蒙特卡洛模拟——随机数的生成

朱连华

南京信息工程大学数学与统计学院

2019-11-20

随机数生成

- 均匀分布

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**。

- 非均匀分布

逆变换法：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
$Y = F^{-1}(U) \sim F(\cdot)$。

- 复合抽样
- 筛选抽样

随机数生成

其它变换方案

- 卡方分布
- F 分布
- T 分布
- Beta 分布

其它变换方案

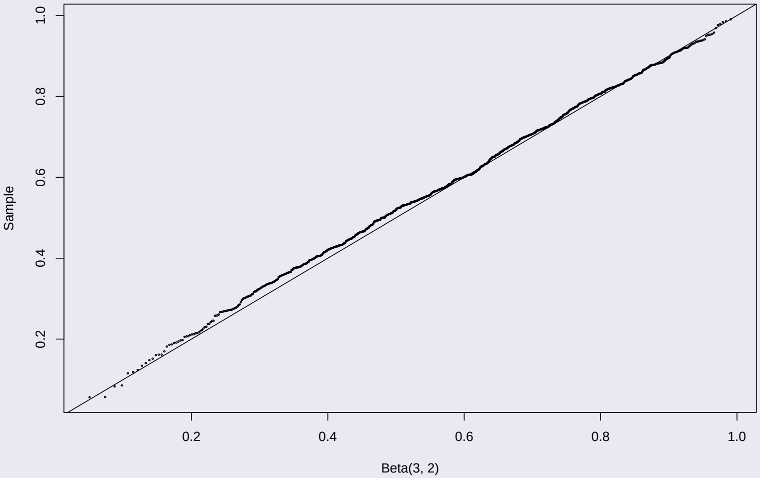
Beta 分布：Gamma(r,lamdba),Gamma(s,lamda)

```r
n <- 1000
a <- 3
b <- 2
u <- rgamma(n, shape=a, rate=1)
v <- rgamma(n, shape=b, rate=1)
x <- u / (u + v)

q <- qbeta(ppoints(n), a, b)
qqplot(q, x, cex=0.25, xlab="Beta(3, 2)", ylab="Sample")
abline(0, 1)
```

其它变换方案

Beta(r,s)



求和变换与混合

- $Z = X + Y$
- $F_Z(z) = pF_X(z) + (1-p)F_Y(z)$

卡方分布—求和变换

```
n <- 1000
nu <- 2
X <- matrix(rnorm(n*nu), n, nu)^2 #matrix of sq. normals
#sum the squared normals across each row: method 1
y <- rowSums(X)
#method 2
y <- apply(X, MARGIN=1, FUN=sum)  #a vector length n
mean(y)
mean(y^2)
```

卡方分布—求和变换

```r
n <- 1000
nu <- 2
X <- matrix(rnorm(n*nu), n, nu)^2 #matrix of sq. normals
#sum the squared normals across each row: method 1
y <- rowSums(X)
#method 2
y <- apply(X, MARGIN=1, FUN=sum)  #a vector length n
mean(y)
```

```
## [1] 2.0714
```

```r
mean(y^2)
```

```
## [1] 8.983285
```

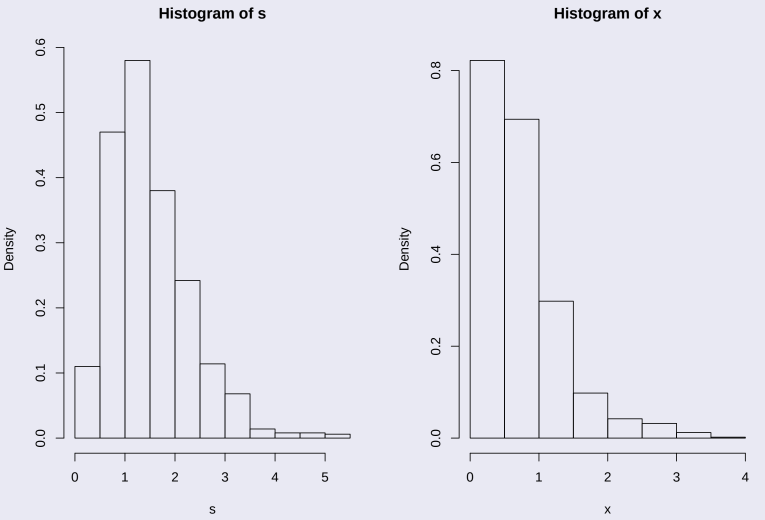
混合分布—gamma 分布

```r
n <- 1000
x1 <- rgamma(n, 2, 2)
x2 <- rgamma(n, 2, 4)
s <- x1 + x2              #the convolution
u <- runif(n)
k <- as.integer(u > 0.5)  #vector of 0's and 1's
x <- k * x1 + (1-k) * x2  #the mixture

par(mfcol=c(1,2))         #two graphs per page
hist(s, prob=TRUE)
hist(x, prob=TRUE)
par(mfcol=c(1,1))         #restore display
```

