05-29-2023(2)

Hengde Ouyang

2023-05-29

4. Horseshoe Prior

Findings:

1.By changing the prior to horseshoe prior, the acceptance rate for beta decreases a lot.

2.Try many hyper-parameter taus, but hard to increase the acceptance rate.

The horseshoe prior is as follows (from Juho & AKi paper):

$$\beta_j|\lambda_j, \tau \sim N(0, \tau^2\lambda_j^2)$$

$$\lambda_j \sim C^+(0,1), \quad j = 1, ..., p$$

In our function, we define:

$$\sigma_j = \tau\lambda_j$$

```
MH_horseshoe_Sampling <- function(Y,delta,tau,
                        A,beta0,sigma0,var.prop,
                        m,B,eta,
                        Wmat_option=0){


  accept_beta = 0
  accept_lambda = 0
  beta = beta0
  lambda = lambda0
  sigma = sigma0

  # What we want to record
  BETA = matrix(0,m,dim(A)[2])
  LAMBDA = matrix(0,m,dim(A)[2])
  ThetaRecord <- matrix(0, m, length(Y))
  C_stat = c()
```

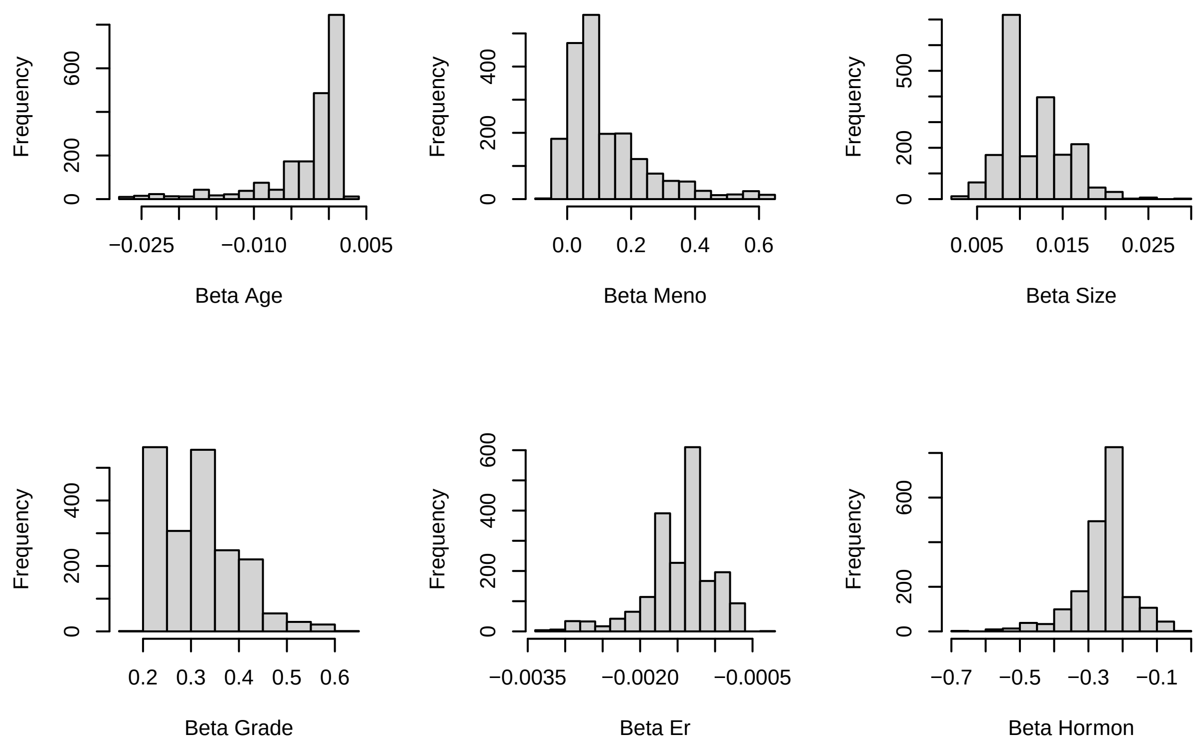
```
# For safety m>B
if (B>m){
  B = 0
}


# 0 means we use Harrell C statistics
# 1 means we use Uno C statistics
if (Wmat_option==0){
  Wmat <- HarrellC_Wmat(Y, delta, tau)
}else if (Wmat_option==1){
  Wmat <- UnoC_Wmat(Y, delta, tau)
}else{  # Other Possible C index...
  Wmat <- HarrellC_Wmat(Y, delta, tau)
}


for (i in 1:m){

  # Sample beta from proposal distribution
  beta.p = t(rmvnorm(1,beta,var.prop))


  # Compute theta from current and last iteration
  theta.p = THETA(A,beta.p)
  theta = THETA(A,beta)

  # Record theta from last iteration
  ThetaRecord[i,] <- theta

  # Compute C-statistics from current and last iteration
  HC.p = HarrellC(theta.p, Wmat)
  HC = HarrellC(theta, Wmat)

  # Record C-statistics from last iteration
  C_stat = c(C_stat,HC)


  #############################################
  # Compute log of MH ratio

  lrMH = eta*log(HC.p) +
        sum(dnorm(beta.p,beta0,sigma,log=T))-
        eta*log(HC) -
        sum(dnorm(beta,beta0,sigma,log=T))

    if (log(runif(1))<lrMH){
      beta = beta.p
      accept_beta = accept_beta + 1
    }
    BETA[i,] = beta
  #############################################
```

