2018-2 Computational Statistics

(Markov Chain) Monte Carlo EM algorithm using Important Sampling

김미성

이화여자대학교 통계학과

Ⅰ. Introduction

MCEM(Monte Carlo EM) 알고리즘은 EM 알고리즘의 E 단계에서 Monte Carlo simulation 을 통해 기댓값을 추정하는 방법이다. 이 때 Monte Carlo simulation 의 구체적인 방법은 다양하다. 특히 target density 로부터 independent sample 을 얻기 어려운 경우 Markov Chain Monte Carlo (MCMC)를 통해서 dependent sample 을 얻어 활용할 수 있다. Richard A. Levine 과 George Casella (2001)는 "Implementations of the Monte Carlo EM Algorithm"이라는 논문에서 MCMC 를 이용하여 MCEM 알고리즘을 적용하는 과정에서 important sampling 을 응용하여 계산 비용을 줄이는 방법 및 Monte Carlo 추정을 함으로써 추가적으로 발생하는 오차, 즉 Monte Carlo error 를 측정하여 sample size 를 결정하는 방법을 제시하였다. 또한 Richard A. Levine 과 Juanjuan Fan (2004) 은 "An automated (Markov chain) Monte Carlo EM algorithm"이라는 논문에서 이를 발전시켜 Monte Carlo sample size 를 자동으로 결정하는 방법을 제시하였다.

이 프로젝트에서는 위의 두 논문을 바탕으로 하여 important sampling 을 이용한 Markov Chain MCEM 알고리즘을 구현해보고자 한다. 2 장에서는 위의 방법들에 대하여 구체적으로 설명할 것이며, 3 장에서는 missing value 가 포함된 Bivariate Normal 분포의 모수 추정 문제 및 위의 두 논문에 공통적으로 나오는 salamander mating data 를 이용한 Probit Normal Model 의 모수 추정 문제에 이를 적용해보도록 하겠다.

Ⅱ. (Markov Chain) Monte Carlo EM algorithm

MCEM 알고리즘에서는 t 번째 iteration 의 E 단계를 다음과 같이 쓸 수 있다.

$$\hat{Q}^{t+1}(\theta|\theta^{(t)}) = \left(\frac{1}{\mathrm{m}^{(t)}}\right)\sum_{j=1}^{m^{(t)}} log f_Y(Y_j^{(t)}|\theta)$$

이 때 $Y_j^{(t)} = (x, Z_j^{(t)})$ 이고, $Z_1^{(t)}, \dots, Z_{m^{(t)}}^{(t)}$는 observaed data($x$)와 모수($\theta$)가 주어졌을 때의 조건부 분포 $g(z_j|x,\theta^{(t)})$로부터 추출한 sample 이며, $Y_j = (x, Z_j)$는 observed data(x)와 missing data(Z_j)를 포함한 complete data 를 의미한다. 또한 $\mathrm{m}^{(t)}$는 t 번째 iteration 에서의 sample size 를 의미한다. 그러나 MCEM 알고리즘의 경우 모든 iteration 에서 sampling 을 해야 하므로 계산 비용이 크다는 단점이 있으며, 각각의 iteration 에서 sample size 를 어떻게 결정할 것인지도 고려해야 한다. 한편 target density 인 g 로부터 Z_j 를 sampling 하기 어려운 경우에는 Markov chain 을 이용하여 sampling 할 수 있다. 이 프로젝트에서는 앞서 언급한 두 가지를 고려하여 MCMC 에 기반한 MCEM 알고리즘을 구현하고자 한다.

