GLMM1997

Congyuan Duan

December 30, 2021

[10]:
```
load("C:/personal/R work/learning/homework/statistics computation/final.RData")
```

The full code is in another R file. Here I only explain some of the key steps.

Data is generated from a logit-norm model

$$
\begin{aligned}
&Y_{ij}|u \sim Bernoulli(p_{ij}) \quad i = 1, \cdots, n \quad j = 1, \cdots, q \\
&ln(p_{ij}/(1-p_{ij})) = \beta x_{ij} + u_j \\
&u_j \sim iidN(0, \sigma^2)
\end{aligned}
$$

with the true value $\beta = 5$, $\sigma^2 = 0.5$, $x_{ij} = i/15$, $n = 15$ and $q = 10$.

[4]:
```
# data generation
u <- rnorm(q, 0, sigma)
X <- p <- Y <- matrix(0, ncol = q, nrow = n)
for (j in 1:q) {
    X[, j] <- c(1:n) / n
    p[, j] <- exp(beta * X[, j] + u[j]) / (1 + exp(beta * X[, j] + u[j]))
}
for (i in 1:n) {
    for (j in 1:q) {
        Y[i, j] <- rbinom(1, 1, p[i, j])
    }
}
```

In either MCEM or MCNR, we use Metropolis algorithms to generate $N = 10000$ values of $u_1, \cdots, u_q$. The candidate distribution is the marginal distribution of u_j, that is, $N(0, \sigma^2)$. Then the acceptence rate is

$$
A_j(u, u^*) = \min\left\{1, e^{y_{+j}(u_j^* - u_j)} \prod_i \frac{1 + e^{\beta x_{ij} + u_j}}{1 + e^{\beta x_{ij} + u_j^*}}\right\}
$$

In the simulation study, we set the first $N_1 = 500$ brun-in and use only the last $N - N_1$ samples.

[5]:
```
# Metropolis
tempu <- matrix(0, ncol = q, nrow = N)
for (t in 2:N) {
    for (j in 1:q) {
        uni <- runif(1, 0, 1)
```

```
        uj <- rnorm(1, 0, sqrt(sigma0[m]))
        A <- min(1, exp(sum(Y[, j]) * (uj - tempu[t-1, j])) *␣
↪prod((1+exp(beta0[m]*X[,1]+tempu[t-1, j]))
                                                                    /
↪(1+exp(beta0[m]*X[,1]+uj)))) #xij is same for j
        if (uni < A) {tempu[t, j] <- uj}
        else {tempu[t, j] <- tempu[t-1, j]}
    }
}
```

In MCEM, The function maxbeta find the optimal β to maximize the function

$$\frac{1}{N}\sum_{k=1}^{N}\beta\sum_{i,j}y_{ij}x_{ij}+\sum_{j}y_{+j}u_j^{(k)}-\sum_{i,j}\log(1+\exp(\beta x_{ij}+u_j^{(k)}))$$

through searching.(page 165)

[6]:
```
maxbeta <- function(beta, tempu) {
  q1 <- q2 <- q3 <- 0
  q1 <- beta * sum(Y * X)
  # for (t in (N1+1):N) { q2 has no relation with beta
  #   q2 <- q2 + matrix(apply(Y, 2, sum), nrow = 1) %*% matrix(tempu[t, ], ncol␣
↪= 1)
  # }
  # q2 <- q2 / (N - N1)
  for (t in (N1+1):N) {
    q3 <- q3 + sum(log(1 + exp(beta * X + matrix(tempu[t, ], nrow = n, ncol =␣
↪q, byrow = TRUE))))
  }
  q3 <- q3 / (N - N1)
  #return(q1 + q2 - q3)
  return(q1 - q3)
}
l <- -1000
for (b in seq(0, 10, by = 0.1)) {
    if (maxbeta(b, tempu) > l) {
        beta0[m+1] <- b
        l <- maxbeta(b, tempu)
    }
}
```

In MCNR, we use Newton-Raphson iteration to get β

$$\beta^{(m+1)}=\beta^{(m)}+E[X'W(\beta^{(m)},U)X|y]^{-1}X'(y-E[\mu(X'W(\beta^{(m)},U)|y])$$

where $\mu_i(\beta,u)=1/(1+\exp\{-\beta x_{ij}-u_j\})$, $W(\beta,u)=diag\{\mu_i(\beta,u)(1-\mu_i(\beta,u))\}$.

```r
funcW <- function(beta, tempu) {
  W <- matrix(0, nrow = n, ncol = n)
  mu <- matrix(0, ncol = 1, nrow = n)
  for (i in 1:n) {
    mu1 <- mean(1 / (1 + exp(-beta * X[i,1] - tempu)))
    mu[i, 1] <- mu1
    W[i, i] <- mu1 * (1 - mu1)
  }
  return(list(W = W, mu = mu))
}
W <- funcW(beta0[m], tempu[((N1+1):N),])
X1 <- matrix(X[,1], nrow = n)
beta0[m+1] <- beta0[m] + 1 / (t(X1) %*% W$W %*% X1) * (t(X1) %*% matrix(apply(Y,1,mean) - W$mu, ncol = 1))
```

For both MCEM and MCNR, the update for σ^2 is

$$\sigma^{2(m+1)} = \frac{1}{Nq}\sum_{k=1}^{N}(\sum_{j=1}^{q} u_j^{(k)2})$$

```r
sigma0[m+1] <- 1 / (N - N1) * sum(tempu[((N1+1):N),]^2 / q)
```

For both methods, we do $M = 50$ replications, and caculate the estimated mean, variance and MSE. The results are as follows

```r
result <- data.frame(MCEM = c(mean(betafinal1), var(betafinal1), mean((betafinal1 - beta)^2),
                            mean(sigmafinal1), var(sigmafinal1), mean((sigmafinal1 - sigma^2)^2)),
                     MCNR = c(mean(betafinal2), var(betafinal2), mean((betafinal2 - beta)^2),
                            mean(sigmafinal2), var(sigmafinal2), mean((sigmafinal2 - sigma^2)^2)))
rownames(result) <- c("beta_mean", "beta_var", "beta_MSE", "sigma_mean", "sigma_var", "sigma_MSE")
```

```r
result
```

A data.frame: 6 × 2

	MCEM <dbl>	MCNR <dbl>
beta_mean	5.12800000	5.11066929
beta_var	0.80450612	0.80533352
beta_MSE	0.80480000	0.80147454
sigma_mean	0.24537578	0.22329725
sigma_var	0.07939811	0.08388519
sigma_MSE	0.14264364	0.15877190

The result are very similar. The estimate for sigma seems not very well. However, if I run MCEM for 100 replications, the estimate of sigma will be 0.565. For time reason, until I hand in the homework, I have not run the MCNR for 100 replications yet. I think it will also be much closer to the true value 0.5 if there are more replications.