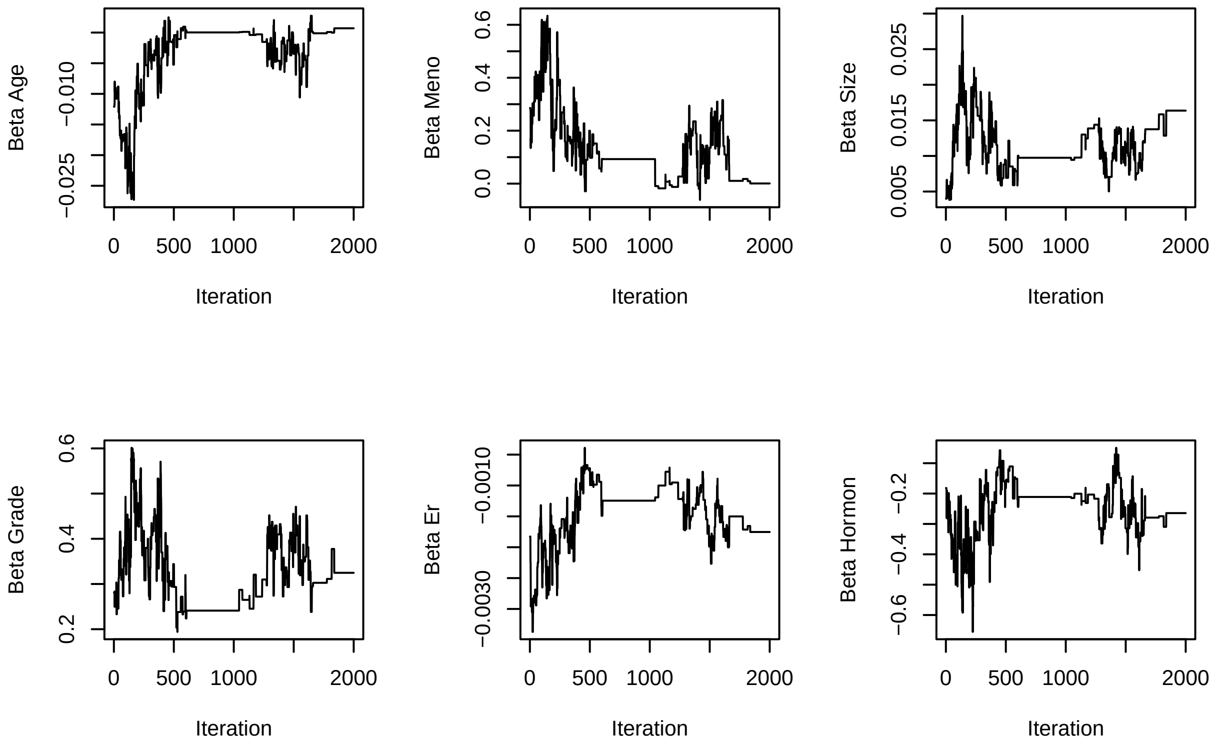
```
    ###############################################################
    # Compute log of MH_lambda ratio
    lambda.p = exp(t(rnorm(dim(A)[2],log(lambda),rep(1,dim(A)[2]))))
    sigma.p = lambda.p*beta_tau

    lrMH_lambda = sum(dnorm(beta,beta0,sigma.p,log=T))-
                  sum(dnorm(beta,beta0,sigma,log=T))

    if (log(runif(1))<lrMH_lambda){
        lambda = lambda.p
        sigma = sigma.p
        accept_lambda = accept_lambda + 1
      }
      LAMBDA[i,] = lambda
  }
  #####################################################################



  if (B == 0){
    return(list(BETA=BETA,
                LAMBDA = LAMBDA,
                accept_beta=accept_beta/m,
                accept_lambda=accept_lambda/m,
                THETA = ThetaRecord,
                C_stat = C_stat))
  }else{
    return(list(BETA=BETA[-c(1:B),],
                LAMBDA = LAMBDA[-c(1:B),],
                accept_beta=accept_beta/m,
                accept_lambda=accept_lambda/m,
                THETA = ThetaRecord[-c(1:B),],
                C_stat = C_stat[-c(1:B)]))
  }


}
```

