《计算机模拟》

第4讲 – 指定分布随机数

胡贤良

浙江大学数学科学学院

本讲内容

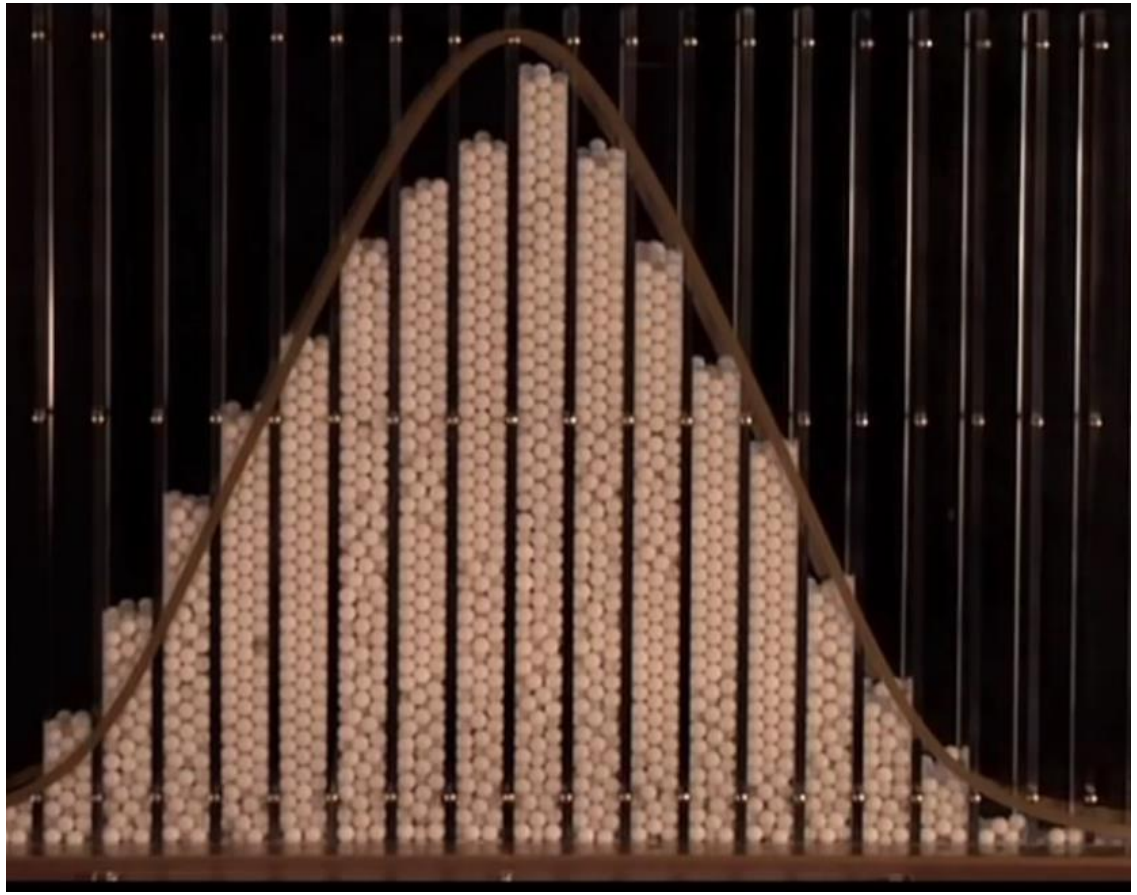
1.典型分布随机数/抽样

2.指定分布随机数/抽样

① 逆变换法

② 拒绝-接受法

3.高维随机数/抽样



1. 典型离散/连续分布随机数

二项分布（Binomial）

在n次伯努利试验中，若以变量X表示事件A出现的次数，则X的取值为$\{0,\cdots,n\}$,其相应的分布（记作$X\sim B(n,p)$）为

$$P(X=k)=\binom{n}{k}\mu^{k}(1-\mu)^{n-k},$$

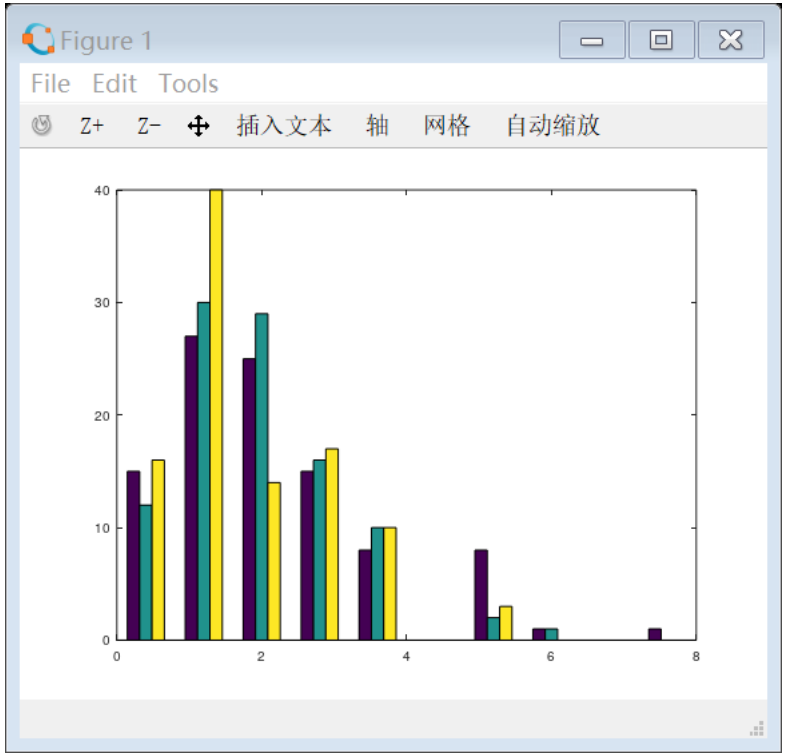
$$k=0,1,2,\cdots,n.$$

```
% Bernoulli test. p is the rate of true.
function y = Bernoulli_test(p)
    if (p > 1 || p < 0)
        fprintf("error!");
        return;
    end
    xi = rand();
    if (xi < p)
        y = 1;  % true.
    else
        y = 0;  % false.
    end
end
```

```
function y = binomial_test(n, p)
    y = 0;
    for i = 1:n
        y = y + Bernoulli_test(p);
    end
end
```

- API: binocdf/binopdf/binornd

```
>> p = binopdf(1, 100,  0.01)
>> B = binornd(100,  0.02, 100, 3);
>> hist(B);
```



正态分布随机数

normrnd(a, b)：给出服从期望为 a 且标准差为 b 的正态分布的一个随机数；

normrnd(a, b, M, N) 或者 **normrnd**(a, b, [M, N])：给出由服从期望为 a 且标准差为 b 的正态分布的随机数组成的 $M \times N$ 矩阵。

例 4.10 我们用 300 个均匀分布的随机变量之和来近似标准正态分布。设 $R_i \sim U\left(-\frac{1}{2}, \frac{1}{2}\right)$, $i = 1, \cdots, 300$, 其期望为 0 且方差为 $\sigma^2 = \frac{1}{12}$。

```
clear all, clf
m = 300;   n = 10000;   nbins = 100;
R = unifrnd(-0.5, 0.5, [m, n]);
Q = sum(R, 1)/5;      % 由累加生成的随机数据
w = (max(Q)-min(Q))/nbins;
[Y, X] = hist(Q, nbins);
Y = Y/n/w;
t = -3.5:0.05:3.5;
Z = 1/sqrt(2*pi)*exp(-(t.^2)/2);      % 标准正态分布的密度函数
hold on
bar(X, Y, 0.5)
plot(t, Z, 'r')
hold off
MSE = norm(Y - normpdf(X))/sqrt(nbins)      % 均方误差
```

上面第四行“除以 5”是中心极限定理公式 (2.52) 中的分母：$\sqrt{300} \times \sqrt{\frac{1}{12}} = 5$。显示在图 4.6 中的结果表明，累加的变量近似为正态分布，其均方误差是：MSE = 0.0118。

例 4.11 在金融市场中，股票的回报 (股价的变化率) 被假设服从期望为 0 且方差为 σ^2 的正态分布，这里方差被称为**波动率**(volatility)。我们令 $s(t)$ 表示在 t 时刻的股价，那么将回报写为

$$R(t) = \log(s(t)) - \log(s(t-1)) \approx \frac{s(t) - s(t-1)}{s(t-1)}。$$

因为，对于所有 t 都有：$R(t) \sim N(0, \sigma^2)$。所以

$$s(t) = s(t-1)\mathrm{e}^R, \quad R \sim N(0, \sigma^2)。$$

于是，我们可以用正态分布的随机数来模拟股价在一年内 (252 个工作日) 的变化轨迹：

```
n = 252;  s = zeros(1, n);
s(1) = 100;   sigma = 0.15;      % 设置初始股价与波动率
R = normrnd(0, sigma, 1, n);
for t = 2:n
  s(t) = s(t-1)*exp(R(t));
end
plot(s)
xlim([0,252])
xlabel('时间'), ylabel('价格')
```

对数正态分布

用 Matlab 生成对数正态分布的随机数的命令是：

lognrnd(a, b)：给出服从对数正态分布的一个随机数，其中与该分布对应的正态分布的期望为 a 且标准差为 b；

lognrnd(a, b, M, N) 或者 **lognrnd**(a, b, [M, N])：给出由服从对数正态分布的随机数组成的 $M \times N$ 矩阵，参数意义同上。

$$f(x)=\begin{cases}\dfrac{1}{x\sigma\sqrt{2\pi}}e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}, & x>0,\\ 0, & x\leqslant 0。\end{cases}$$

```
t = 20;
s0 = 100;    r=0.05/360;  sigma = 0.03;     % 设置初始股价、无风险利率与波动率
[mu, v] = lognstat(log(s0)+r*t, sigma*sqrt(t))
pr = 1 - logncdf((1+0.15)*s0, log(s0)+r*t, sigma*sqrt(t))
n = 500000;     % 模拟50万次
s = lognrnd(log(s0)+r*t, sigma*sqrt(t), 1, n);
muhat = mean(s)      % 以模拟结果的平均作为期望
pshat = sum(s>(1+0.15)*s0)/n      % 以模拟结果计算事件的频率
```

计算结果给出：那时的期望价格为 101.1847，上涨超过 15% 的概率为 0.1536，而模拟得到的相应结果分别是 101.1967 和 0.1533；两者是很接近的。

```
help lognstat
```

```
'lognstat' is a function from the file /usr/share/octave/packages/statistics

 -- Function File: [M, V] = lognstat (MU, SIGMA)
     Compute mean and variance of the lognormal distribution.

     Arguments
     ---------

        * MU is the first parameter of the lognormal distribution

        * SIGMA is the second parameter of the lognormal distribution.
          SIGMA must be positive or zero
     MU and SIGMA must be of common size or one of them must be scalar

     Return values
     -------------

        * M is the mean of the lognormal distribution

        * V is the variance of the lognormal distribution

     Examples
     --------

          mu = 0:0.2:1;
          sigma = 0.2:0.2:1.2;
          [m, v] = lognstat (mu, sigma)

          [m, v] = lognstat (0, sigma)

     References
     ----------

       1. Wendy L. Martinez and Angel R. Martinez.  'Computational
          Statistics Handbook with MATLAB'. Appendix E, pages 547-557,
          Chapman & Hall/CRC, 2001.

       2. Athanasios Papoulis.  'Probability, Random Variables, and
          Stochastic Processes'.  McGraw-Hill, New York, second edition,
          1984.
```

指数分布

当人们考察相继发生事件的时间间隔，或者事件的存续时间，往往发现这些时间的长度是随机的，指数分布常被用来刻画它们。指数概率密度函数为

$$f(x)=\frac{1}{a}\mathrm{e}^{-\frac{x}{a}},\quad x\geqslant 0, \tag{4.6}$$

其中参数 $a>0$ 是分布的期望或标准差，而其倒数 a^{-1} 则反映在单位时间内发生时间的次数，即发生率。

Matlab 提供的生成指数概率密度函数和累积分布函数的命令分别是：

exppdf(X, a)：给出概率密度函数在 X 各个点上的值；

expcdf(X, a)：给出累积分布函数在 X 各个点上的值；

exprnd(a)：生成服从参数为 a 的指数分布的随机数；

exprnd(a, [M, N])：生成由服从指数分布的随机数所组成的 $M\times N$ 矩阵 (分布参数

χ^2 分布

定义： 设随机变量$X_1, X_2, \ldots, X_n$相互独立，

$$X_i \sim N(0,1) \quad (i=1,2,\cdots,n)$$

则称

$$\chi^2 = \sum_{i=1}^{n} X_i^2 \qquad (1)$$

服从自由度为n的χ^2 分布，也记.为$\chi^2(n)$.

