STAT 428: Project 1

Yifan Sun

4/17/2020

Contents

Bivariate Normal Distribution

Initial set-up
Let's generate 1000 random samples from bivariate normal distribution $x = (x_1, x_2)' \sim N(\mu, \Sigma)$.

```
set.seed(1)
n = 1000
# Firstly, Set target parameters (mean, standard deviation and correlation) for purposal distribution
rho = -0.5
mu1 = 0
mu2 = 0
s1 = 1
s2 = 1
# Mean vector for purposal bivariate normal distribution
mu = c(mu1,mu2)
# Covariance matrix for purposal bivariate normal distribution
sigma = matrix(c(s1^2, s1*s2*rho, s1*s2*rho, s2^2),2)
```

Cholesky Decomposition

Basic idea
Cholesky decomposition is a very useful and efficient method of decompositing positive-definite matrix. The positive-definite matrix is defined as a symmetrix matrix where for all

$$x'Ax > 0$$

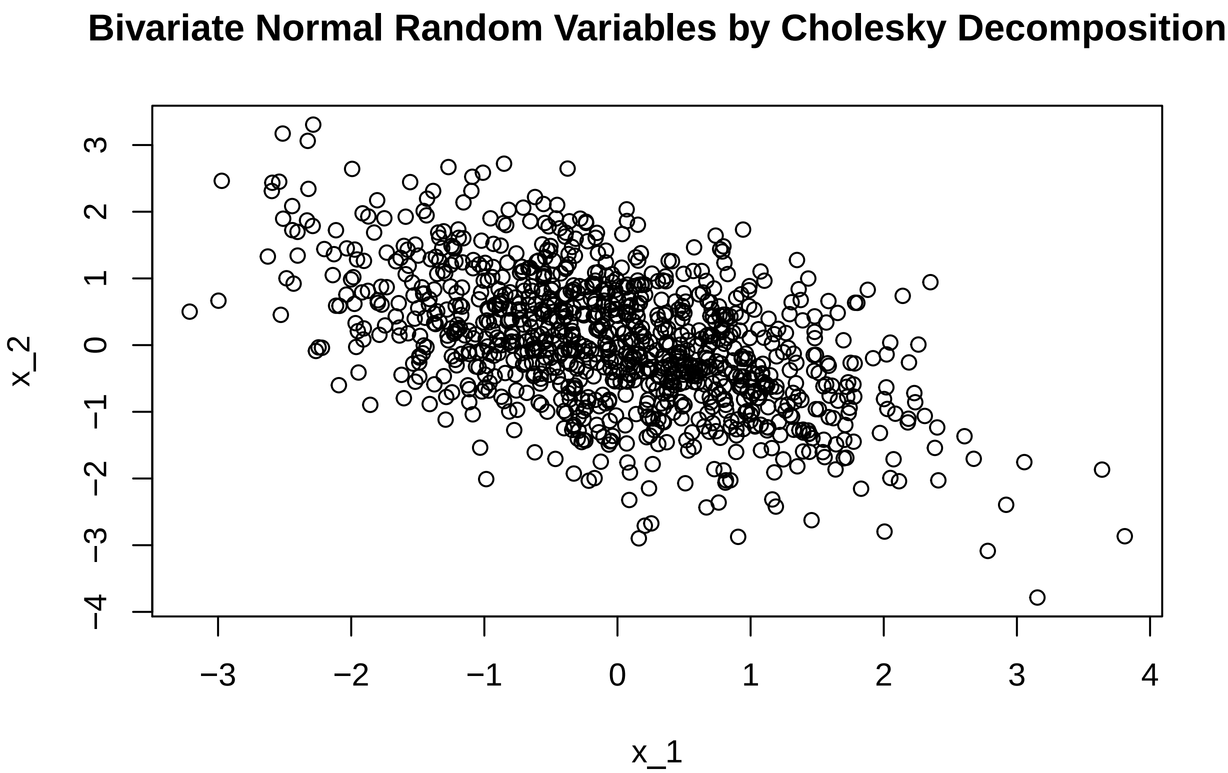
Because covariance matrix in bivariate normal distributiin is a symmetric matrix and

$$x'\Sigma x > 0$$

for all vector x, where Σ denotes the covariance matrix, so it is easy to conclude that covariance matrix in a given bivariate normal distribution is positive-definite matrix.
Algorithm
(1) Decomposition of covariance matrix Σ: we can use r function `chol` to make the decomposition and set the lower triangular matrix as L. So $L = Chol(\Sigma)$ and $LL' = \Sigma$.
(2) Let vector z follows standard normal distribution $N(0, 1)$, then $z = (z_1, z_2)'$.