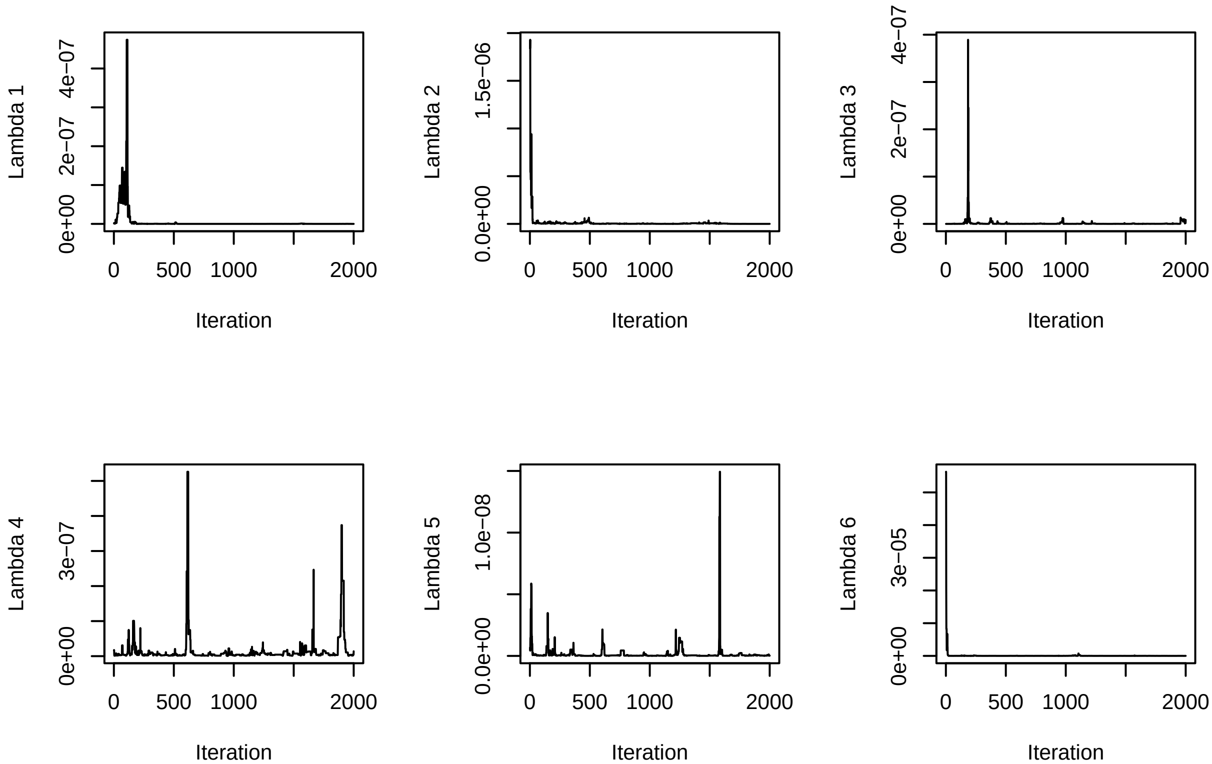
```
m = 2200
B = 200

beta0 = rep(0,dim(A)[2])
lambda0 = rep(1,dim(A)[2])
beta_tau = 100000000
sigma0 = beta_tau*lambda0

system.time({
result_horseshoe = MH_horseshoe_Sampling(Y,delta,tau,
                      A,beta0,sigma0,var.prop,
                      m,B,eta,
                      Wmat_option)
})
```