性质:

1. $E(\chi^2) = n, Var(\chi^2) = 2n;$
2. 设$Y_i \sim \chi^2(n_i), i = 1,2$ 且相互独立，则

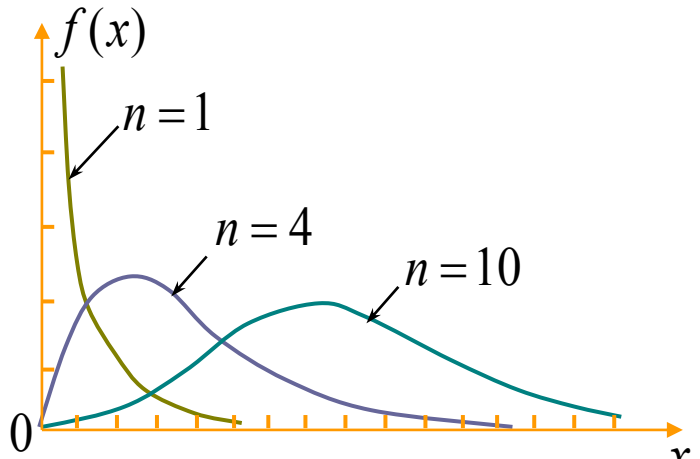
$$Y_1 + Y_2 \sim \chi^2(n_1 + n_2)$$

称为χ^2分布可加性。

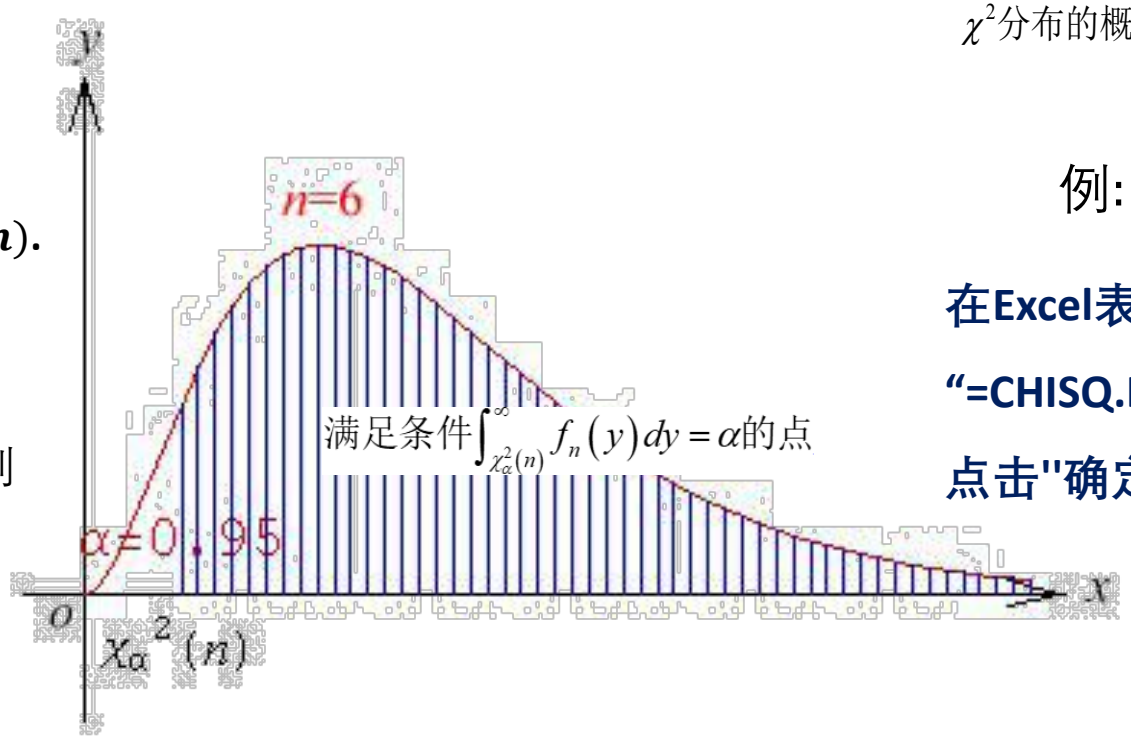
$\chi^2(n)$分布的概率密度函数为：

$$f(y) = \begin{cases} \dfrac{1}{2\Gamma(n/2)}\left(\dfrac{y}{2}\right)^{\frac{n}{2}-1} e^{-\frac{y}{2}}, & y > 0, \\ 0, & y \le 0, \end{cases}$$

其中，$\Gamma(\alpha) = \int_0^{+\infty} x^{\alpha-1} e^{-x} dx.$



χ^2分布的概率密度函数



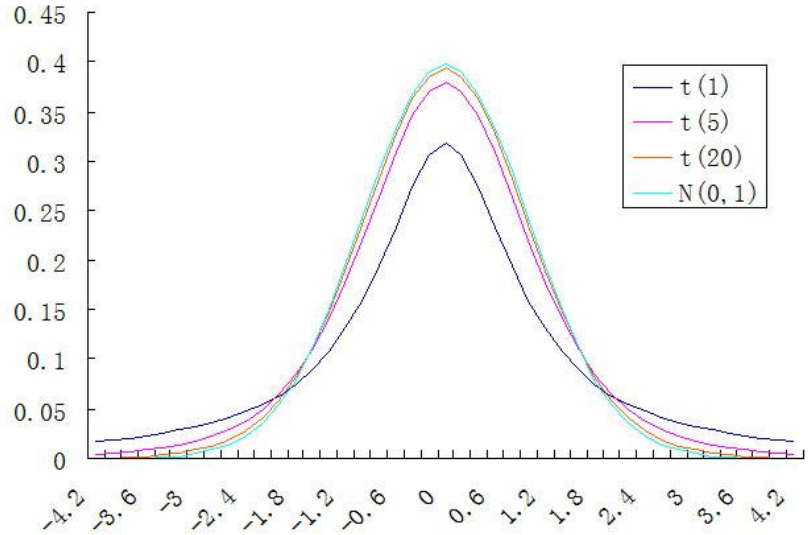
例: 求$\chi^2_{0.1}(25)$

在Excel表单的任一单元格输入

“=CHISQ.INV.RT (0.1,25)”

点击"确定" 即出现 "34.382".

t分布(Student's distribution, 学生氏分布, W . S . Gusset, 1908年)



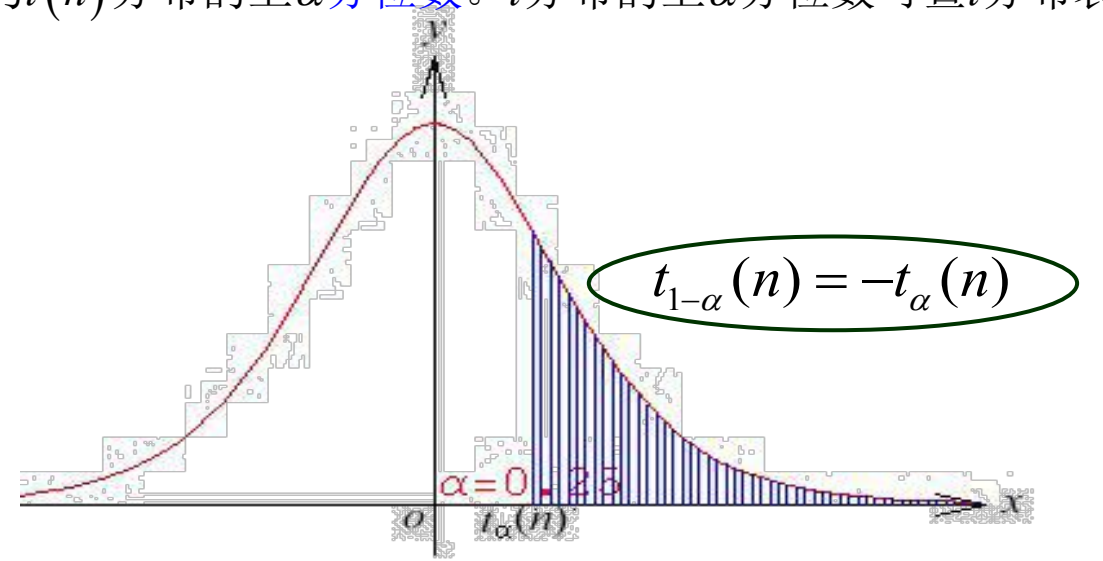
设 $X \sim N(0,1)$，$Y \sim \chi^2(n)$，并且假设 X, Y 相互独立，

则称 $T = \dfrac{X}{\sqrt{Y/n}}$ 服从自由度为 n 的 t 分布. 记为 $T \sim t(n)$

$t(n)$分布的概率密度函数为：

$$f(t) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}}, -\infty < t < +\infty$$

对给定的α, $0<\alpha<1$, 称满足条件$\int_{t_\alpha(n)}^{\infty} f(t,n)dt = \alpha$的点$t_\alpha(n)$为$t(n)$分布的上$\alpha$分位数。$t$分布的上$\alpha$分位数可查$t$分布表



$$t_{1-\alpha}(n) = -t_\alpha(n)$$

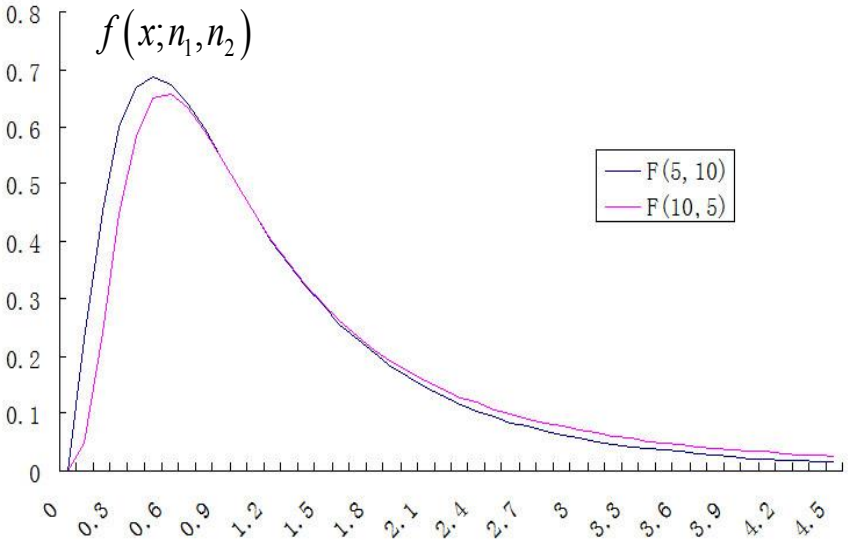
例 利用Excel求 $t_{0.05}(25)$.

