Lab6

Bingying Liu

February 24, 2020

Exercise 1: Given the multivariate normal distribution above, what are the posterior complete conditonals for X,Y and Z?

$

$$p(X|Y,Z) \sim N(0 + \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix} \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}^{-1} (\begin{pmatrix} Y \\ Z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}), 0.1899)$$

$

$

$$p(Y|X,Z) \sim N(0 + \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix} \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}^{-1} (\begin{pmatrix} X \\ Z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}), 0.1899)$$

$

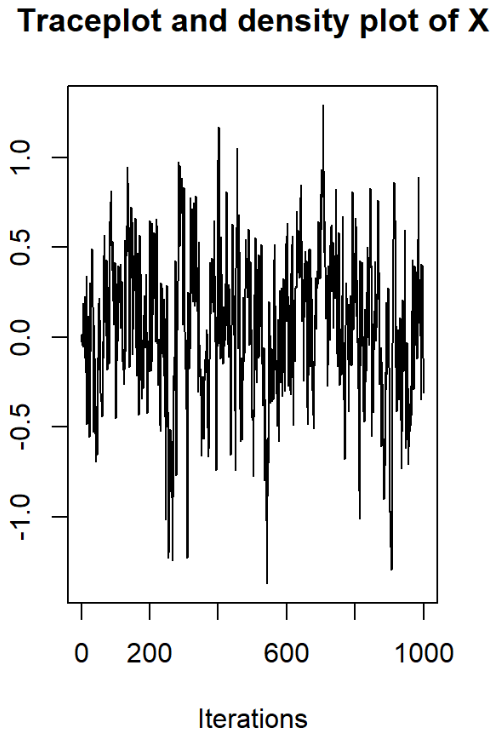
$

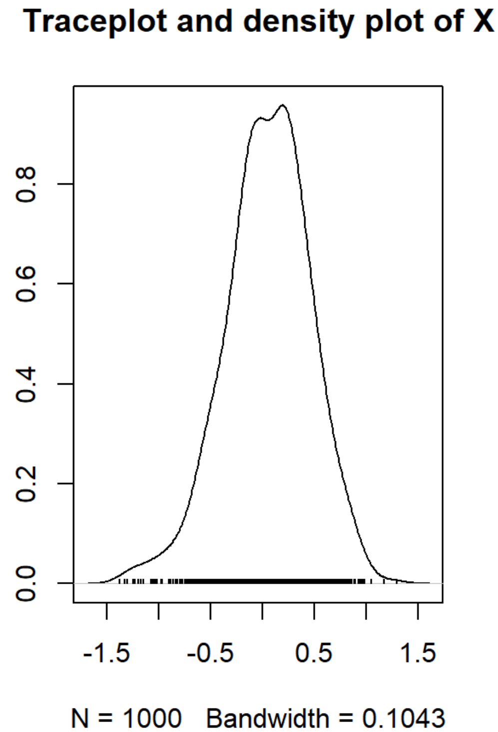
$$p(Z|X,Y) \sim N(0 + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}^{-1} (\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}), 0.9895)$$

