PS3_MCMC

Boyu Chen

2023-06-09

Read Data

group data

```
data_list <- list()
for (i in 1:76) {
  filename <- paste0("./problem_set_3_sample/group/group", i, ".dat")
  data <- read.table(filename)
  data_list[[i]] <- data
}
names(data_list) <- paste0("g", 1:76)
```

network data

```
W <- list()
for (i in 1:76) {
  filename <- paste0("./problem_set_3_sample/network/network", i, ".dat")
  data <- read.table(filename)
  W[[i]] <- as.matrix(data)
}
names(W) <- paste0("W", 1:76)
```

Set N, X, Y

```
N <- sapply(1:76, function(i){nrow(W[[i]])})
X <- list()
Y <- list()
for (i in 1:76) {
  X[[i]] <- data_list[[i]][, c(1:6, 8:14)] %>% as.matrix()
  Y[[i]] <- data_list[[i]][, 18] %>% as.matrix()
}
```

```
col_names <- c('age', 'male', 'black', 'asian', 'hisp', 'race.other',
               'both.par', 'less.hs', 'more.hs', 'mom.edu.mis', 'welfare',
               'mom.job.miss', 'prof', 'job.other', 'sport', 'white',
               'yr.school', 'gpa', 'overage')
```

MCMC

Set init.param

```
T = 20000 # number of iterations during Markov process
# assign param in prior distribution
# beta k = 13 so 2k = 26
b_0 <- rep(0, 26)
B_0 <- diag(0.5, 26, 26)
# alpha
a_0 <- 5; A_0 <- 0.1
# sigma2
k_0 <- 0; v_0 <- 0
lambda_prime = 0

# saver
lambda_T <- rep(NA, T)
beta_T <- matrix(nrow = T, ncol = 26)
alpha_T <- matrix(nrow = T, ncol = 76)
sigma2_T <- rep(NA, T)
```

```
# starting value of draw
tao_G = sapply(1:76, function(g){W[[g]] %>% rowSums() %>% max()}) %>% max()
lambda_T[1] <- 0.0397
beta_T[1,] <- c(-0.1845, rep(0, 25))
alpha_T[1,] <- rep(5, 76)
sigma2_T[1] <- 0.5
```

Main Process

```
# propose lambda*
for (t in 2:T){
    accept = 0
    while (accept == 0){
        if (t<2){
            lambda_prime <- rnorm(1, lambda_T[t-1], 0.1^2 )
        }else if(t == 2){
            lambda_prime <- rnorm(1, lambda_T[t-1], 0) * 0.95 +
                rnorm(1, lambda_T[t-1], 0.1^2) * 0.05
        }else{
            lambda_prime <- rnorm(1,
                                  lambda_T[t-1],
                                  2.38^2 * var(lambda_T[1:t-1])) * 0.95 +
                rnorm(1, lambda_T[t-1], 0.1^2) * 0.05
        }
        accept = ifelse(
            lambda_prime >= -1/tao_G && lambda_prime <= 1/tao_G, 1, 0)
    }

    pp_l = 0
```

```r
for (g in 1:76){
    S_1 <- diag(N[g]) - lambda_prime * W[[g]] %>% as.matrix()
    S_2 <- diag(N[g]) - lambda_T[t-1] * W[[g]] %>% as.matrix()
    XX.g <- cbind(X[[g]], W[[g]]%*%X[[g]])
    ep_1 <- S_1 %*% Y[[g]] - XX.g %*% beta_T[t-1, ] - rep(1, N[g]) * alpha_T[t-1, g]
    ep_2 <- S_2 %*% Y[[g]] - XX.g %*% beta_T[t-1, ] - rep(1, N[g]) * alpha_T[t-1, g]
    like_1 <- log(det(S_1)) - 0.5 * sigma2_T[t-1]^(-1) * t(ep_1) %*% ep_1
    like_2 <- log(det(S_2)) - 0.5 * sigma2_T[t-1]^(-1) * t(ep_2) %*% ep_2
    pp_l = pp_l + like_1 - like_2
}
pp_l = min(c(exp(pp_l), 1))
lambda_T[t] = ifelse(runif(1) <= pp_l, lambda_prime, lambda_T[t-1])

# THE SAMPLING OF BETA FROM PROSTERIOR DISTRIBUTION
B <- sigma2_T[t-1]^-1 * Reduce("+",  lapply(1:76, function(g){
    XX.g <- cbind(X[[g]], W[[g]]%*%X[[g]])
    return(t(XX.g) %*% XX.g)
}))
B <- solve(solve(B_0) + B)

beta_hat <- lapply(1:76, function(g){
    S.g <- diag(N[g]) - lambda_T[t] * W[[g]] %>% as.matrix()
    XX.g <- cbind(X[[g]], W[[g]]%*%X[[g]])
    Y.g <- Y[[g]]
    l.g <- rep(1, N[g])
    a.g <- alpha_T[t-1]
    return(t(XX.g) %*% (S.g%*%Y.g - l.g*a.g))
})
beta_hat <- sigma2_T[t-1]^-1 * Reduce("+", beta_hat)
beta_hat <- B %*% (solve(B_0)%*%b_0 + beta_hat)

beta_T[t,] <- mvrnorm(n=1, mu=beta_hat, Sigma=B)

# THE SAMPLING OF SIGMA_E^2 FROM PROSTERIOR DISTRIBUTION
sum_ep.gTep.g <- sapply(1:76, function(g){
    S.g <- diag(N[g]) - lambda_T[t] * W[[g]] %>% as.matrix()
    Y.g <- Y[[g]]
    XX.g <- cbind(X[[g]], W[[g]]%*%X[[g]])
    l.g <- rep(1, N[g])
    a.g <- alpha_T[t-1]
    beta <- beta_T[t,]
    ep.g <- S.g %*% Y.g - XX.g %*% beta - l.g*a.g
    return(t(ep.g)%*%ep.g)
}) %>% sum()

sigma2_T[t] <- 1/rgamma(1, (k_0 + sum(N))/2, (v_0+ sum_ep.gTep.g)/2)

# THE SAMPLING OF ALPHA_G FROM PROSTERIOR DISTRIBUTION
for (g in 1:76){
    sigma2 <- sigma2_T[t]
    l.g <- rep(1, N[g])
    S.g <- diag(N[g]) - lambda_T[t] * W[[g]] %>% as.matrix()
    Y.g <- Y[[g]]
```

```
        XX.g <- cbind(X[[g]], W[[g]] %*% X[[g]])
        beta <- beta_T[t,]
        cbind(X[[g]], W[[g]]%*%X[[g]])
        R.g <- (A_0^-1 + sigma2^-1 * t(l.g) %*% l.g) ^-1
        a.g_hat <- R.g * (A_0^-1 * a_0 + sigma2^-1 * t(l.g)%*%(S.g%*%Y.g-XX.g%*%beta))
        alpha_T[t, g] <- rnorm(1, a.g_hat, R.g)
    }
}
beep(4)
```