在Excel表单的任一单元格输入

"=T.INV (1–0.05, 25)" 或 "=T.INV.2T (0.05*2, 25)"；

点击"确定" 即在单元格中出现 "1.708".

F分布



设$X\sim\chi^2(n_1),Y\sim\chi^2(n_2)$，且$X,Y$独立，则

称随机变量$F=\dfrac{X/n_1}{Y/n_2}$服从自由度(n_1,n_2)的F分布，

记为$F\sim F(n_1,n_2)$.

其中，n_1称为第一自由度，n_2称为第二自由度.

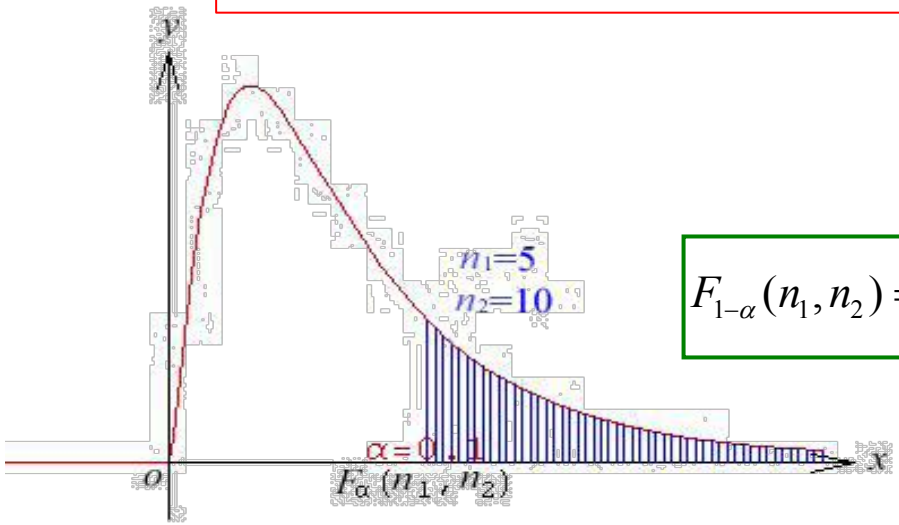
性质：$F\sim F(n_1,n_2)$,则$F^{-1}\sim F(n_2,n_1)$.

对于给定的α，$0<\alpha<1$,称满足条件

$$\int_{F_\alpha(n_1,n_2)}^{\infty} f(x;n_1,n_2)\,dx=\alpha$$

的点$F_\alpha(n_1,n_2)$为$F(n_1,n_2)$分布的上α分位数.

$F_\alpha(n_1,n_2)$的值可查F分布表.



$$F_{1-\alpha}(n_1,n_2)=[F_\alpha(n_2,n_1)]^{-1}$$

例: 利用Excel求 $F_{0.1}(9,10)$.

在Excel表单的任一单元格输入

“=F.INV.RT (0.1, 9, 10)”或 “=F.INV (1–0.1, 9, 10)”；

点击“确定” 即在单元格中出现 “2.347”.

典型分布/随机数的Octave实现汇总

1. 均匀(uniform, unidpdf/unidrnd)分布

   >> K = unidrnd (6, 1, 10); % (NMAX, M, N);

   >> hist(K, [1,2,3,4,5,6]); % 1:6

2. 二项(binomial, binopdf/binornd)分布

   某机床次品率0.01，求100件产品中恰有1件次品的概率

   >> p = **binopdf**(1, 100, 0.01)

3. 泊松(Poisson, poisspdf/poissrnd)分布:

$$P(k) = e^{-\lambda}\frac{\lambda^k}{k!}, \qquad k = 0,1,2,\cdots$$

>> bar(poisspdf(0:15, 5));

1. 均匀(uniform,unifpdf, **unirnd**)分布:

   >> unifrnd(-.5, .5, 3, 5) % 3x5 rand in (-0.5,0.5)

   >> U = **rand**(1, 10000); % (0,1)均匀分布

2. 正态(normal, normpdf, **normrnd**)分布

   >> W = **randn**(1,10000); % 标准正态分布

   >> hist(W, 50);

3. 指数(exponial, exppdf, exprnd)分布

$$f(x) = \frac{1}{\alpha}e^{-\frac{x}{\alpha}}, \qquad (x \geq 0)$$

   >> expf = @(x,alpha) exp(-x/alpha)/alpha;

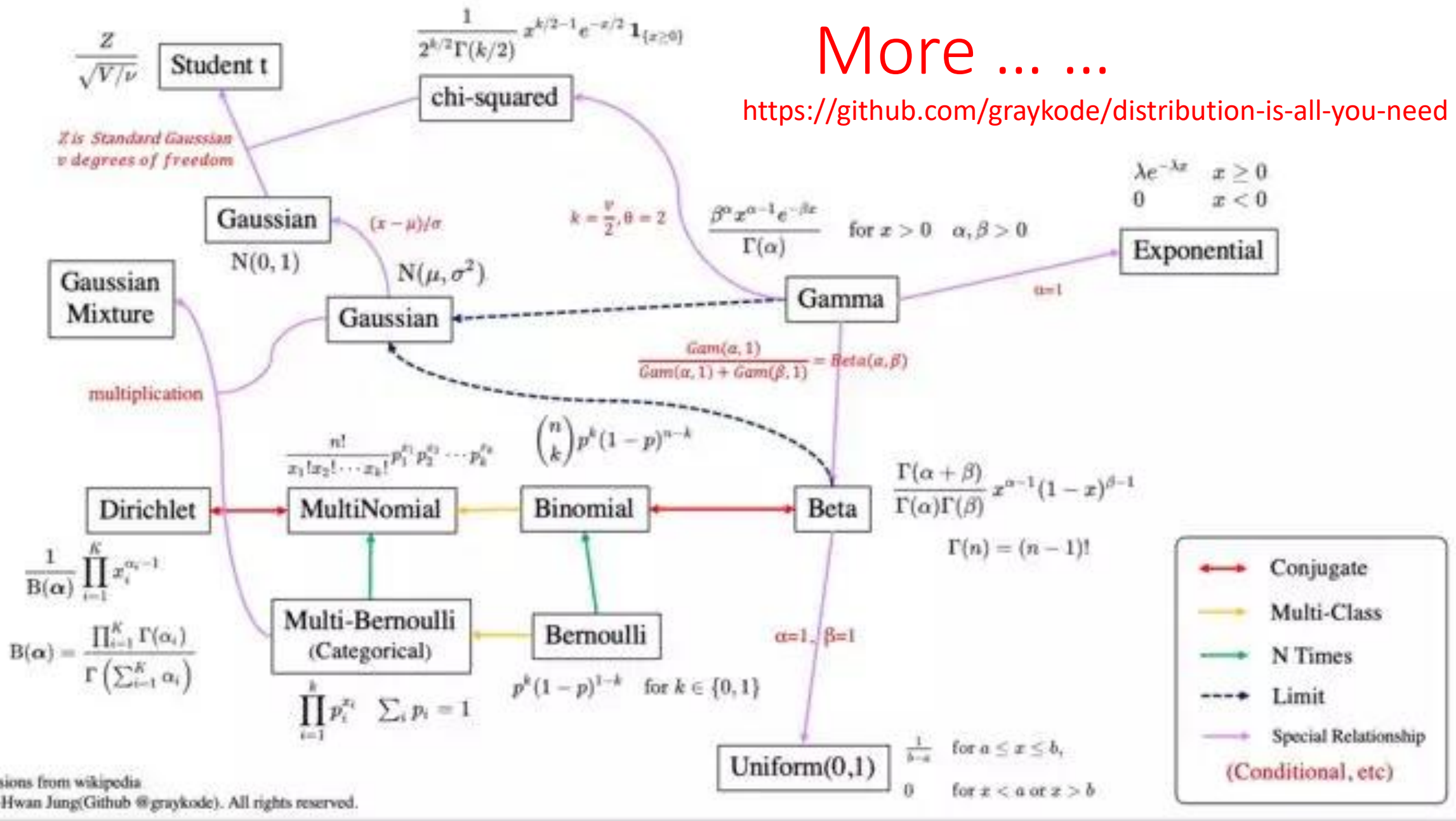
   >> t = (0:0.05:10)';

   >> u1 = expf(t, 1); u2 = expf(t, 2); u3 = expf(t, 3);

   >> plot(t, [u1, u2, u3], '—'); title('exppdf');

   >> legend('\alpha=1', ' \alpha=2', ' \alpha=3');

Relationship of distribution probability focused on Deep Learning



Copyright 2019 Tae-Hwan Jung(Github @graykode). All rights reserved.

2.生成指定分布的随机数

抽样方法：逆变换法、拒绝-接受法

问题:

若所需的随机数所满足的概率分布（概率密度函数）的解析表达式

- 未知

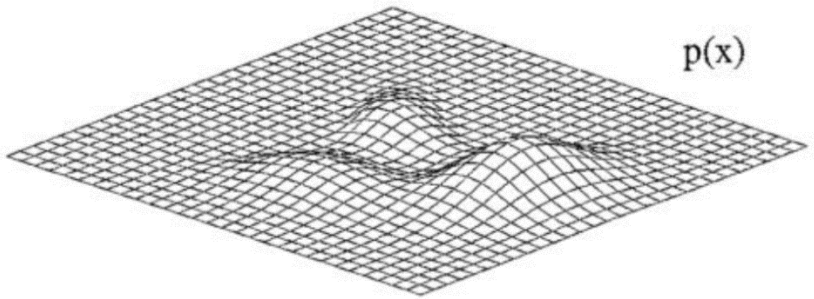
或

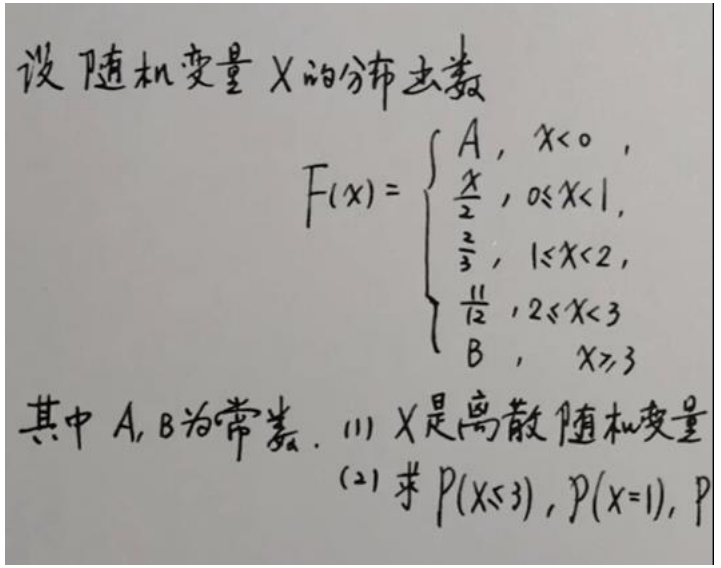
- 不完全已知

或

- 十分复杂

时，该如何获得随机数?