1) Important Sampling

첫번째로 계산 비용이 크다는 문제에 대하여 Richard A. Levine 과 George Casella(2001)는 다음과 같이 standardized important sampling 을 응용한 E 단계 추정 방법을 제시하였다.

$$\hat{Q}^{t+1}(\theta|\theta^{(t)}) = \sum_{j=1}^{m^{(t)}} \{\log f_Y(Y_j^{(0)}|\theta) * w_j\} \Big/ \sum_{j=1}^{m^{(t)}} w_j,$$

$$where\ Y_j^{(0)} = (x, Z_j^{(0)}), \quad w_j = \frac{g(z_j|x,\theta^{(t)})}{g(z_j|x,\theta^{(0)})}$$

이처럼 초기에 추출한 sample $Z_j^{(0)}$에 weight 를 부여하여 기댓값을 추정하면, 모든 iteration 에서 다시 sampling 을 하지 않고도 새로운 Monte Carlo 추정치를 구할 수 있다. 이 때 $\theta^{(0)}$는 모수의 초기치이며, $\theta^{(t)}$ 는 가장 최근의 iteration 에서 얻은 모수의 추정치이다. 그러나 초기 EM 단계에서는 θ 의 추정치가 참값에 가깝지 않을 수 있고, 이에 important sampling 을 적용하여 얻은 sample 을 이용할 경우 추정의 정확도가 떨어질 수 있다. 따라서 초기 몇 번의 iteration 동안에는 작은 sample size 로 새로운 sample 을 반복하여 뽑고, 이를 통해 초기값을 재설정한 후에 important sampling 을 적용해야 한다.

2) Decision of Sample Size

두번째로 sample size 를 결정하는 문제에 대하여 Richard A. Levine 과 Juanjuan Fan(2004)는 Monte Carlo error 를 계산하여 자동으로 sample size 를 결정하는 방법을 제시하였다. EM 알고리즘에서는 t 번째 iteration 에서의 추정치와 (t+1) 번째 iteration 에서의 추정치 사이에 나타나는 오차, 즉 EM step 에 의한 오차가 존재하는데, MCEM 알고리즘에서는 이에 더해 E 단계에서 Monte Carlo 추정을 함으로써 발생하는 오차가 추가적으로 존재한다. 따라서 Monte Carlo 추정에 의한 오차가 EM step 에 의한 오차보다 커지면 sample size 를 늘려 Monte Carlo error 를 작게 만들어야 한다.

그런데 MCMC sample 은 dependent sample 이기 때문에 Central Limit Theory(CLT)를 적용하기 어려워 오차를 계산하기 어렵다는 단점이 있다. 따라서 Poisson subsampling scheme 을 통해 근사적으로 독립인 subsample 을 얻은 후에 이를 통해 오차를 구할 수 있다. Poisson subsampling scheme 은 m 개의 sample 에서 다음과 같이 N_m 개의 point 를 설정하여 subsample 을 얻는 방법이다.

$$t_k = x_1 + \cdots + x_k, \qquad where\ x_k - 1 \sim Poisson(v_k),$$
$$k = 1, \ \ldots, N_m, \qquad N_m = \sup\{n: t_n \le m\}$$

이러한 방법으로 얻은 subsample $Z_{t_1}, \ \ldots, Z_{t_{N_m}}$ 은 근사적으로 독립이며(Robert et al., 1999), 따라서 다음과 같이 근사 CLT 를 적용할 수 있다.(Booth and Hobert, 1999)

$$\sqrt{N_m}\left(\theta_{N_m}^{(t+1)} - \hat{\theta}^{(t+1)}\right) \div N(0, \hat{\Sigma})$$

위에서 $\theta^{(t)}$, $\theta_{N_m}^{(t)}$, $\hat{\theta}^{(t)}$는 각각 t 번째 iteration 에서의 Monte Carlo 추정치, subsample 을 이용한 Monte Carlo 추정치, sampling 을 하지 않고 구한 추정치를 의미하며, 따라서 (t+1)번째 iteration 에서 Monte Carlo 추정에 의해 발생한 오차를 의미하는 $\hat{\Sigma}$ 에 대해서는 다음과 같은 근사치를 얻을 수 있다.

