HW#4 Gibbs Sampling of Joint porbabilities using created function and JAGS

Elijah Hall

April 25, 2018

## Homework 4

### 1

Suppose A and B are two binary random variables such that P(A=1,B=1) = .25, P(A=0,B=0)=.25, P(A=1,B=0)=.25, and P(A=0,B=1)=.25. Write a Gibbs sampler to draw 200 samples of (A,B). Do the results approximate the distribution of (A,B)?

#1)----  
#define joint probabilities   
Pa1b0 <- .25  
Pa1b1 <- .25  
Pa0b1 <- .25  
Pa0b0 <- .25  
  
  
#create gibbs sampler funtion  
gibbs<-function (n, Pa1b0, Pa1b1, Pa0b1,Pa0b0)   
{  
 #Define conditional probabilities  
 # P(A=1|B=0)  
 con\_Pa1b0 <- Pa1b0/(Pa1b0+Pa0b0)  
 # P(A=1|B=1)  
 con\_Pa1b1 <- Pa1b1/(Pa1b1+Pa0b1)  
 # P(B=1|A=0)   
 con\_Pb1a0 <- Pa0b1/(Pa0b0+Pa0b1)  
 # P(B=1|A=1)  
 con\_Pb1a1 <- Pa1b1/(Pa1b0+Pa1b1)  
   
 #create matrix to log each step as it moves from A[i] to B[i] and from B[i] to A[i+1] and so on.  
 mat <- matrix(ncol = 2, nrow = n)  
 A = 0  
 B = 0  
 #define starting point  
 mat[1, ] = c(A, B)  
 #Make n steps staring at mat[2,1] s.t. the first column is A and the second is B  
 for (i in 2:n) {  
 if (mat[c(i-1),2] == 0){  
 A = rbinom(n = 1,size = 1,prob = con\_Pa1b0) } else {  
 A = rbinom(n = 1,size = 1,prob = con\_Pa1b1)  
 }  
 if (A == 0){  
 B = rbinom(n = 1,size = 1,prob = con\_Pb1a0)} else {  
 B = rbinom(n = 1,size = 1,prob = con\_Pb1a1)  
 }  
 mat[i, ] <- c(A, B)  
 }  
 mat  
}  
  
#Run sample for n=200  
bvn = gibbs(200,Pa1b0, Pa1b1, Pa0b1,Pa0b0)  
  
# create prior table  
x <- c(Pa1b0,Pa1b1)  
y <- c(Pa0b0, Pa0b1)  
p\_table\_prior <- data.frame(x,y)  
  
# create post table and look at both  
x <- c(sum(bvn[bvn[,2]== 0,1])/200,sum(bvn[bvn[,2]== 1,1])/200)  
y <- c((length(bvn[bvn[,2]== 0,1]) - sum(bvn[bvn[,2]== 0,1]))/200,   
 (length(bvn[bvn[,2]== 1,1]) - sum(bvn[bvn[,2]== 1,1]))/200)  
p\_table\_post <- data.frame(x,y)  
  
p\_table\_prior

## x y  
## 1 0.25 0.25  
## 2 0.25 0.25

p\_table\_post

## x y  
## 1 0.285 0.21  
## 2 0.255 0.25

#The distributions are not very different and will be closer with larger samples (LLN)

The distributions are not very different and will be closer with larger samples (LLN).

### 2

Suppose A and B are two binary random variables such that P(A=1,B=1) = .499, P(A=0,B=0)=.499, P(A=1,B=0)=.001, and P(A=0,B=1)=.001. Write a Gibbs sampler to draw 200 samples of (A,B). Do the results approximate the distribution of (A,B)? What might explain any difference from your answer to 3?