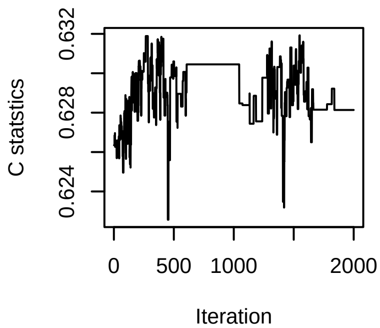
```
##    user  system elapsed
##   52.56   14.37  112.23
```

```
lambda_iter <- function(result,m,B){
  par(mfrow=c(2,3))
  plot(1:(m-B),result$LAMBDA[,1],xlab = "Iteration",
     ylab = "Lambda 1",type = "l")
  plot(1:(m-B),result$LAMBDA[,2],xlab = "Iteration",
     ylab = "Lambda 2",type = "l")
  plot(1:(m-B),result$LAMBDA[,3],xlab = "Iteration",
     ylab = "Lambda 3",type = "l")
  plot(1:(m-B),result$LAMBDA[,4],xlab = "Iteration",
     ylab = "Lambda 4",type = "l")
  plot(1:(m-B),result$LAMBDA[,5],xlab = "Iteration",
     ylab = "Lambda 5",type = "l")
  plot(1:(m-B),result$LAMBDA[,6],xlab = "Iteration",
     ylab = "Lambda 6",type = "l")

}
```




Beta Age
−0.025
−0.010
Beta Meno
0.0
0.2
0.4
0.6
Beta Size
0.005
0.015
0.025
Beta Grade
0.2
0.4
0.6
Beta Er
−0.0030
−0.0010
Beta Hormon
−0.6
−0.4
−0.2
0
500
1000
2000
Iteration


Lambda 1
0e+00
2e−07
4e−07
0
500
1000
2000
Iteration
Lambda 2
0.0e+00
1.5e−06
0
500
1000
2000
Iteration
Lambda 3
0e+00
2e−07
4e−07
0
500
1000
2000
Iteration
Lambda 4
0e+00
3e−07
0
500
1000
2000
Iteration
Lambda 5
0.0e+00
1.0e−08
0
500
1000
2000
Iteration
Lambda 6
0e+00
3e−05
0
500
1000
2000
Iteration




```r
# Acceptance Rate
result_horseshoe$accept_beta
```

```
## [1] 0.1313636
```

```r
result_horseshoe$accept_lambda
```

```
## [1] 0.2222727
```