本文将用非参数经验 Bayes 方法研究 σ^2 未知的情况下正态分布位置参数的 EB 检验问题，并得到了其收敛速度

考虑如下正态分布

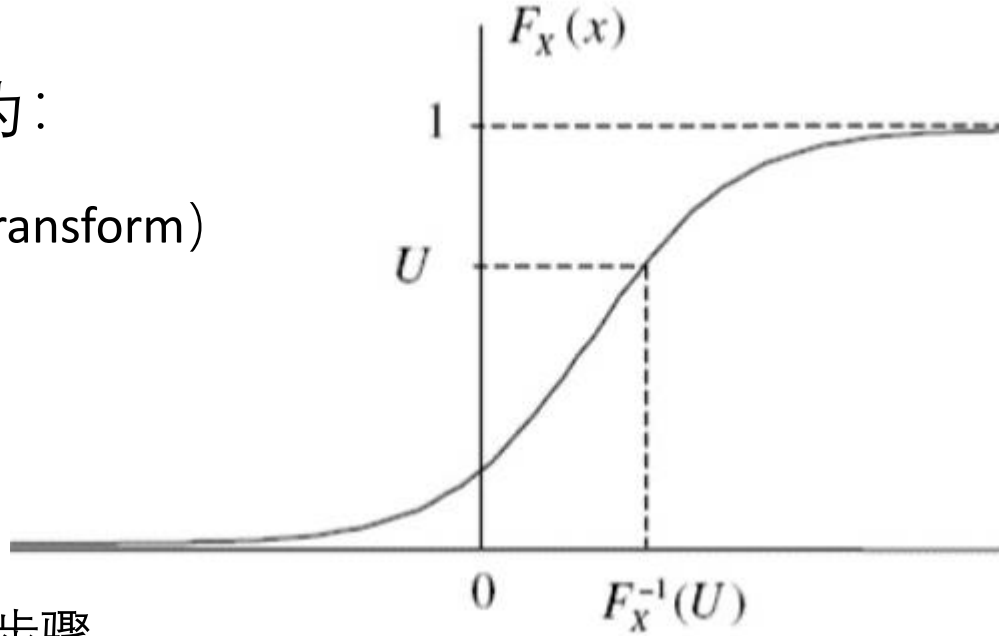
$$f(x/\theta,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-(x-\theta)^2/(2\sigma^2)\right]$$

其中 θ 和 σ 分别是位置参数和尺度参数，$\theta\in\Omega=(-\infty,+\infty)$，$\Omega$ 为位置参数空间，$\sigma>0$.

1. 逆变换法(inverse transformation method)

是一类产生伪随机数的基本方法，也被称为：

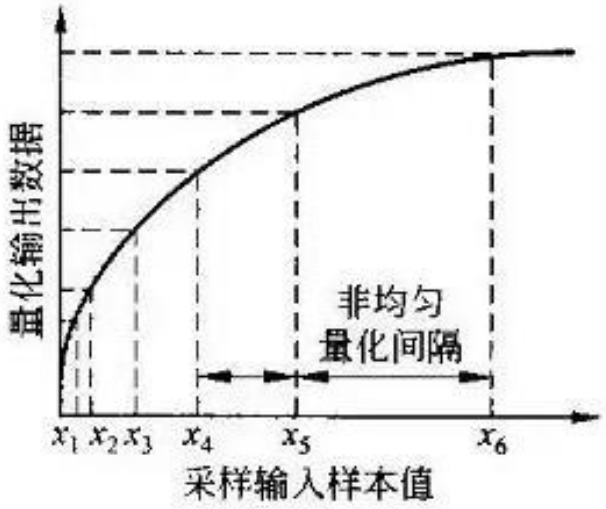
- **逆概率积分变换**（inverse probability integral transform）
- **斯米尔诺夫变换**（**Smirnov** transform）等



算法描述:

设$F(x)$是某个特定的一维概率分布函数，可用如下步骤生成服从该分布的随机数：

1. 写出该分布函数的反函数$\mathrm{F}^{-1}(u)$;
2. 生成随机数$u \sim U(0,1)$;
3. 计算$x = \mathrm{F}^{-1}(u)$，所得即为满足特定分布的随机数



理论解释: 随机变量的变换

记一个连续随机变量X，设其概率密度函数和分布函数分别为f_X和F_X; 另有严格单调递增的可微函数 $\varphi: R \to R$.

若令$U = \varphi(X)$也是一个连续随机变量，那么

$$F_Y(u) := P\{U \le u\} = P\{\varphi(X) \le u\}$$
$$= P\{X \le \varphi^{-1}(u)\} := F_X(\varphi^{-1}(u)) \qquad (*)$$

X与U直接的概率密度函数关系为

$$f_U(u) = \frac{dF_X(\varphi^{-1}(u))}{du} = \frac{dF_X}{dx}\frac{d\varphi^{-1}(u)}{du} = \frac{1}{\varphi'(\varphi^{-1}(u))} f_X(\varphi^{-1}(u))$$

逆变换法的导出

特别地，取均匀分布U[0,1]，则对应的分布:

$$F_U(u) = \begin{cases} 1, & u > 1, \\ u, & 0 \le u \le 1, \\ 0, & u < 0. \end{cases}$$

设$G(x)$为已知分布函数, 且$X = G^{-1}(U)$, 则随机变量X的分布函数为:

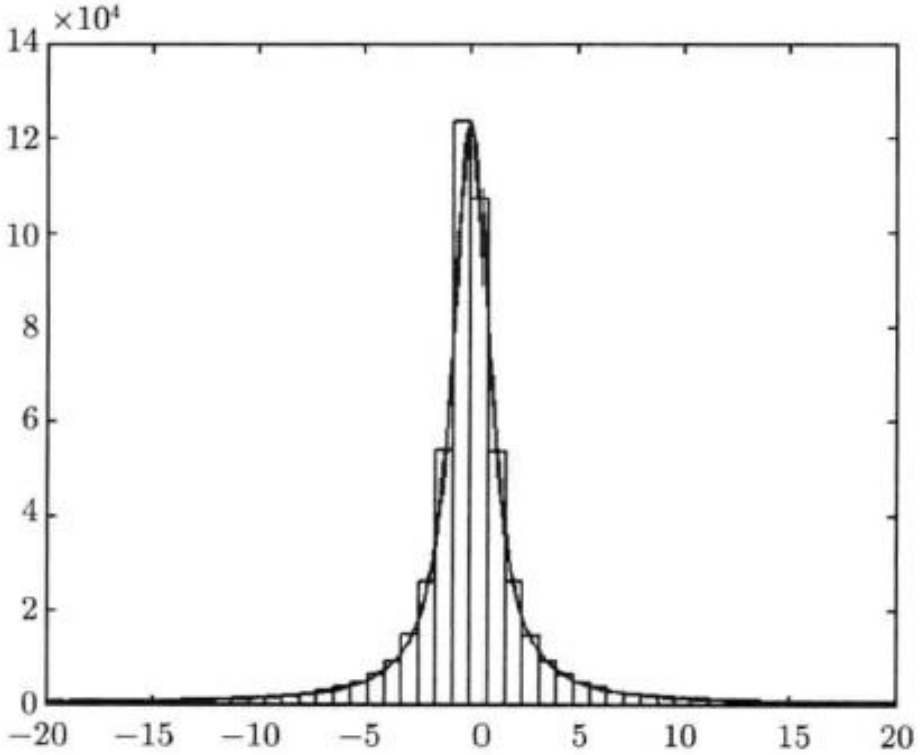
$$F_X(x) = F_U\left((G^{-1})^{-1}(u)\right) = G(x).$$

- 上述结论表明：只要对均匀分布的随机变量(数)做”**逆**”变换，便可获得指定分布函数G的随机变量(数)!

例4-1：柯西分布

- 也称为洛伦兹分布或Breit-Wigner分布
- 概率密度函数：

$$f(x) = \frac{1}{\pi(1+x^2)}$$




```
N= 50000;
X = zeros(1,N); count=0;
for i = 1:n
   U = unifrnd(0,1);
    tmp = tan(pi*(U - 0.5));
    if tmp >-20 && tmp < 20
          count  = count + 1;
          X(count) = tmp;
    end
end
hist(X, 60);
t = -20:0.1:20;
y = 1./(pi*(1 + t.*t));
scale = count * (40 / 60);
hold on;
plot(t, scale*y, ‘r’);
hold off;
```

```
N = 500000;


U = unifrnd(0, 1, 1, N);    %  They are
tmp = tan(pi * (U - 0.5)); %   Enough!


X = tmp((tmp > -20) & (tmp < 20));
count = length(X);



hist(X, 60);
 t = -20:0.1:20;
y = 1./(pi * (1 + t.*t));
scale = count * (40 / 60);
hold on;
plot(t, scale * y, 'r');
hold off;
```

逆变换法的局限性

1. 反函数求值失败导致程序出错、此外，还需注意排除0和1
2. 要求分布函数必须严格单调递增（反函数存在性、CDF）
3. 若反函数过于复杂(tan,sin,log,exp计算代价?)，会导致计算过慢!
4. 对于无法写出反函数解析表达式的情形，则无法计算!