(3) Use z and rescale it to generate target bivariate normal random variables. $\mu + Lz \sim N(\mu, \Sigma)$
Source
https://www2.stat.duke.edu/courses/Spring12/sta104.1/Lectures/Lec22.pdf
http://math.ubbcluj.ro/~tradu/Randvargen.pdf
[Mathematical Statistics: Old School]http://www.istics.net/pdfs/mathstat.pdf

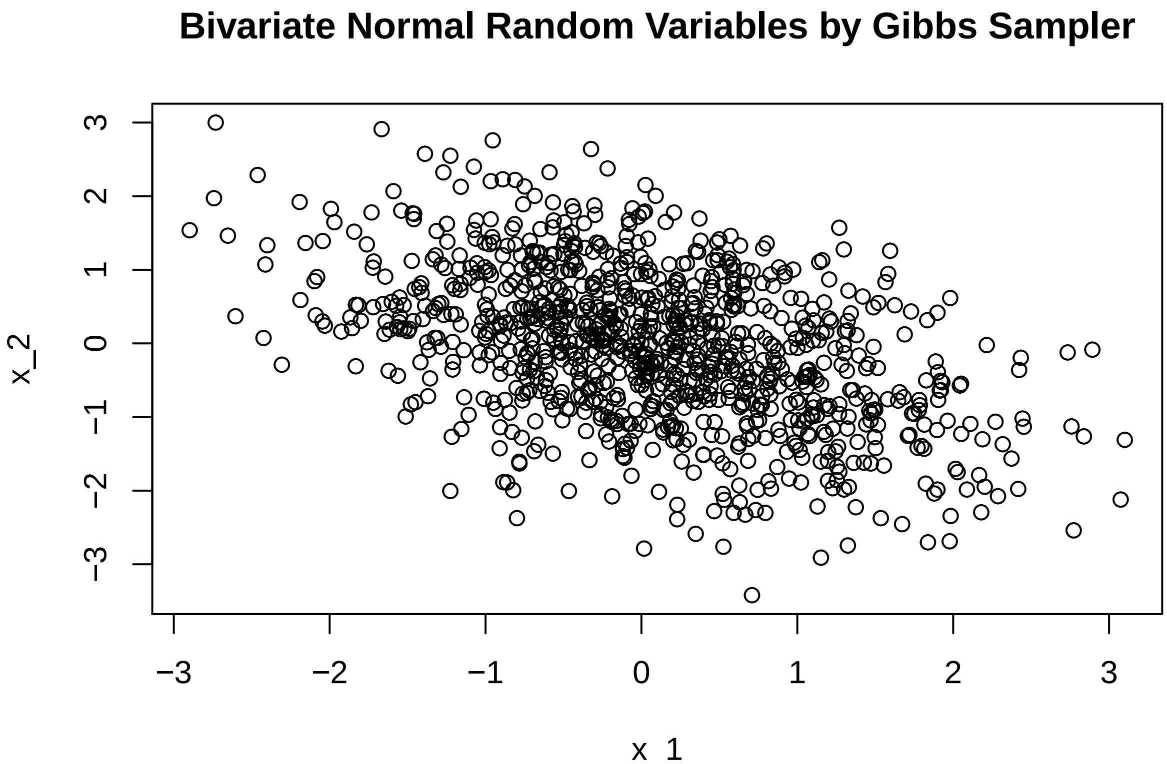
```
# Transpose the lower triangular matrix
U = t(chol(sigma))
# Standard normal vector z
z = matrix(rnorm(2*n), nrow = 2, ncol = n)
x = matrix(rep(mu, n), ncol = 2, byrow = TRUE) + t(U %*% z)
plot(x, main = "Bivariate Normal Random Variables by Cholesky Decomposition", xlab = "x_1", ylab = "x_2
```



Gibbs Sampler

Basic idea
The Gibbs sampler is applied when the target distribution is multivariate distribution and draws from conditional distribution. The chain is generated by the marginal distribution of target distribution. Let's $x = (x_1, x_2, \dots)$ and $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1} \dots)$. The Gibbs sampler generate chain by sampling from every conditional density function $f(x_i|x_{-i})$. For bivariate normal distribution, we have conditional density function as follow: $f(x_1|x_2) \sim N(\mu_1 + \frac{\rho * \sigma_1}{\sigma_2}(x_2 - \mu_2), (1-\rho^2)\sigma_1^2)$
Algorithm
(1) Initialize x_1 and x_2 first.
(2) Generate x_1 and x_2 from conditional functions $f(x_1|x_2)$ and $f(x_2|x_1)$.
(3) Update the x_1 and x_2 to the dataset.
(4) Repeat step 2 and step 3 n times.
Source
https://www.aptech.com/resources/tutorials/bayesian-fundamentals/gibbs-sampling-from-a-bivariate-normal-distribution/
[Mathematical Statistics: Old School]http://www.istics.net/pdfs/mathstat.pdf