Posterior Results

```
posterior_result <- list()
posterior_result$alpha_mean <- sapply(1:76, function(g){mean(alpha_T[1000:T,g])})
posterior_result$alpha_sd <- sapply(1:76, function(g){sd(alpha_T[1000:T,g])})
posterior_result$beta_mean <- sapply(1:26, function(k){mean(beta_T[1000:T,k])})
posterior_result$beta_sd <- sapply(1:26, function(k){sd(beta_T[1000:T,k])})
posterior_result$lambda_mean <- mean(lambda_T)
posterior_result$lambda_sd <- sd(lambda_T)
posterior_result$sigma2_mean <- mean(sigma2_T)
posterior_result$sigma2_sd <- sd(sigma2_T)

var.name <- c("lambda", paste0("b1.", 1:13), paste0("b2.", 1:13), paste0("a", 1:76), "sigma2")
var.mean <- c(posterior_result$lambda_mean, posterior_result$beta_mean,
          posterior_result$alpha_mean, posterior_result$sigma2_mean)
var.sd <- c(posterior_result$lambda_sd, posterior_result$beta_sd,
          posterior_result$alpha_sd, posterior_result$sigma2_sd)
df <- data.frame(cbind(var.name, var.mean, var.sd))
kable(df)
```

var.name	var.mean	var.sd
lambda	-0.0708958188749159	0.011578695077608
b1.1	-0.113636536783257	0.0046973853557212
b1.2	-0.148909219225346	0.0360869880498838
b1.3	-0.199537572402767	0.0507140818443512
b1.4	-0.0460699879899416	0.134117515988765
b1.5	0.0496113877001371	0.0751066300024922
b1.6	-0.150374972720021	0.0691907880190834
b1.7	-0.216954712914122	0.0604618457787536
b1.8	0.174889464827976	0.0396130237164799
b1.9	-0.0433632171605844	0.0636590791470198
b1.10	0.131720123305804	0.182896976755592
b1.11	-0.099318033690457	0.0639062701312967
b1.12	0.0171391021655984	0.0486680970738496
b1.13	0.00164205605835822	0.0425084396551652
b2.1	0.0199095355945588	0.00174995795527866
b2.2	-0.0538830657275737	0.0222112781358359
b2.3	-0.0509081897263153	0.016164097816729
b2.4	-0.075297064263207	0.107851417075092
b2.5	-0.0825183468104169	0.0423973866020547
b2.6	-0.102409445264895	0.0375178383216341
b2.7	-0.145776827312778	0.0381819874445837
b2.8	0.0365023362664575	0.022820090959106
b2.9	0.0195047451552801	0.0454676723539597
b2.10	0.0077660057195312	0.143717164718548
b2.11	-0.106756046224223	0.04243795742651
b2.12	-0.0179375318072978	0.0279012561786054
b2.13	-0.00818919737905385	0.0225802664178183
a1	4.67459553387252	0.0540871395490367
a2	4.82652780012049	0.0511456386803611
a3	4.43866557417013	0.0578056347738494
a4	5.21690974829629	0.0654109832480609
a5	4.32122266163588	0.0541156090090815
a6	4.58559852061833	0.0555570970559378
a7	4.42745651513563	0.0513587302975708
a8	4.85579840340878	0.0536052595253195
a9	4.66332550755543	0.0590497331730744
a10	4.68925786505001	0.0596684923707994
a11	4.4961452269442	0.0586524432207756
a12	4.56141757247287	0.0593243694762128
a13	4.76040576757184	0.0610105811895935
a14	4.84876275115578	0.0662052517163438
a15	4.73743665305983	0.0485945087515817
a16	4.81256973155575	0.0609008507800622
a17	4.94443035679038	0.0649012292101993
a18	4.98242507673871	0.0581378949933872
a19	4.85909331418388	0.0560705583797072
a20	5.12109887984329	0.0653139867152812
a21	4.57708646171657	0.0532263552477426
a22	4.40632211725274	0.0541768679037777
a23	4.80675435393438	0.054971798447883
a24	4.71405002034926	0.057322344202737
a25	5.01387319450504	0.0607983799239624
a26	4.58707640721942	0.0691921669367336
a27	4.45760698573691	0.0717463859025091
a28	4.77675910440274	0.0611635359256394
a29	4.98044308388652	0.0655498978890541
a30	5.10166794232814	0.0715006832326935
a31	[illegible]	[illegible]