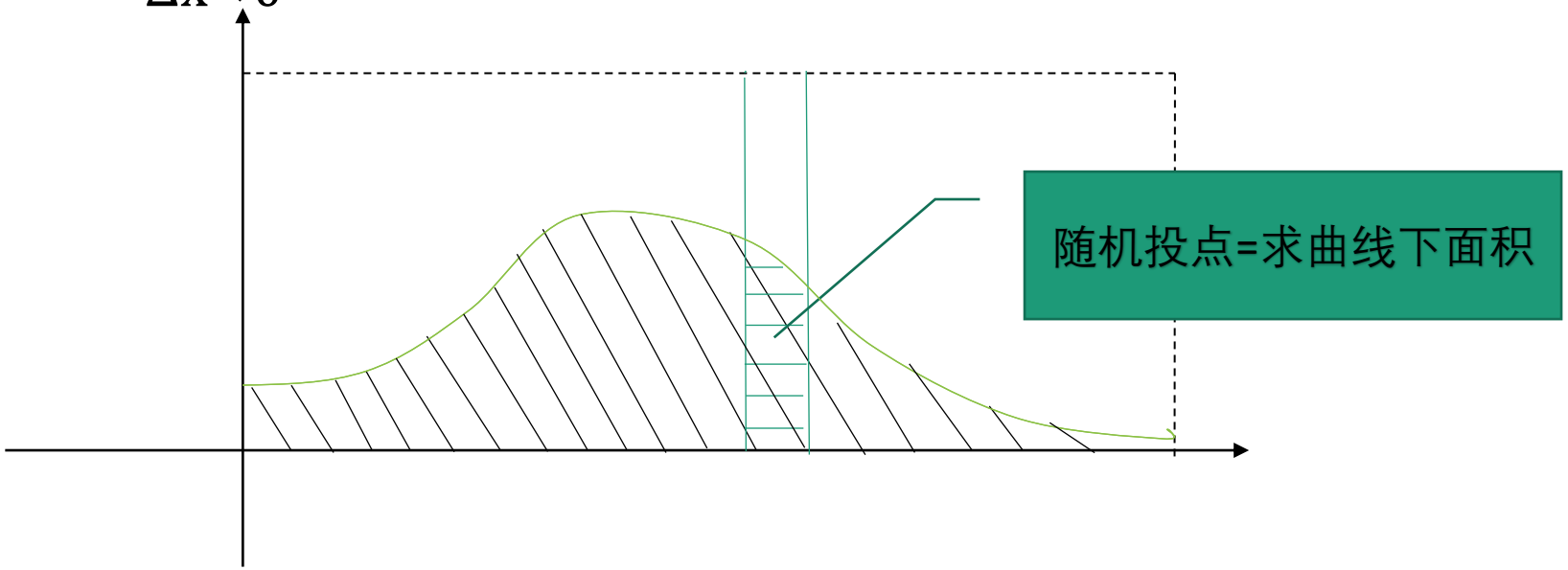
讨论：对概率密度函数$f(x)$的理解

- “函数”的观点

$$f(x) := P(X = x)$$

- “微元”的观点

$$f(x) \cdot \Delta x := \lim_{\Delta x \to 0} P(x \le X < x + \Delta x) \rightarrow f(x) \cdot dx := P(X = x)$$



2. 接受-拒绝(Acceptance-Rejection)方法

1. 选择简单函数$g(x)$，满足$f(x) \le Mg(x)$，如图所示：
2. 根据$g(x)$生成“建议随机数” y
3. 生成**均匀分布**随机数 u
4. 计算接受准则

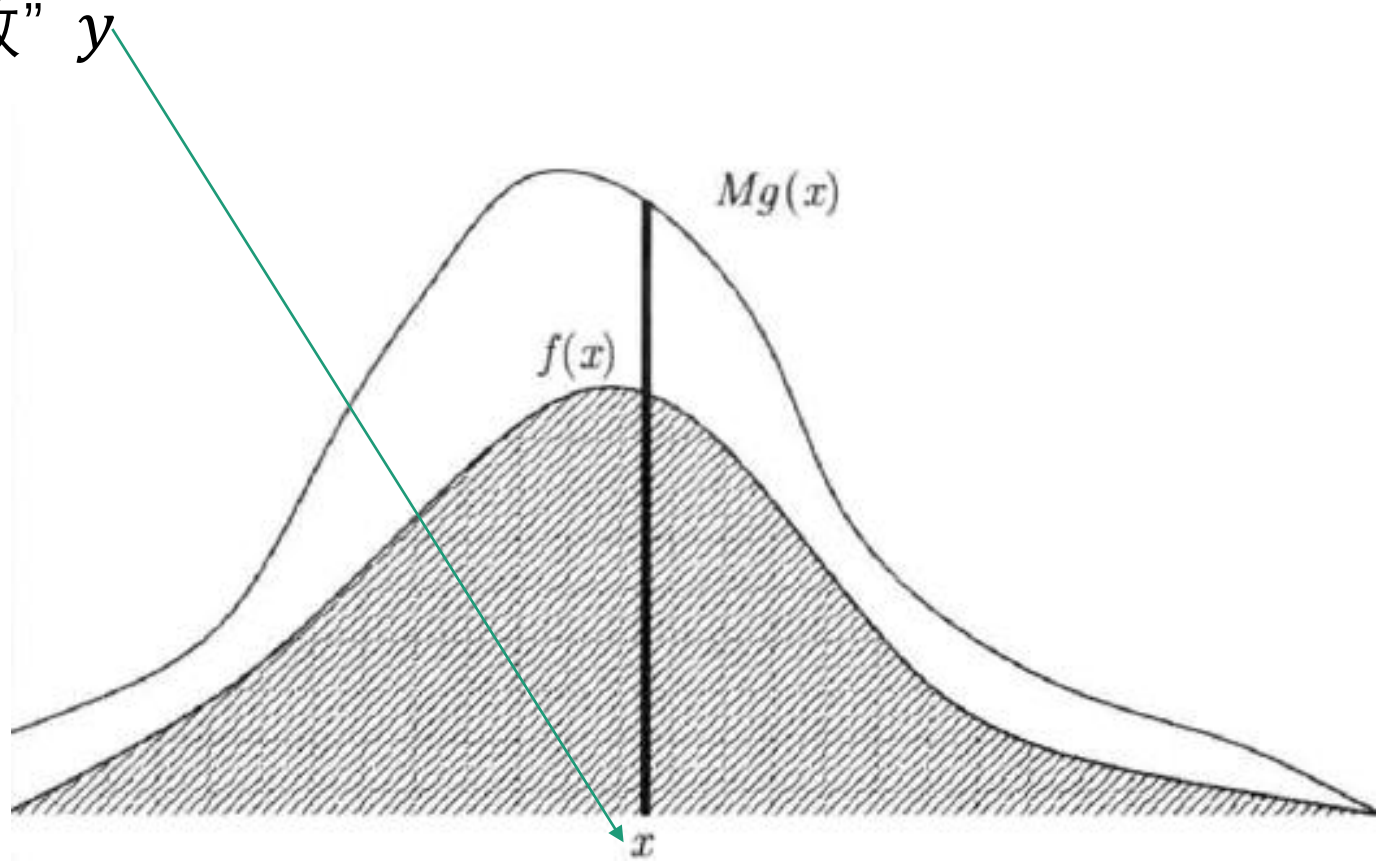
$$h(y) = \frac{f(y)}{Mg(y)},$$

若 $\mathbf{u < h(y)}$，

接受y；

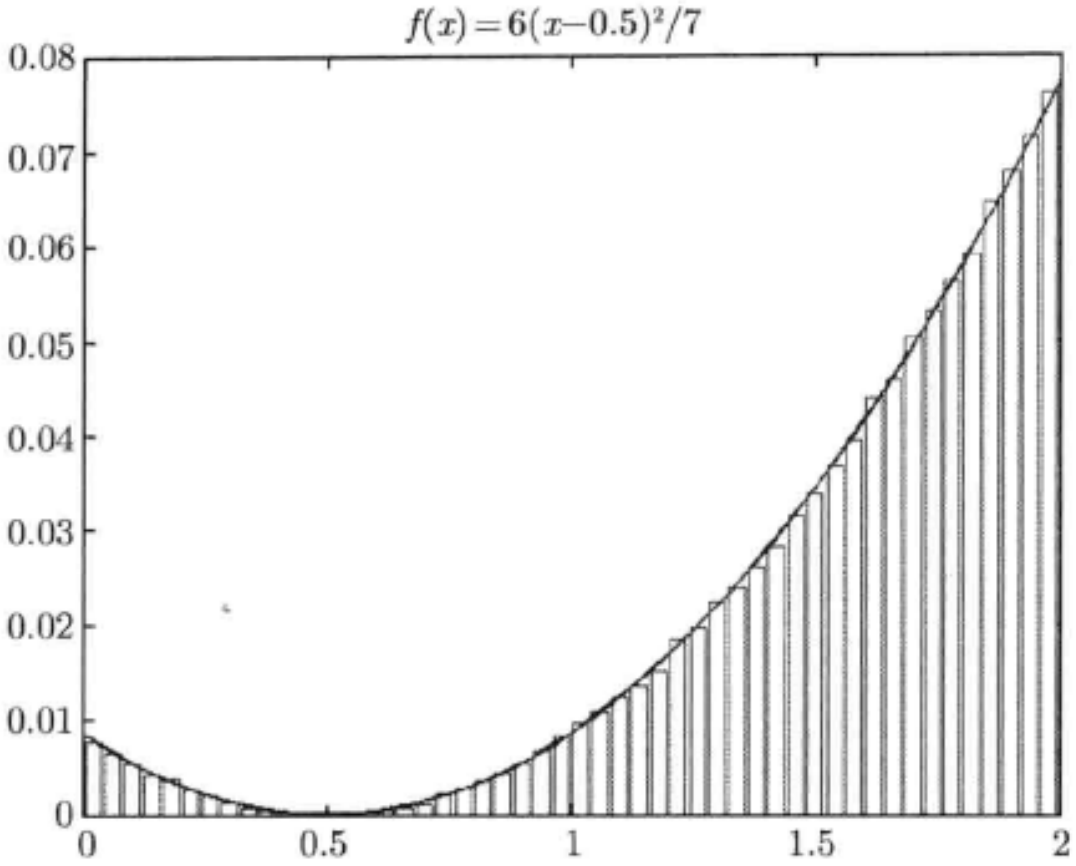
否则，

丢弃y，并转第2步；



例 4-2：$f(x) = \frac{6(x-0.5)^2}{7}, x \in (0,2)$

```
N = 500000;
Y = unifrnd(0, 2, 1, N);
U = unifrnd(0, 1, 1, N);
fy = 6 * (Y - 0.5).^2 / 7;
gy = 0.5; M = 3.858;
X = Y(U < fy ./ gy/ M);  % 根据面积比取舍
sample = length(X);
[Xnumber, Xcenters] = hist(X, 50);
bar(Xcenters, Xnumber / sample);
title("f(x) = 6(x - 0.5)^2/7");
t = 0:0.04:2;  z = 6 * (t - 0.5).^2 / 7; scale = 2/50;
hold on;  plot(t, scale * z, 'r');  hold off;
```



参考：demo_accept-reject.pdf

二项分布的拒绝-接受实现

```
function k = binomial(n, p)
    x = 0:1:n;
    T = binopdf(x, n, p);
    r = rand();
    S = 0;
    for i = 1:(n + 1)
        S = S + T(i);
        if r < S
            break
        end
    end
    k = i - 1;
end
```

```
% pkg load statistics;   % when use octave
N = 500;  % Total test times.
n = 10;   % Space size, 1, 2, ...,n
p = 0.5;  % rate of happen.
happened = zeros(1, n + 1);   % times of happened. including 0.
for i = 1:N
    k = binomial(n, p);  % in one test, k times succeed.
    happened(k + 1) = happened(k + 1) + 1;   % counting. build data for H.
end
bar(0:n, happened/N);
```

- 事实上:
  - ➢先生成均匀分布随机数
  - ➢再转换成某种分布的随机数

A-R方法的理论解释/证明

$$P\left(U \leqslant \frac{f(Y)}{Mg(Y)}\right) = \int_{-\infty}^{\infty} P\left(U \leqslant \frac{f(Y)}{Mg(Y)} \middle| Y = y\right) g(y)\mathrm{d}y$$

$$= \int_{-\infty}^{\infty} \frac{f(y)}{Mg(y)} g(y)\mathrm{d}y = \frac{1}{M}\int_{-\infty}^{\infty} f(y)\mathrm{d}y = \frac{1}{M}$$