$$\hat{\Sigma} \approx \left\{\hat{Q}_{N_m}^{(2)}\left(\theta^{(t+1)}|\theta^{(t)}\right)\right\}^{-1}\left[\frac{1}{m}\sum_{j=1}^{m^{(t)}}\left\{\frac{\partial}{\partial\theta} log\, f_Y\left(Y_j^{(t+1)}|\theta\right) - \hat{\mu}_m\right\}\left\{\frac{\partial}{\partial\theta} log\, f_Y\left(Y_j^{(t+1)}|\theta\right) - \hat{\mu}_m\right\}^T\right]$$
$$\times\left\{\hat{Q}_{N_m}^{(2)}\left(\theta^{(t+1)}|\theta^{(t)}\right)\right\}^{-1}|_{\theta=\theta^{(t+1)}}$$
$$, where\ \hat{Q}_{N_m}^{t+1}\left(\theta|\theta^{(t)}\right) = \left(\frac{1}{N_m}\right)\sum_{k=1}^{N_m} log\, f_Y(Y_{t_k}^{(t)}|\,\theta)$$

따라서 (t+1)번째 iteration 에서 Monte Carlo 추정에 의해 발생한 오차의 (1-α) 신뢰영역을 다음과 같이 쓸 수 있다.

$$N_m\left(\theta_{N_m}^{(t+1)} - \hat{\theta}^{(t+1)}\right)^T\hat{\Sigma}^{-1}\left(\theta_{N_m}^{(t+1)} - \hat{\theta}^{(t+1)}\right) \le \chi^2_{d;1-\alpha}$$

만약 t 번째 iteration 에서 얻은 Monte Carlo 추정치 $\theta^{(t)}$가 위의 신뢰영역에 포함된다면, 이는 EM step 에 의한 error($\theta_{N_m}^{(t+1)} - \theta^{(t)}$)가 Monte Carlo 추정에 의해 발생하는 error($\theta_{N_m}^{(t+1)} - \hat{\theta}^{(t+1)}$)보다

작다는 것을 의미한다. 이를 바꾸어 말하면 Monte Carlo 추정에 의한 error 가 EM step 에 의한 error 보다 크다는 의미이므로 sample size 를 증가시켜 Monte Carlo error 를 줄여야 한다.

sample size 를 얼만큼 증가시킬 것인가에 대해서는 Monte Carlo error 와 EM step 에 의한 error 가 같아지는 경우, 즉 $\theta^{(t)}$가 위의 신뢰영역의 경계에 위치할 때의 sample size 를 구함으로써 필요한 sample size 의 최소치를 얻을 수 있다. 이 때 먼저 최소한의 subsample size 인 N_m을 구할 수 있으며, 이를 통해 필요한 sample size 인 m 의 값을 구할 수 있을 것이다.

Ⅲ. Examples

1) Bivariate Normal with Missing Values

다음과 같이 결측치가 포함된 data 로부터 Bivariate Normal 분포를 가정하고 EM 알고리즘을 적용하여 모수를 추정하고자 한다. Monte Carlo 추정에는 MCMC 기법 중 하나인 Gibbs Sampling 을 이용하였다.

data)

$\boldsymbol{W_1}$	8	11	16	18	6	4	20	25	9	13
$\boldsymbol{W_2}$	10	14	16	15	20	4	18	22	?	?

model)