#define joint probabilities  
Pa1b0 <- .001  
Pa1b1 <- .499  
Pa0b1 <- .001  
Pa0b0 <- .499  
  
#Run sample for n=200  
bvn = gibbs(200,Pa1b0, Pa1b1, Pa0b1,Pa0b0)  
  
# create prior table  
x <- c(Pa1b0,Pa1b1)  
y <- c(Pa0b0, Pa0b1)  
p\_table\_prior <- data.frame(x,y)  
  
# create post table  
x <- c(sum(bvn[bvn[,2]== 0,1])/200,sum(bvn[bvn[,2]== 1,1])/200)  
y <- c((length(bvn[bvn[,2]== 0,1]) - sum(bvn[bvn[,2]== 0,1]))/200,   
 (length(bvn[bvn[,2]== 1,1]) - sum(bvn[bvn[,2]== 1,1]))/200)  
p\_table\_post <- data.frame(x,y)  
  
p\_table\_prior

## x y  
## 1 0.001 0.499  
## 2 0.499 0.001

p\_table\_post

## x y  
## 1 0.00 0.265  
## 2 0.73 0.005

The distribution does not approximate the distribution. The need for many more thousands of samples will better represent the prior distribution. This is due to the very small probability to jump to the other two values.

### 3

Drug A and Drug B are each cholesterol lowering medications. 10 patients took Drug A for a month, and their cholesterol levels were reduced by 11.46, 14.71, 9.52, 10.04, 9.78, 10.32, 9.88, 8.27, 10.07, and 13.62 points. 10 other patients took Drug B, and their cholesterol levels were reduced by 11.00, 10.25, 15.94, 14.10, 12.25, 12.91, 10.67, 12.58, 14.36, and 13.12 points. What is the probability that Drug B is on average more effective than Drug A? What is the 95% HDI for the difference in average effect of Drug B and Drug A?

Step 1: Assume that the Drug A data came from a normal distribution with mean μA and standard deviation σA. Assume that the Drug B data came from a normal distribution with mean μB and standard deviation σB.

Step 2: Assign identical independent normal prior distributions with mean 0 and standard deviation 100 to μA and μB. Assign identical independent uniform [0,10] priors to σA and σB. (These priors are very wide and reflect little prior information about the parameters.)

Step 3: Use JAGS to draw samples from the joint posterior distribution of μA and μB.

#### Hints:

Note that this assignment is very similar to the 2 coins example from class, except we now are comparing two means of normal distributions instead of two means of Bernoulli distributions. You can use the beta-Bernoulli code from class as a template, modifying the data preparation, model specification, initial value specification, parameter specification, etc. portions as appropriate.

In JAGS, the specification of a normal distribution is a little weird. Instead of telling JAGS the mean and standard deviation, you tell it the mean and “precision”. “Precision” means 1/variance. So if you want to say that Y is distributed according to a Normal distribution with mean mu and standard deviation sigma, you write Y ~ dnorm(mu, pow(sigma,-2)) where pow(sigma,-2) means 1/sigma2

Example 1 in this tutorial might be helpful to look at as well: <http://www.johnmyleswhite.com/notebook/2010/08/20/using-jags-in-r-with-the-rjags-package/>

source("C:/Users/usmc\_/Documents/Bayesian Analysis/DBDA2Eprograms/DBDA2E-utilities.R") # Must download DBDA2Eprograms from book's site and modify path to its location. Includes unique functions from the author such as diagMCMC for inspecting how well the models chains converged.

##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:  
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.  
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Warning: package 'rjags' was built under R version 3.4.3

## Warning: package 'coda' was built under R version 3.4.3

## Warning: package 'runjags' was built under R version 3.4.3

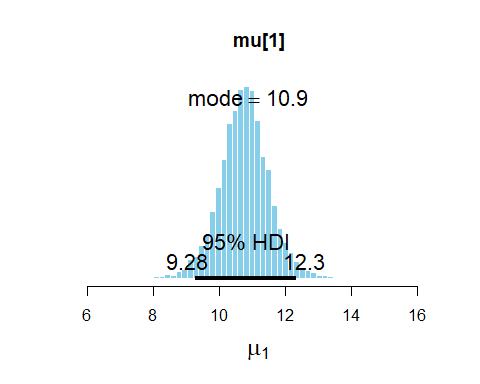
require(rjags) # Must have previously installed package rjags.  
  