记事件 $A = \{Y \leqslant y\}$ 和 $B = \left\{U \leqslant \frac{f(Y)}{Mg(Y)}\right\}$。

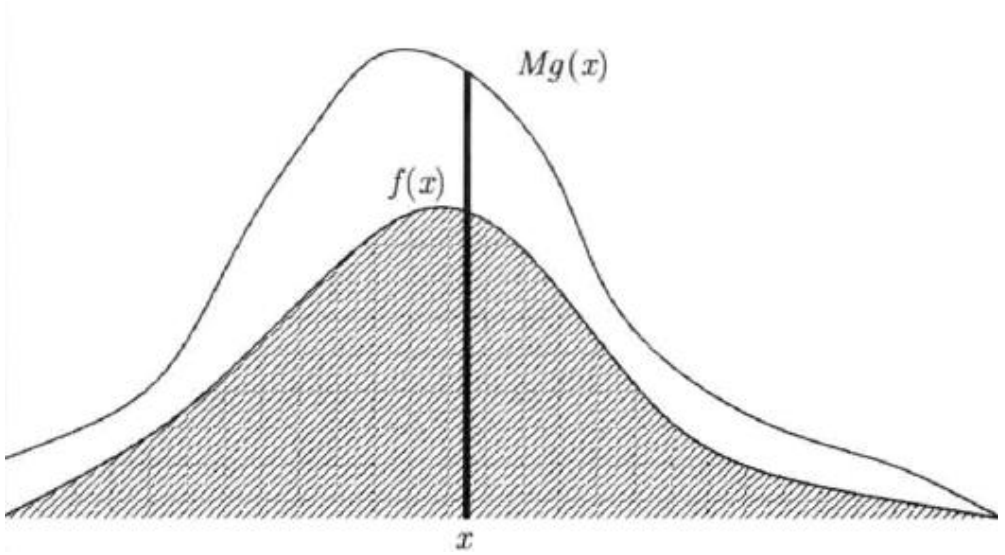
由条件概率公式 $P(A|B) = P(B|A)P(A)/P(B)$，可得

$$P\left(Y \leqslant y \middle| U \leqslant \frac{f(Y)}{Mg(Y)}\right) = P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$= \frac{P\left(U \leqslant \frac{f(Y)}{Mg(Y)} \middle| Y \leqslant y\right) P(Y \leqslant y)}{1/M}$$

$$= M\int_{-\infty}^{y} P\left(U \leqslant \frac{f(Y)}{Mg(Y)} \middle| Y = w < y\right) g(w)\mathrm{d}w$$

$$= M\int_{-\infty}^{y} \frac{f(w)}{Mg(w)} g(w)\mathrm{d}w = \int_{-\infty}^{y} f(w)\mathrm{d}w = F(y)$$



注意事项：

- A-R方法效率较低
- 合适的“建议概率密度函数”$g(x)$：
  1. $Mg(x) > f(x)$且“尽可能贴近”
  2. 让M尽可能地小
  3. $g(x)$容易被抽样

例4-3：“好一点”的建议分布(结合反变换法)

```
N = 500000;   M = 1.01;   % f,g尽可能贴合
U = unifrnd(0, 1, 1, N);
V = unifrnd(0, 1, 1, N);
T = 7/2 * U - 0.5^3;
Y = sign(T) .* abs(T) .^ (1/3) + 0.5;
gy = 6 * (Y - 0.5).^2 / 7;      % “建议”分布
fy = 6 * (Y - 0.5).^2 / 7;      % “目标”分布
X = Y(V < fy ./ gy / M);        % 只取所需
count = length(X);
bins = 50;
[Xnumber, Xcenters] = hist(X, bins);
bar(Xcenters, Xnumber / count);
title(“f(x) = 6(x - 0.5)^2/7”);   hold on;
t = 0:0.04:2;  z = 6 * (t - 0.5).^2 / 7;
scale = 2 / bins;
plot(t, scale * z, ‘r‘);  hold off;
```

```
N = 500000;
U = unifrnd(0, 1, 1, N);
T = 7/2 * U - 0.5^3;
Y = sign(T) .* abs(T) .^ (1/3) + 0.5;
[Xnumber, Xcenters] = hist(Y, 50);
bar(Xcenters, Xnumber / N);
title("f(x) = 6(x - 0.5)^2/7");  hold on;
t = 0:0.04:2;   z = 6 * (t - 0.5).^2 / 7;
scale = 2/50;
plot(t, scale * z, 'r’); hold off;
```

例4-4：半正态分布随机数的A-R方法

```
N = 500000;
U = unifrnd(0, 1, 1, N);
Y = -log(U);   % 逆变换法获负指数分布
hY = exp(-(Y - 1).^2 / 2);
V = unifrnd(0, 1, 1, N);

X = Y(V < hY & Y < 10);

count = length(X);
bins = 50;
[Xnumber, Xcenters] = hist(X, bins);
bar(Xcenters, Xnumber / count);
scale = max(X) / bins;
t = 0 : 0.04 : 10;
z = sqrt(2 / pi) * exp(-t.^2 / 2);
hold on;  plot(t, scale * z, 'r');   hold off;
```

$$f(x)=\sqrt{\frac{2}{\pi}}e^{-\frac{x^2}{2}}=\cdots=\sqrt{\frac{2e}{\pi}}e^{-\frac{(x-1)^2}{2}}e^{-x}$$

故令

$$M=\sqrt{\frac{2e}{\pi}},\qquad g(x)=e^{-x}$$

则有

$$h(x)=\frac{f(x)}{Mg(x)}=e^{-\frac{(x-1)^2}{2}}$$

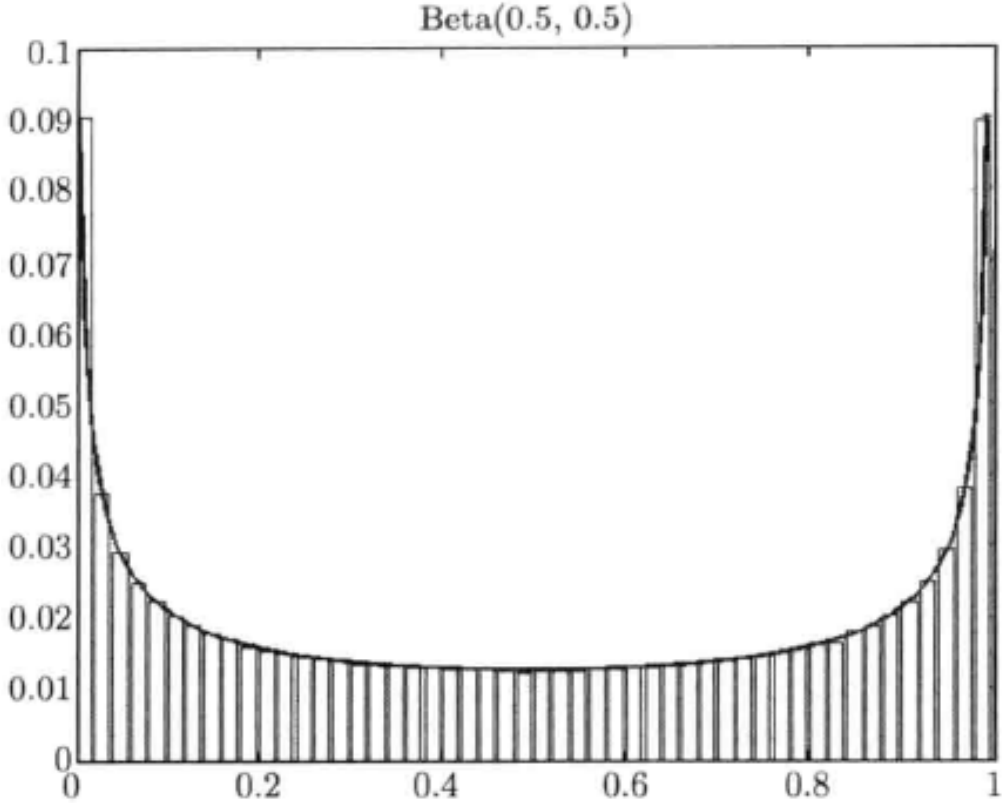
参考：demo_conti-var.pdf

上述方法是产生正态分布随机数的方法之一，但性能不高，偶尔可以用一下。

例4-5：无界概率密度函数 - Beta分布抽样

- 概率密度函数：

$$f(x) = cx^{\alpha-1}(1-x)^{\beta-1}, x \in (0,1)$$



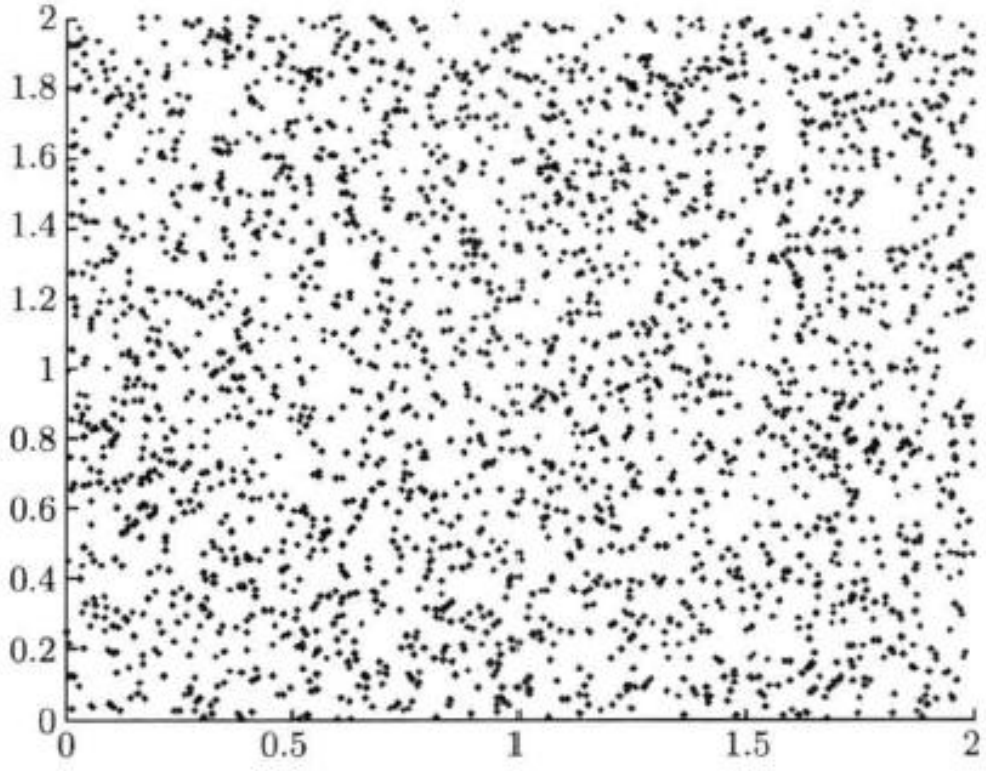

```
N = 500000;      alpha = 0.5;   % beta=0.5
U = unifrnd(0, 1, 1, N);
Y = 0.5 * U.^(1 / alpha);
h = (2 * (1 - Y)).^(alpha - 1);
R = unifrnd(0, 1, 1, N);
X = Y(R < h);

sample = length(X);              bins = 50;
S = unifrnd(0, 1, 1, sample);  S = (S > 0.5);
X = (1 - S).*X + S.*(1 - X)

[Xnumber, Xcenters] = hist(X, bins);
bar(Xcenters, Xnumber / sample);
title('Beta(0.5, 0.5)'); hold on;
dt = 0.005; t = dt : dt : 1 - dt;
z = (t.^(alpha - 1).*((1 - t).^(alpha - 1))) / pi;
scale = 1 / bins; plot(t, scale * z, 'r'); hold off;
```

3. 高维抽样

1. 多维联合分布抽样

- 若各分量独立，则独立完成：

```
>> X = unifrnd(0, 2, 2500, 2);
>> scatter(X(:, 1), X(:, 2), 'k.');
```



- 若各分量**不**独立，则各分量之间的相关性由**协方差矩阵** 或 **相关系数矩阵** 表示，记为ρ.