$

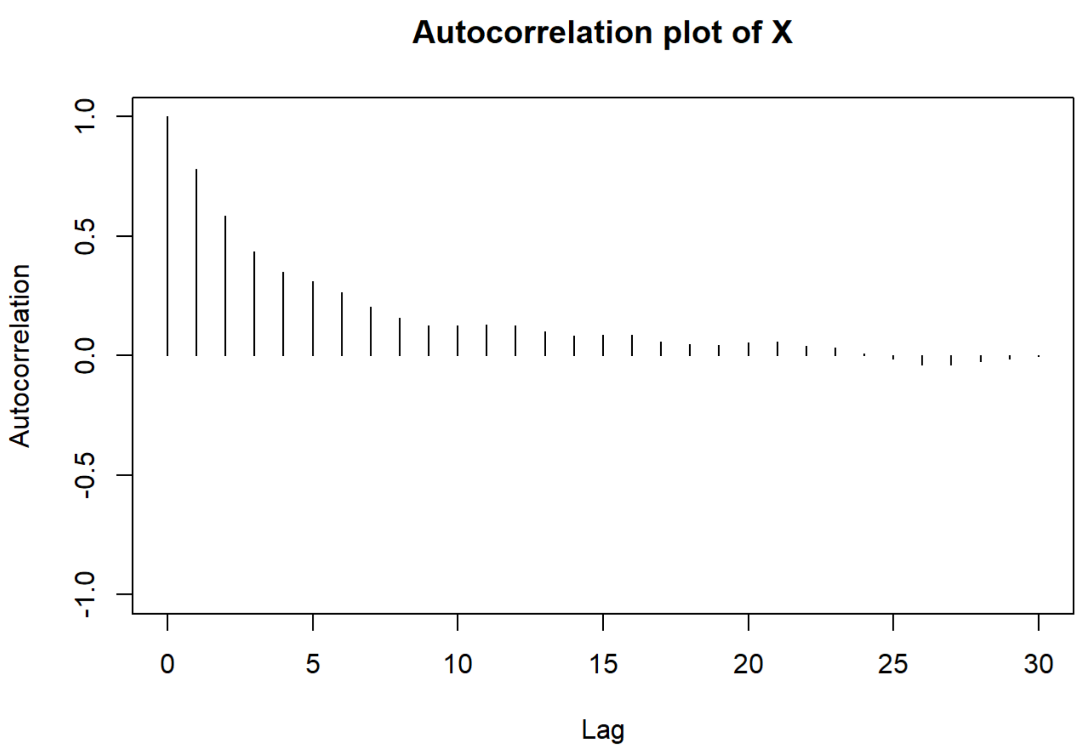
Exercise 2: Write a Gibbs sampler that alternates updating each of the variables and comment on the plots.

```r
# p(x|y,z)
mux_1 = 0
mux_2 = c(0,0)
sigmax_11 = 1
sigmax_22 = matrix(c(1,0.1,0.1,1),nrow=2,ncol=2)
sigmax_12 = sigmax_21 = c(0.9,0.1)

# p(y|x,z)
muy_1 = 0
muy_2 = c(0,0)
sigmay_11 = 1
sigmay_22 = matrix(c(1,0.1,0.1,1),nrow=2,ncol=2)
sigmay_12 = sigmay_21 = c(0.9,0.1)


# p(z|x,y)
muz_1 = 0
muz_2 = c(0,0)
sigmaz_11 = 1
sigmaz_22 = matrix(c(1,0.9,0.9,1),nrow=2,ncol=2)
sigmaz_12 = sigmaz_21 = c(0.1,0.1)

## Gibbs sampler
n_iter <- 1000; burn_in <- 0.3*n_iter
set.seed(1234)

# Innitialize the values
X_n = Y_n = Z_n = 0
X = Y = Z = NULL

for (s in 1:(n_iter+burn_in)){
  #update X_n
  mux = mux_1 + sigmax_12%*%solve(sigmax_22)%*%(c(Y_n,Z_n) - mux_2)
  varx = sigmax_11 - sigmax_12%*%solve(sigmax_22)%*%sigmax_21
  X_n <- rnorm(1, mean = mux, sd= varx)

  #update Y_n
  muy = muy_1 + sigmay_12%*%solve(sigmay_22)%*%(c(X_n,Z_n) - muy_2)
  vary = sigmay_11 - sigmay_12%*%solve(sigmay_22)%*%sigmay_21
  Y_n <- rnorm(1, mean = muy, sd= vary)

  #update Z_n
  muz = muz_1 + sigmaz_12%*%solve(sigmaz_22)%*%(c(X_n,Y_n) - muz_2)
  varz = sigmaz_11 - sigmaz_12%*%solve(sigmaz_22)%*%sigmaz_21
  Z_n <- rnorm(1, mean = muz, sd= varz)

  #save results only past burn-in
  if(s > burn_in){
    X <- rbind(X,X_n)
    Y <- rbind(Y,Y_n)
    Z <- rbind(Z,Z_n)
  }
}

# trace plot of X
X.mcmc <- mcmc(X,start=1)
plot(X.mcmc, main='Traceplot and density plot of X')
```






```
# autocorrelation plot of X
autocorr.plot(X.mcmc, main='Autocorrelation plot of X')
```



In the traceplot, there exists some “snaking” behavior with cyclic local trebds in the mean, which shows high posterior correlation in the parameters, especially X and Y. The autocorrelation plot of X shows that autocorrelation doesn’t completely go to zero before lag 20, we have slow mixing problem.