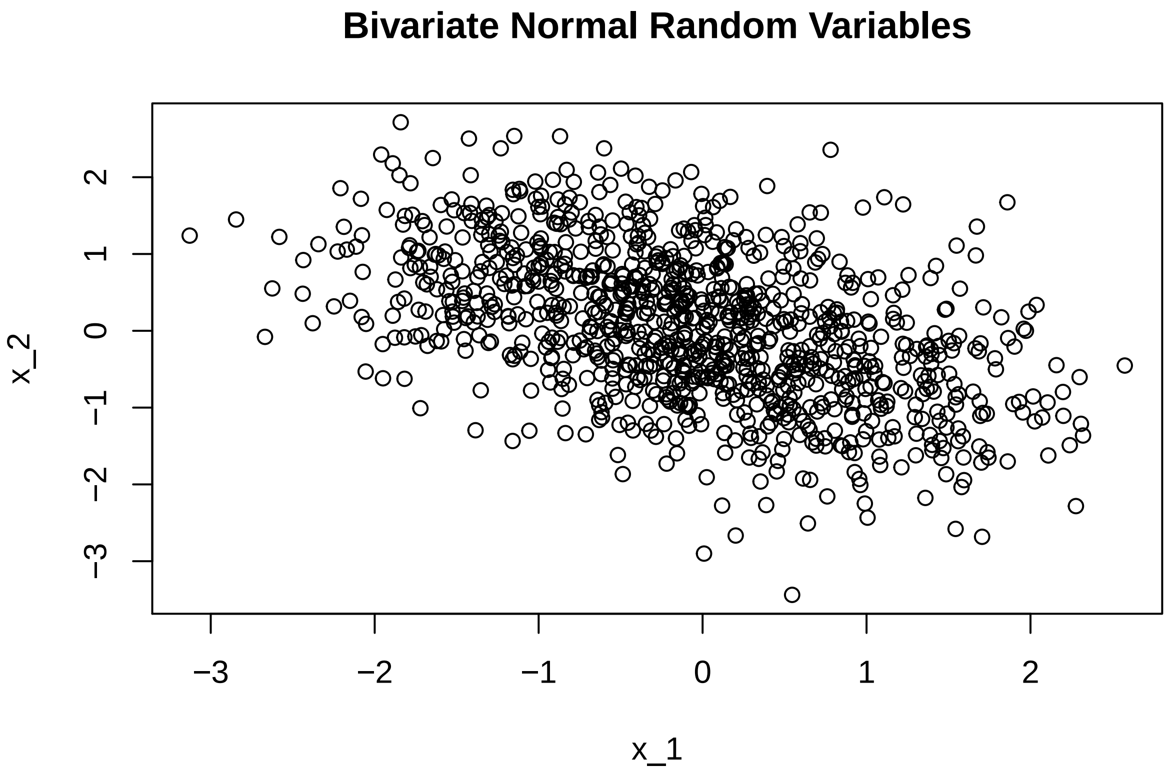
```r
x = matrix(ncol = 2, nrow = n)
# Set x_1 and x_2 to zero
x_1 = 0
x_2 = 0
x[1, ] = c(x_1, x_2)
for (i in 2:n) {
  x_1 = rnorm(1, mean = mu[1] + (s1/s2) * rho * (x_2 - mu[2]), sd = sqrt((1 - rho^2)*s1^2))
  x_2 = rnorm(1, mean = mu[2] + (s2/s1) * rho * (x_1 - mu[1]), sd = sqrt((1 - rho^2)*s2^2))
  x[i, ] = c(x_1, x_2)
}
plot(x, main = "Bivariate Normal Random Variables by Gibbs Sampler", xlab = "x_1", ylab = "x_2", )
```



Check

There is a function `mvtnorm` in r generating bivariate normal distribution directly.

```r
library(mvtnorm)
x = rmvnorm(n = n, mean = mu, sigma = sigma)
plot(x, main = "Bivariate Normal Random Variables", xlab = "x_1", ylab = "x_2", )
```



Feedback