(1) 生成各维度上的独立的正态分布随机数，将它们合成向量 $\boldsymbol{Y}$;

(2) 将相关系数矩阵 ρ 做乔莱斯基分解，得到矩阵 $L(\rho = L\,L^{\mathrm{T}})$;

(3) 计算 $\boldsymbol{X} = L\boldsymbol{Y}$，即可得到服从上述要求的多元正态分布随机向量

例：(分量不独立的)三维正态分布

```
N = 10000;
Y = [normrnd(2, 3, 1, N); normrnd(-1, 2, 1, N); normrnd(0, 1, 1, N)];

rho = [   1,  0.3, 0.4; …
          0.3,    1, 0.2; …
          0.4,  0.2,   1];
L = chol(rho, 'lower');
X = L * Y;

scatter3(X(1, :), X(2, :), X(3, :), 'marker', '.', 'sizedata', 10);
xlabel('X轴'); ylabel('Y轴'); zlabel('Z轴'); axis equal;
```

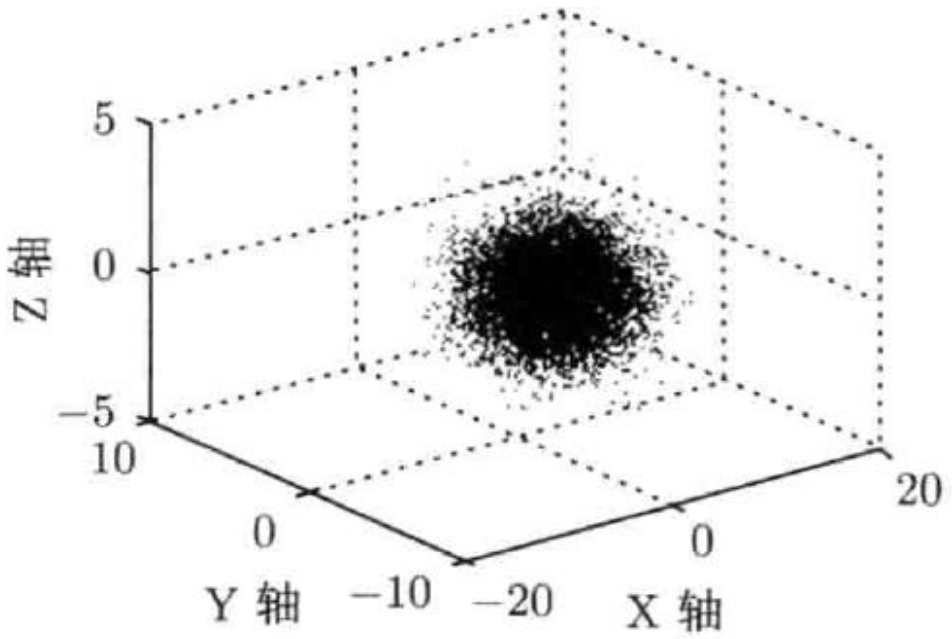
(1) 生成各维度上的独立的正态分布随机数，将它们合成向量 $\boldsymbol{Y}$；

(2) 将相关系数矩阵 ρ 做乔莱斯基分解，得到矩阵 $L(\rho = L\,L^{\mathrm{T}})$；

(3) 计算 $\boldsymbol{X} = L\boldsymbol{Y}$，即可得到服从上述要求的多元正态分布随机向量

维　度	均　值	标准差
(1)	2	3
(2)	−1	2
(3)	0	1

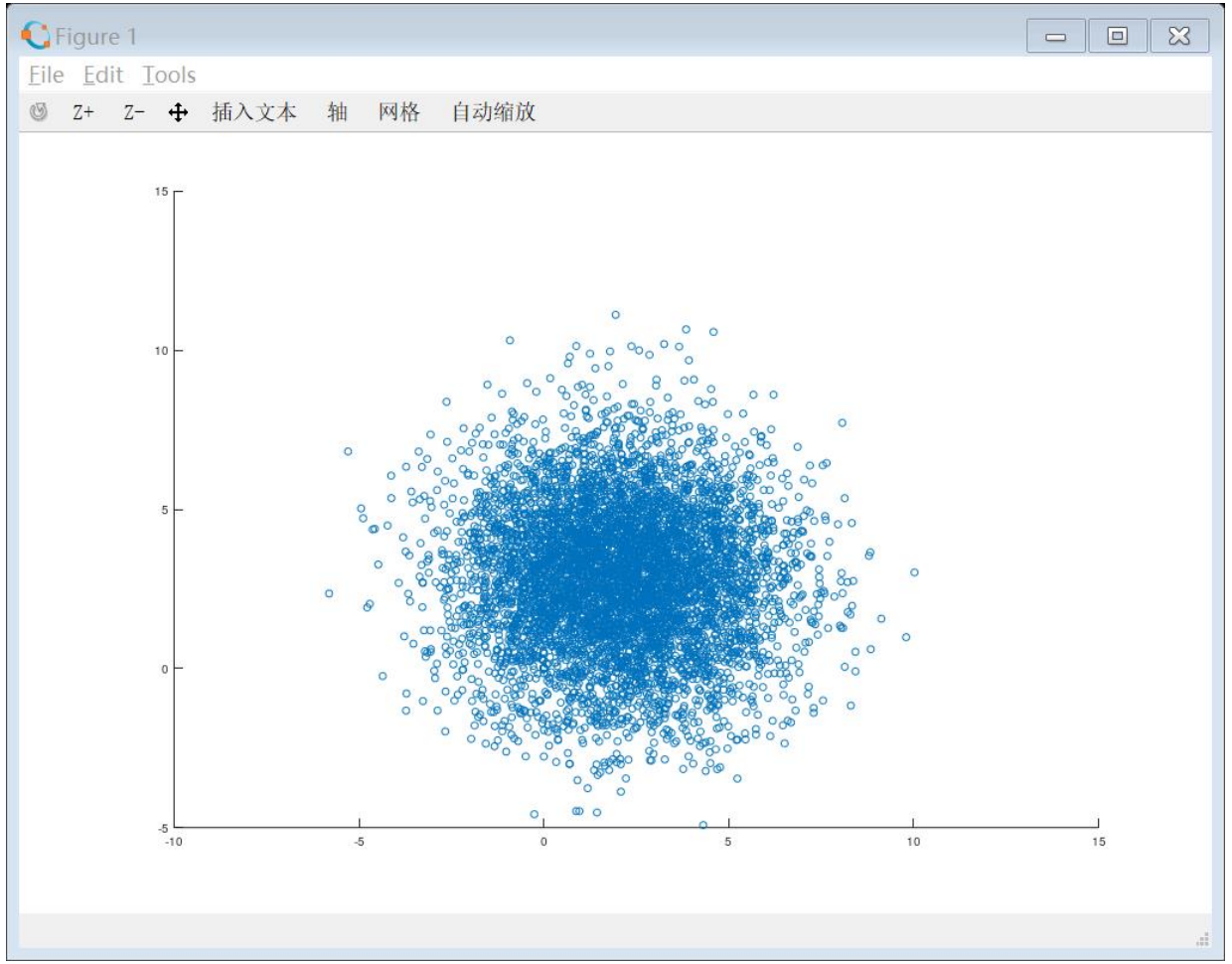
$$\rho = \begin{pmatrix} 1 & 0.3 & 0.4 \\ 0.3 & 1 & 0.2 \\ 0.4 & 0.2 & 1 \end{pmatrix}$$



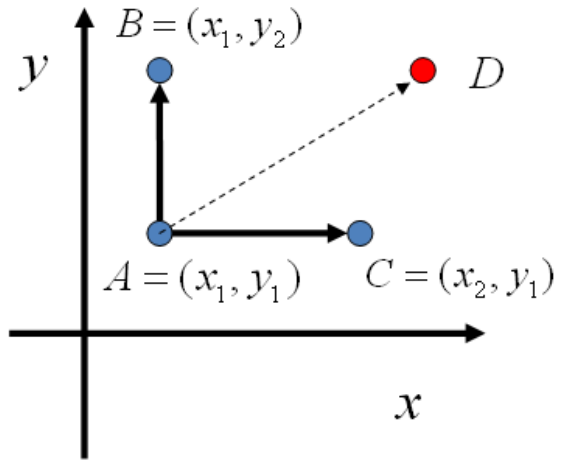
2.二维正态分布的M-H抽样示例

inGibbs.m * demoMH2D.m

```
pkg load statistics;
% 二元正态分布的参数:
mu = [2;3]; Sigma = [ 4, 1; 1, 4];

% 建议分布函数，可构造
fun_f = @(x, mu, sigma) 1/(2*pi*sqrt(det(sigma)))* ...
                exp(-0.5*sum(sigma\((x - mu).*(x - mu))));

# 用MCMC的 Metropolis-Hastings 抽样法生成后验分布
Nsample = 10000;
x = zeros(2, Nsample);
n_accept = 0;
x_curr = [0;0];  % initialize
f_curr = fun_f(x_curr, mu, Sigma);
sensitivity = 2.0;
for k = 1:Nsample
  x_recom = x_curr + sensitivity*normrnd(0,1,2,1);
  f_recom = fun_f(x_recom, mu, Sigma);
  if(rand < f_recom/f_curr)  % accept
    n_accept = n_accept + 1;
    x(:,n_accept) = x_recom;
    x_curr = x_recom;
    f_curr = f_recom;
  end
end
printf('Accept rate: %f. \n', n_accept/Nsample);
scatter(x(1,:), x(2,:))
```



3.Gibbs Sampling – 满足细致平稳分布的抽样



于是我们可以如下构造平面上任意两点之间的转移概率矩阵Q

$$
\begin{aligned}
Q(A \to B) &= p(y_B|x_1) \quad \text{如果} \quad x_A = x_B = x_1 \\
Q(A \to C) &= p(x_C|y_1) \quad \text{如果} \quad y_A = y_C = y_1 \\
Q(A \to D) &= 0 \qquad\qquad\qquad\qquad \text{其它}
\end{aligned}
$$