Exercise 3: Give the conditional distributions for $(X, Y)|Z$ and $Z|(X, Y)$.

$

$$p(X,Y|Z) \sim N(\begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} (\, 1 \,)^{-1}((\, Z \,) - (\, 0 \,)), \begin{pmatrix} 0.99 & 0.89 \\ 0.89 & 0.99 \end{pmatrix})$$

$

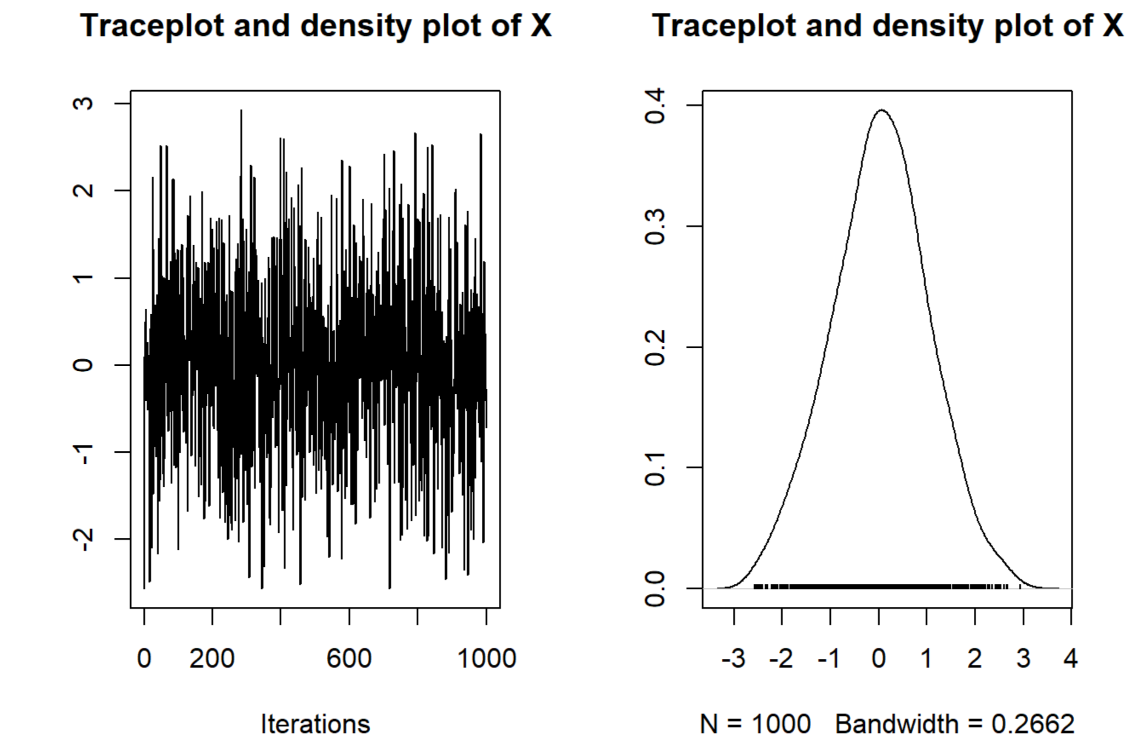
$

$$p(Z|X,Y) \sim N(0 + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}^{-1} (\begin{pmatrix} X \\ Y \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}), 0.9895)$$