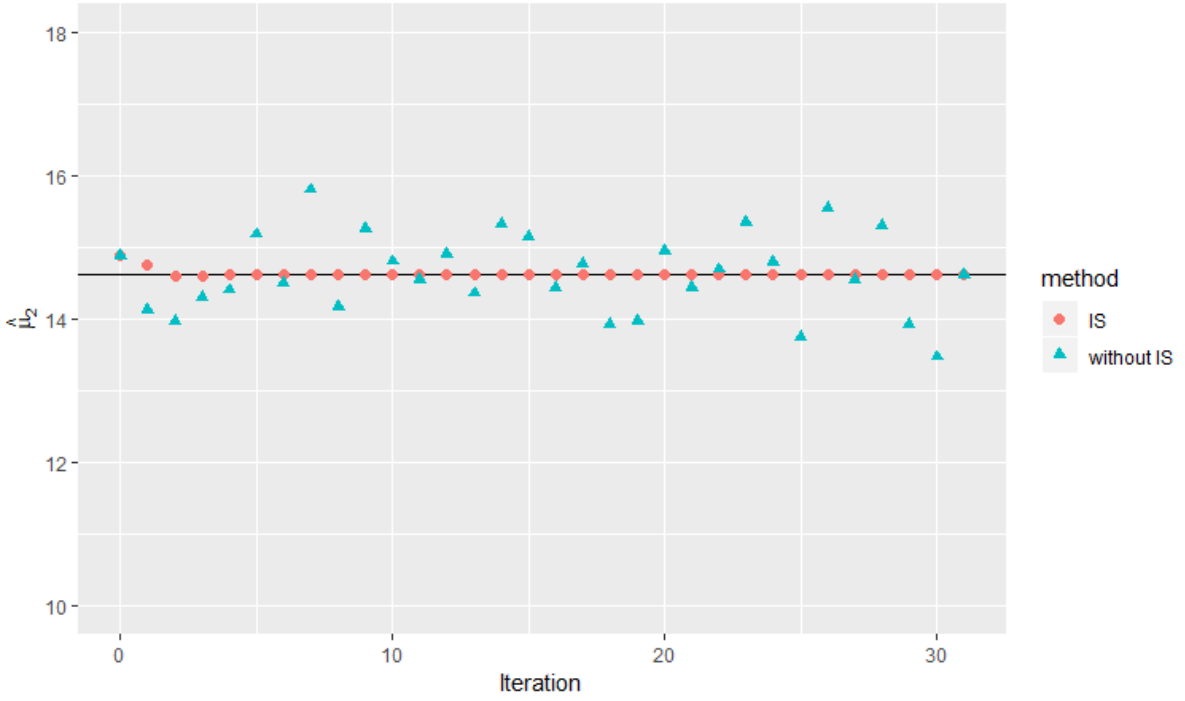
$$\mathbf{W} = \begin{pmatrix} \boldsymbol{W_1} \\ \boldsymbol{W_2} \end{pmatrix} \sim \mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \begin{cases} \boldsymbol{\mu} = (\boldsymbol{\mu_1}, \boldsymbol{\mu_2})' \\ \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\sigma_{11}} & \boldsymbol{\sigma_{12}} \\ \boldsymbol{\sigma_{12}} & \boldsymbol{\sigma_{22}} \end{pmatrix} \end{cases}$$

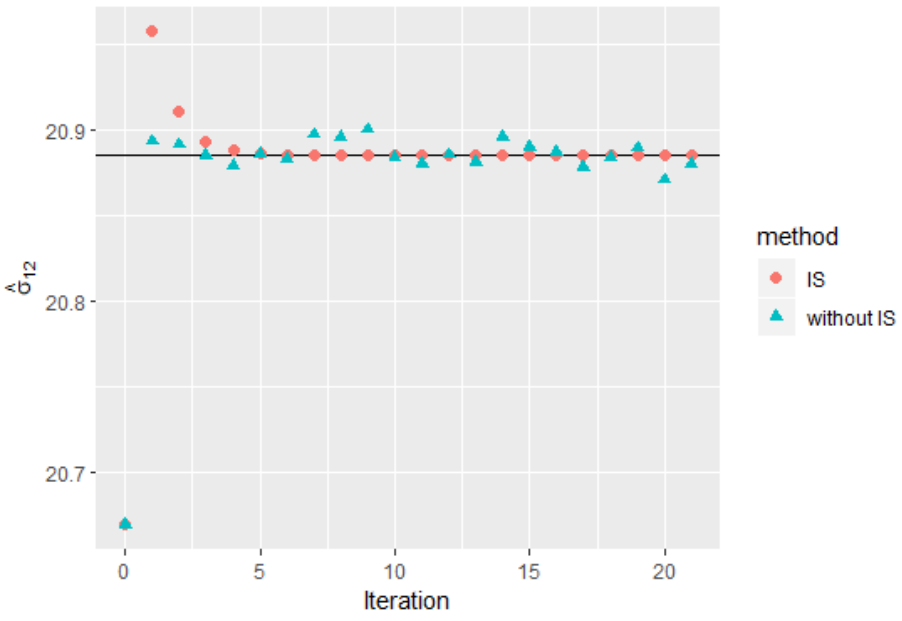
$$\boldsymbol{\theta} = (\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \boldsymbol{\sigma_{11}}, \boldsymbol{\sigma_{12}}, \boldsymbol{\sigma_{22}})'$$