其它变换方案

混合分布—gamma 分布



混合变量—多个 gamma 分布

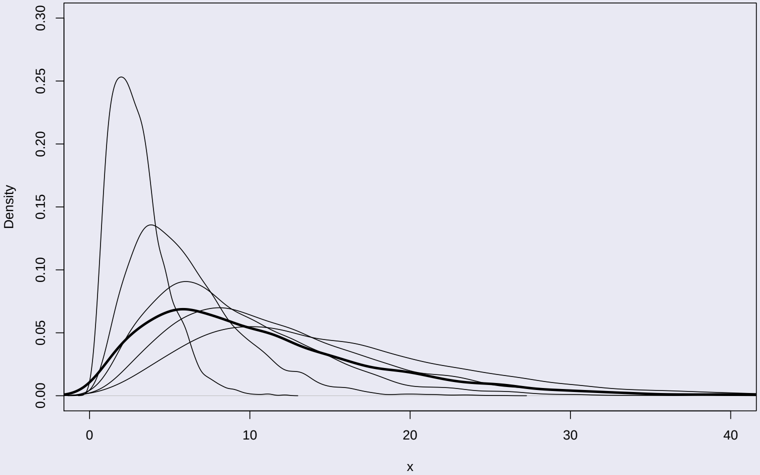
$$X_j \sim Gamma(r=3, \lambda=1/j), \theta_i = j/15, F_Z(z) = \sum_{j=1}^{5} \theta_i F_{X_j}(z)$$

```r
n <- 5000
k <- sample(1:5, size=n, replace=TRUE, prob=(1:5)/15)
rate <- 1/k
x <- rgamma(n, shape=3, rate=rate)

#plot the density of the mixture
#with the densities of the components
plot(density(x), xlim=c(0,40), ylim=c(0,.3),
    lwd=3, xlab="x", main="")
for (i in 1:5)
    lines(density(rgamma(n, 3, 1/i)))
```

其它变换方案

混合变量—多个 gamma 分布



混合分布—多个 gamma 分布

$X_j \sim Gamma(3, \lambda_j), \lambda = (1, 1.5, 2, 2.5, 3)$

$\theta_i = (0.1, 0.2, 0.2, 0.3, 0.2), F_Z(z) = \sum_{j=1}^{5} \theta_i F_{X_j}(z)$

```
n <- 5000;p <- c(.1,.2,.2,.3,.2)
lambda <- c(1,1.5,2,2.5,3)
k <- sample(1:5, size=n, replace=TRUE, prob=p)
rate <- lambda[k];
x <- rgamma(n, shape=3, rate=rate)
k[1:8];    rate[1:8]
```

```
## [1] 4 2 3 3 3 2 5 4
```

```
## [1] 2.5 1.5 2.0 2.0 2.0 1.5 3.0 2.5
```