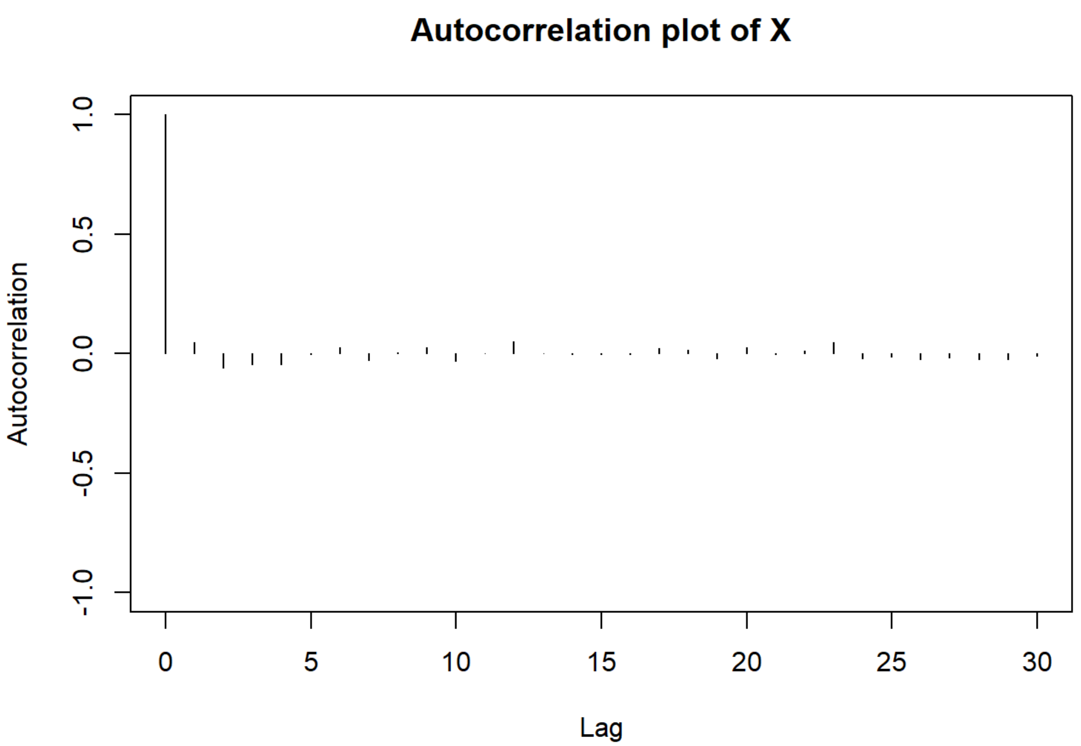
$

Exercise 4: Write a Gibbs sampler using the conditional distributions in Exercise 3 above. Comment on the plots.

```
# p((x,y)|z)
muxy_1 = c(0,0)
muxy_2 = 0
sigmaxy_11 = matrix(c(1,0.9,0.9,1),nrow=2,ncol=2)
sigmaxy_22 = 1
sigmaxy_12 = sigmaxy_21 = c(0.1,0.1)

#p(z|(x,y) is given above

## Gibbs Sampler
n_iter <- 1000; burn_in <- 0.3*n_iter
set.seed(1234)

#Innitialize the values
Z_n = 0
XY = Z = NULL

for (s in 1:(n_iter+burn_in)){
  #update X_n, Y_n
  muxy = muxy_1 + sigmaxy_12%*%solve(sigmaxy_22)%*%(c(Z_n) - muxy_2)
  varxy = sigmaxy_11 - sigmaxy_12%*%solve(sigmaxy_22)%*%sigmaxy_21
  XY_n <- rmvnorm(1, mean = muxy, sigma= varxy)

  #update Z_n
  muz = muz_1 + sigmaz_12%*%solve(sigmaz_22)%*%(c(XY_n) - muz_2)
  varz = sigmaz_11 - sigmaz_12%*%solve(sigmaz_22)%*%sigmaz_21
  Z_n <- rnorm(1, mean = muz, sd= varz)

  #save results only past burn-in
  if(s > burn_in){
    XY <- rbind(XY,XY_n)
    Z <- rbind(Z,Z_n)
  }
}

colnames(XY) <- c("X","Y")

# Diagnostics
XY.mcmc <- mcmc(XY,start=1);

plot(XY.mcmc[,'X'], main = 'Traceplot and density plot of X')
```




```
autocorr.plot(XY.mcmc[,'X'], main='Autocorrelation plot of X')
```



The traceplot of X doesn’t show the cyclic local trends, which means it has a good mixing. In addition, in the autocorrelation plot X, autocorrelation completely decreases to zero after lag 1, which also shows a good mixing.

Exercise 5: Comment on the difference between the performance of the two Gibbs samplers. Why is the second more efficient?

Since X and Y are highly correlated, if we sample them individually, then it will result in poor mixing. However, if we group X and Y together using block updates and then sample Z conditioned on X and Y, there won't be high correlation problem since X, Y as a group has low correlation with Z.

The more dependence there is between X and Y, the more efficient it will be to update them jointly. Sampling X and Y which are highly correlated individually is less efficient.