# ----Load the data:----  
y = c(11.46, 14.71, 9.52, 10.04, 9.78, 10.32, 9.88, 8.27, 10.07, 13.62,   
 11.00, 10.25, 15.94, 14.10, 12.25, 12.91, 10.67, 12.58, 14.36, 13.12)  
s = c(rep(1,10),rep(2,10))  
Ntotal = length(y)  
N\_subjects = length(unique(s))  
dataList = list( # Put the information into a list.  
 y = y , s = s,  
 Ntotal = Ntotal, N\_subjects = N\_subjects   
)  
  
# ----Define the model:----  
modelString = "  
model {  
#Likelihood  
 for ( i in 1:Ntotal ) {  
 y[i] ~ dnorm( mu[s[i]],pow(sigma[s[i]],-2) )  
 }  
#Priors: Weakly informative with wide range  
 for(k in 1:N\_subjects){  
 mu[k] ~ dnorm(0,pow(100,-2))  
 sigma[k] ~ dunif( 0 , 10 )  
 }  
delta <- mu[s[1]]-mu[s[2]] #shortcut to find difference of two samples  
}" # close quote for modelString  
writeLines( modelString , con="TEMPmodel.txt" )  
  
# ----Run the chains:----  
jagsModel = jags.model( file="TEMPmodel.txt" , data=dataList ,   
 n.chains=3 , n.adapt=500 )

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 20  
## Unobserved stochastic nodes: 4  
## Total graph size: 55  
##   
## Initializing model

update( jagsModel , n.iter=500 )  
codaSamples = coda.samples( jagsModel , variable.names=c("mu","sigma", "delta") ,  
 n.iter=3334 )  
save( codaSamples , file="example\_Mcmc.Rdata")   
  
# ----Examine the chains:----  
# Convergence diagnostics:  
# diagMCMC is a function from the author of the book used in this course   
diagMCMC( codaObject=codaSamples , parName="mu[1]" )  
diagMCMC( codaObject=codaSamples , parName="mu[2]" )

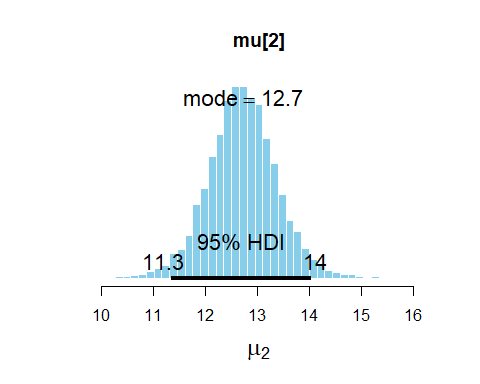
The chains have converged very well

all\_samps = do.call(rbind,codaSamples)   
# Posterior descriptives:  
plotPost( all\_samps[,"mu[1]"] , main="mu[1]" , xlab=bquote(mu[1]) )



## ESS mean median mode hdiMass hdiLow hdiHigh  
## mu[1] 10138.87 10.76408 10.77005 10.88607 0.95 9.282266 12.34001  
## compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE  
## mu[1] NA NA NA NA NA NA NA

plotPost( all\_samps[,"mu[2]"] , main="mu[2]" , xlab=bquote(mu[2]) )



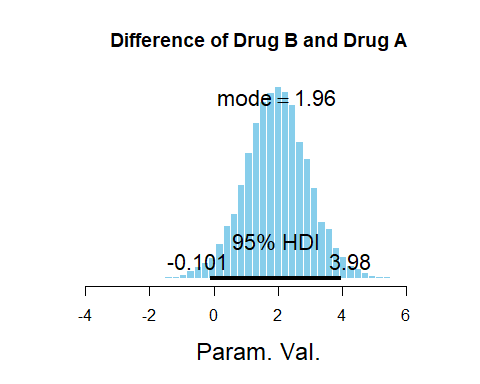
## ESS mean median mode hdiMass hdiLow hdiHigh  
## mu[2] 9674.311 12.71481 12.71225 12.72548 0.95 11.34314 14.03042  
## compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE  
## mu[2] NA NA NA NA NA NA NA

difference = all\_samps[,"mu[2]"] - all\_samps[,"mu[1]"]  
#What is the probability that Drug B is on average more effective than Drug A?  
sum(difference>0) / length(difference) #probability that Drug B is on average more effective than Drug A

## [1] 0.9703059

There is a probabitility of greater than .97 that Drug B on average performs better than Drug A.

plotPost(difference , main="Difference of Drug B and Drug A")



## ESS mean median mode hdiMass hdiLow hdiHigh  
## Param. Val. 10002 1.950729 1.950833 1.964115 0.95 -0.1009588 3.977097  
## compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE  
## Param. Val. NA NA NA NA NA NA NA