求和变换与混合

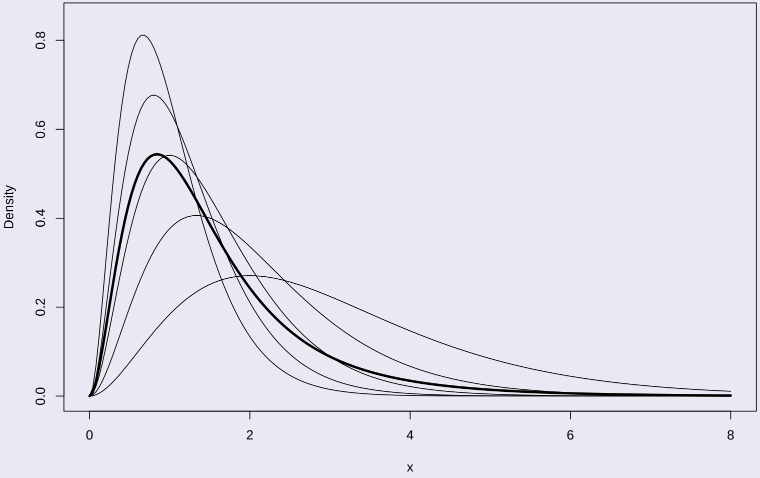
混合分布—多个 gamma 分布

$$f_Z(z) = \sum_{j=1}^{5} \theta_i f_{X_j}(z)$$

```
f <- function(x, lambda, theta) {
    #density of the mixture at the point x
    sum(dgamma(x, 3, lambda) * theta)   }
p <- c(.1,.2,.2,.3,.2)
lambda <- c(1,1.5,2,2.5,3)
x <- seq(0, 8, length=200)
dim(x) <- length(x)
y <- apply(x, 1, f, lambda=lambda, theta=p)
plot(x, y, type="l", ylim=c(0,.85), lwd=3, ylab="Density")
for (j in 1:5) {
    y <- apply(x, 1, dgamma, shape=3, rate=lambda[j])
    lines(x, y)
}
```

求和变换与混合

混合分布—多个 gamma 分布



多维正态随机变量的生成

二元正态分布

$$f(x_1,x_2)=\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}$$

$$\exp\left\{\frac{-1}{2(1-\rho^2)}\left[\frac{(x_1-\mu_1)^2}{\sigma_1^2}-2\rho\frac{(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_1}+\frac{(x_2-\mu_2)^2}{\sigma_2^2}\right]\right\}$$

多维正态分布

设随机向量 $\boldsymbol{X}=(X_1,X_2,\ldots,X_n)^T$ 服从多元正态分布 $N(\boldsymbol{\mu},\Sigma)$，联合密度函数为:

$$f(\boldsymbol{x})=(2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\},\boldsymbol{x}\in R^n$$

正定矩阵 Σ 有 Cholesky 分解 $\Sigma=CC^T$，其中 C 为下三角矩阵。

多维正态随机变量的生成

多维标准正态分布

- 设 $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_d)^T$ 服从 d 元标准正态分布 $N(\mathbf{0}, I_d)(I_d$ 表示单位阵)
- 则 $\boldsymbol{X} = \boldsymbol{\mu} + C\boldsymbol{Z}$ 服从 $N(\boldsymbol{\mu}, \Sigma)$ 分布
- $X = ZQ +_J \mu^T$, 其中 $QQ^T = \sum, Z = Z_{ij}$ 为 $n \times d$ 矩阵，Z_{ij} 相互独立且服从 $N(0, 1)$ 分布。

```
Z=matrix(rnorm(n*d),nrow=n,ncol=d)
X=Z%*%Q+matrix(mu,n,d,byrow=T)
```

- 谱分解
- Choleski 分解
- 奇异值分解（SVD）

多维正态随机变量的生成（谱分解法）

- $X = ZQ + J\mu^T$, 其中 $QQ^T = \sum, Z = Z_{ij}$ 为 $n \times d$ 矩阵，Z_{ij} 相互独立且服从 $N(0,1)$ 分布。
- $\sum^{1/2} = P\Lambda^{1/2}P^{-1} = P\Lambda^{1/2}P^T$, 其中 Λ 和 P 分别为 $\sum$ 的特征根和对应特征向量
- 谱分解 $X \sim N(\mu, \sum)$, $\mu = 0$, $\Sigma = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix}$

```r
# mean and covariance parameters
mu <- c(0, 0)
Sigma <- matrix(c(1, .9, .9, 1), nrow = 2, ncol = 2)
```

多维正态随机变量的生成（谱分解法）

- $X = ZQ + J\mu^T$，其中 $QQ^T = \sum, Z = Z_{ij}$ 为 $n \times d$ 矩阵，Z_{ij} 相互独立且服从 $N(0,1)$ 分布。

```r
rmvn.eigen <-
function(n, mu, Sigma) {
    # generate n random vectors from MVN(mu, Sigma)
    d <- length(mu)
    ev <- eigen(Sigma, symmetric = TRUE)
    lambda <- ev$values
    V <- ev$vectors
    R <- V %*% diag(sqrt(lambda)) %*% t(V)
    Z <- matrix(rnorm(n*d), nrow = n, ncol = d)
    X <- Z %*% R + matrix(mu, n, d, byrow = TRUE)
    X
}
```

多维正态随机变量的生成（谱分解法）

- $X = ZQ +_J \mu^T$, 其中 $QQ^T = \sum, Z = Z_{ij}$ 为 $n \times d$ 矩阵，Z_{ij} 相互独立且服从 $N(0,1)$ 分布。
- 谱分解 $X \sim N(\mu, \sum)$, $\mu = 0$, $\Sigma = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix}$

