术平均值：

$$\overline{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i$$

- 作为所求解的近似值。由大数定律可知，
- 如X_1，X_2，…，X_N独立同分布，且具有有限期望值（$E(X)<\infty$），则

$$P\left(\lim_{N\to\infty} \overline{X}_N = E(X)\right) = 1$$

- 即随机变量X的简单子样的算术平均值 $\overline{X}_N$ ，当子样数N充分大时，以概率1收敛于它的期望值E(X)。

➢ 误差

- 蒙特卡罗方法的近似值与真值的误差问题，概率论的中心极限定理给出了答案。该定理指出，如果随机变量序列X_1，X_2，…，X_N独立同分布，且具有有限非零的方差σ^2，即

$$0 \neq \sigma^2 = \int (x - E(X))^2 f(x) dx < \infty$$

- $f(X)$是X的分布密度函数。则

$$\lim_{N \to \infty} P\left(\frac{\sqrt{N}}{\sigma} \left| \overline{X}_N - E(X) \right| < x \right) = \frac{1}{\sqrt{2\pi}} \int_{-x}^{x} e^{-t^2/2} dt$$

- 当N充分大时，有如下的近似式

$$P\left(\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}}\right) \approx \frac{2}{\sqrt{2\pi}} \int_0^{\lambda_\alpha} e^{-t^2/2} dt = 1-\alpha$$

- 其中α称为置信度，1－α称为置信水平。

- 这表明，不等式 $\left|\overline{X}_N - E(X)\right| < \frac{\lambda_\alpha \sigma}{\sqrt{N}}$ 近似地以概率1－α成立，且误差收敛速度的阶为 $O(N^{-1/2})$ 。

- 通常，蒙特卡罗方法的误差ε定义为

$$\varepsilon = \frac{\lambda_\alpha \sigma}{\sqrt{N}}$$

- 上式中λ_α 与置信度α是一一对应的，根据问题的要求确定出置信水平后，就可以确定出 λ_α 。

- 下面给出几个常用的α与的数值：

α	0.5	0.05	0.003
λ_α	0.6745	1.96	3

- 关于蒙特卡罗方法的误差需说明两点：第一，蒙特卡罗方法的误差为概率误差，这与其他数值计算方法是有区别的。第二，误差中的均方差σ是未知的，必须使用其估计值

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2 - (\frac{1}{N}\sum_{i=1}^{N} X_i)^2}$$

- 来代替，在计算所求量的同时，可计算出 $\hat{\sigma}$ 。

Monte Carlo积分的优点

与一般的数值积分方法比较，Monte Carlo方法具有以下优点：

1. 一般的数值方法很难推广到高维积分的情形，而Monte Carlo方法很容易推广到高维情形

2. 一般的数值积分方法的收敛阶为 $O(n^{-2/d})$，而由中心极限定理可以保证 Monte Carlo 方法的收敛阶为 $O(n^{-1/2})$ 。此收敛阶与维数无关，且在高维时明显优于一般的数值方法。

python™

谢谢Q/A