有了如上的转移矩阵 Q, 我们很容易验证对平面上任意两点 X, Y, 满足细致平稳条件

$$p(X)Q(X \to Y) = p(Y)Q(Y \to X)$$

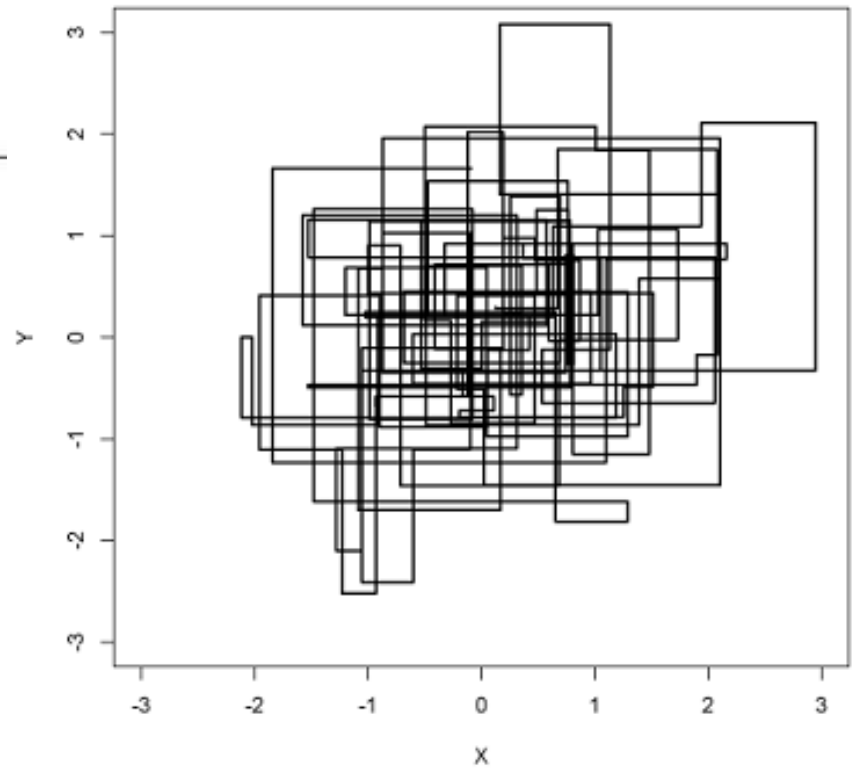
于是这个二维空间上的马氏链将收敛到平稳分布 $p(x, y)$。而这个算法就称为 Gibbs Sampling 算法,是 Stuart Geman 和Donald Geman 这两兄弟于1984年提出来的，之所以叫做Gibbs Sampling 是因为他们研究了Gibbs random field, 这个算法在现代贝叶斯分析中占据重要位置。

Algorithm 7 二维Gibbs Sampling 算法

1: 随机初始化$X_0 = x_0 Y_0 = y_0$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $y_{t+1} \sim p(y|x_t)$
2. $x_{t+1} \sim p(x|y_{t+1})$



Gibbs Sampling：高维情形……

Algorithm 8 n维Gibbs Sampling 算法

1: 随机初始化$\{x_i : i = 1, \cdots, n\}$

2: 对$t = 0, 1, 2, \cdots$ 循环采样

1. $x_1^{(t+1)} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \cdots, x_n^{(t)})$
2. $x_2^{(t+1)} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \cdots, x_n^{(t)})$
3. $\cdots$
4. $x_j^{(t+1)} \sim p(x_j | x_1^{(t+1)}, \cdots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \cdots, x_n^{(t)})$
5. $\cdots$
6. $x_n^{(t+1)} \sim p(x_n | x_1^{(t+1)}, x_2^t, \cdots, x_{n-1}^{(t+1)})$

Gibbs Sampling 算法中的马氏链转移

以上采样过程中，如图所示，马氏链的转移只是轮换的沿着坐标轴 x轴和y轴做转移，于是得到样本 $(x_0, y_0), (x_0, y_1), (x_1, y_1), (x_1, y_2), (x_2, y_2), \cdots$ 马氏链收敛后，最终得到的样本就是 $p(x, y)$ 的样本，而收敛之前的阶段称为 burn-in period。额外说明一下，我们看到教科书上的 Gibbs Sampling 算法大都是坐标轴轮换采样的，但是这其实是不强制要求的。最一般的情形可以是，在t时刻，可以在x轴和y轴之间随机的选一个坐标轴，然后按条件概率做转移，马氏链也是一样收敛的。轮换两个坐标轴只是一种方便的形式。

以上的过程我们很容易推广到高维的情形，对于(***) 式，如果x_1 变为多维情形$\mathbf{x_1}$，可以看出推导过程不变，所以细致平稳条件同样是成立的

$$p(\mathbf{x_1}, y_1)p(y_2|\mathbf{x_1}) = p(\mathbf{x_1}, y_2)p(y_1|\mathbf{x_1}) \tag{5}$$

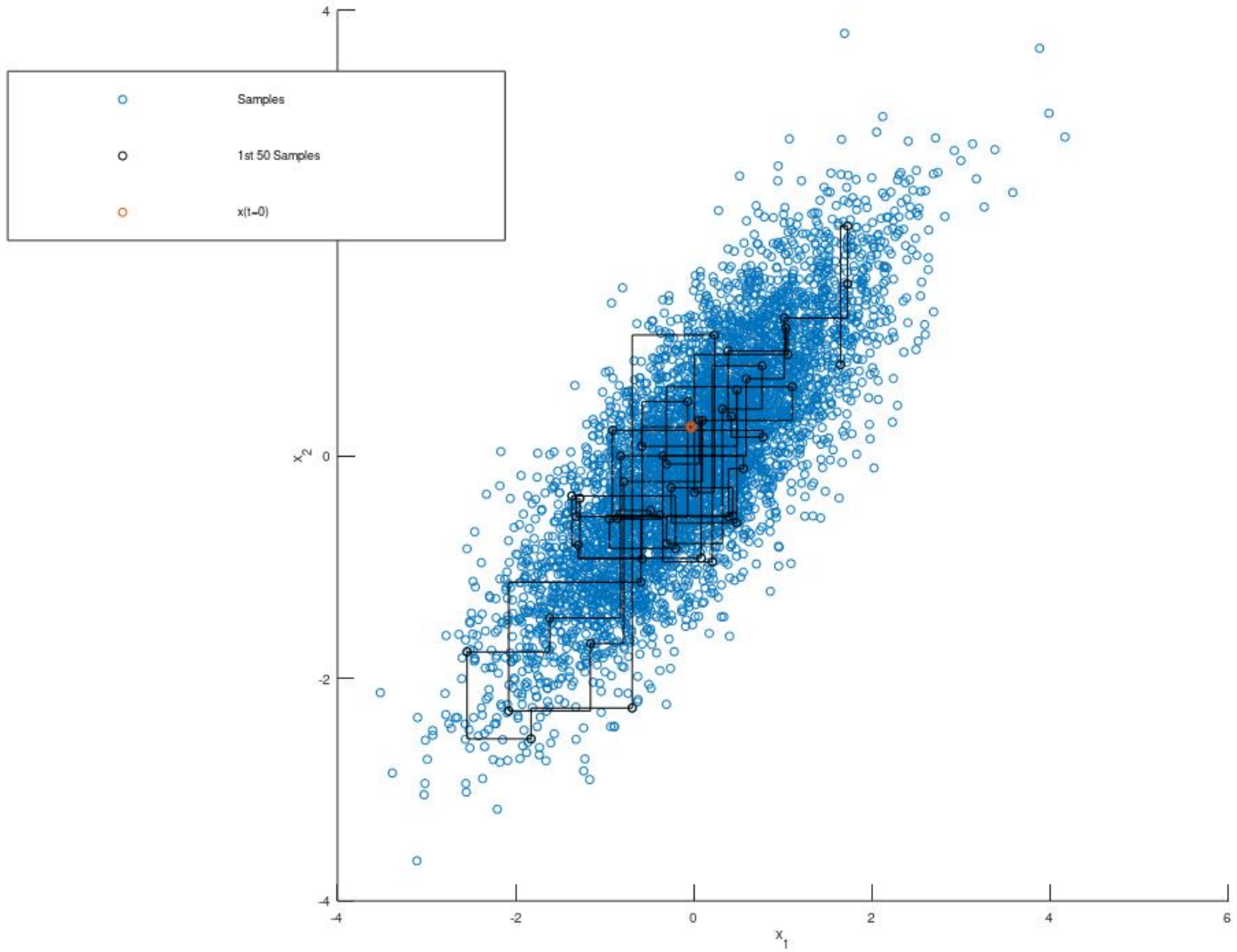
此时转移矩阵 Q 由条件分布 $p(y|\mathbf{x_1})$ 定义。上式只是说明了一根坐标轴的情形，和二维情形类似，很容易验证对所有坐标轴都有类似的结论。所以n维空间中对于概率分布 $p(x_1, x_2, \cdots, x_n)$ 可以如下定义转移矩阵

1. 如果当前状态为$(x_1, x_2, \cdots, x_n)$，马氏链转移的过程中，只能沿着坐标轴做转移。沿着 x_i 这根坐标轴做转移的时候，转移概率由条件概率 $p(x_i|x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ 定义；
2. 其它无法沿着单根坐标轴进行的跳转，转移概率都设置为 0。

于是我们可以把Gibbs Smapling 算法从采样二维的 $p(x, y)$ 推广到采样n 维的 $p(x_1, x_2, \cdots, x_n)$

示例: mainGibbs.m

```
nSamples = 5000;
mu = [0 0]; % TARGET MEAN目标均值
rho(1) = 0.8; % rho_21目标方差
rho(2) = 0.8; % rho_12目标方差

% INITIALIZE THE GIBBS SAMPLER
propSigma = 1; % PROPOSAL VARIANCE
minn = [-3 -3];
maxx = [ 3  3];

% INITIALIZE SAMPLES
x = zeros(nSamples,2);
x(1,1) = unifrnd(minn(1), maxx(1));  %unifrnd生成
x(1,2) = unifrnd(minn(2), maxx(2));
dims = 1:2; % INDEX INTO EACH DIMENSION

t = 1;  % RUN GIBBS SAMPLER
while t < nSamples%总共采样出5000个采样点
    t = t + 1;      T = [t-1,t];
    for iD = 1:2 % LOOP OVER DIMENSIONS
        % *NOT* THE CURRENT DIMENSION
        nIx = dims~=iD; %
        % CONDITIONAL MEAN, μ(1)+ρ(1)*(x(n,2)-μ(2
        muCond = mu(iD) + rho(iD)*(x(T(iD),nIx)-m
        varCond = sqrt(1-rho(iD)^2);% 计算 CONDIT
        % DRAW FROM CONDITIONAL
        x(t,iD) = normrnd(muCond,varCond);
    end
end
```



Homework 4

- 课本$P_{107-108}$：1、3、5、7;
- 上机练习（将所作练习的概要写到作业本上）：
  1. 请换用Gibbs采样方法，重做圆周率的随机投点模拟（二维）
  2. 从下列抽样算法中**自选一类**，并查阅相关文献，解释所选算法并实现该算法（给出抽样的统计结果）：
     1. Fisher-Yates shuffle(洗牌)算法
     2. Knuth-Durstenfeld Shuffle(洗牌)算法
     3. 蓄水池(**Reservoir**)抽样算法