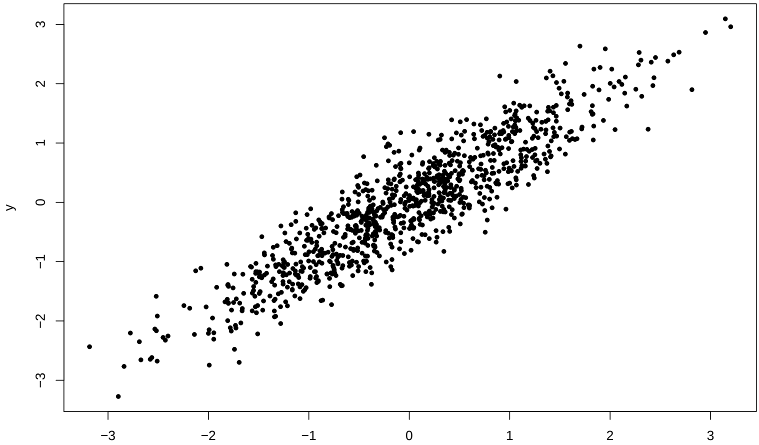
```r
X <- rmvn.eigen(1000, mu, Sigma)
plot(X, xlab = "x", ylab = "y", pch = 20)
print(colMeans(X))
print(cor(X))
```

```
## [1] -0.0009884311 -0.0085352025
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.9119063
## [2,] 0.9119063 1.0000000
```

多维正态随机变量的生成（谱分解法）

- 谱分解 $X \sim N(\mu, \sum)$, $\mu = 0$, $\Sigma = \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix}$



多维正态随机变量的生成方法比较

```r
library(MASS)
library(mvtnorm)
n <- 100          #sample size
d <- 30           #dimension
N <- 2000         #iterations
mu <- numeric(d)
mu <- numeric(d)
set.seed(100)
system.time(for (i in 1:N)
    rmvn.eigen(n, mu, cov(matrix(rnorm(n*d), n, d))))
set.seed(100)
system.time(for (i in 1:N)
    rmvn.svd(n, mu, cov(matrix(rnorm(n*d), n, d))))
set.seed(100)
system.time(for (i in 1:N)
    rmvn.Choleski(n, mu, cov(matrix(rnorm(n*d), n, d))))
```

多维正态随机变量的生成方法比较

```
set.seed(100)
system.time(for (i in 1:N)
    mvrnorm(n, mu, cov(matrix(rnorm(n*d), n, d))))
set.seed(100)
system.time(for (i in 1:N)
    rmvnorm(n, mu, cov(matrix(rnorm(n*d), n, d))))
set.seed(100)
system.time(for (i in 1:N)
    cov(matrix(rnorm(n*d), n, d)))
detach(package:MASS)
detach(package:mvtnorm)
```

多元混合正态分布

Example Multivariate normal mixture

```
library(MASS)  #for mvrnorm
#ineffecient version loc.mix.0 with loops
loc.mix.0 <- function(n, p, mu1, mu2, Sigma) {
    #generate sample from BVN location mixture
    X <- matrix(0, n, 2)

    for (i in 1:n) {
        k <- rbinom(1, size = 1, prob = p)
        if (k)
            X[i,] <- mvrnorm(1, mu = mu1, Sigma) else
            X[i,] <- mvrnorm(1, mu = mu2, Sigma)
        }
    return(X)
}
```

Example Multivariate normal mixture

```
#more efficient version
loc.mix <- function(n, p, mu1, mu2, Sigma) {
    #generate sample from BVN location mixture
    n1 <- rbinom(1, size = n, prob = p)
    n2 <- n - n1
    x1 <- mvrnorm(n1, mu = mu1, Sigma)
    x2 <- mvrnorm(n2, mu = mu2, Sigma)
    X <- rbind(x1, x2)            #combine the samples
    return(X[sample(1:n), ])      #mix them
}
```

Example Multivariate normal mixture

```
#more efficient version
x <- loc.mix(1000, .5, rep(0, 4), 2:5, Sigma = diag(4))
r <- range(x) * 1.2
par(mfrow = c(2, 2))
for (i in 1:4)
    hist(x[ , i], xlim = r, ylim = c(0, .3), freq = FALSE,
    main = "", breaks = seq(-5, 10, .5))
detach(package:MASS)
par(mfrow = c(1, 1))
```

多元混合正态分布