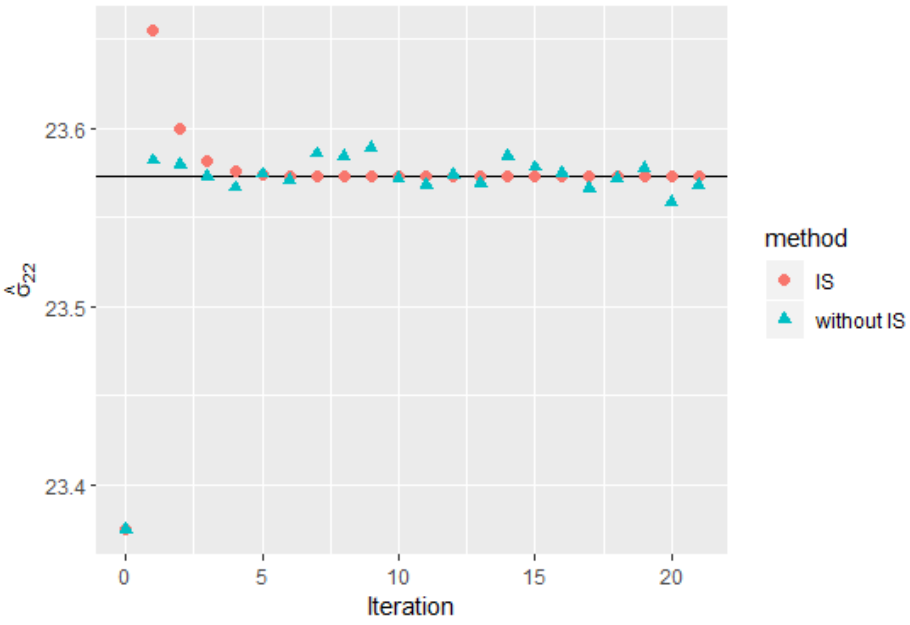
이 문제의 경우, Monte Carlo simulation 을 하지 않아도 EM 알고리즘을 이용하여 MLE 를 추정할 수 있다. 따라서 이 문제에 위의 알고리즘을 적용하여 MLE 를 추정해보고 MC simulation 없이 구한 결과와 비교해보고자 한다. 또한 important sampling 을 이용하지 않고 모든 iteration 에서 Monte Carlo sampling 을 시행하여 추정한 결과도 함께 비교해보았다. 결과는 다음과 같았다.

	Important Sampling	Without Important Sampling	Original EM
$\widehat{\mu_2}$	14.61010	14.61337	14.61253
$\widehat{\sigma_{12}}$	20.87033	20.85495	20.88516
$\widehat{\sigma_{22}}$	23.55883	23.54358	23.57334
time (minute)	0.24	0.76	

추정치는 Monte Carlo simulation 을 하지 않고 EM 알고리즘을 사용하여 구한 결과와 거의 일치하였고(소수점 첫째자리까지 같다.), 계산 시간을 비교해본 결과는 important sampling 을 이용하여 구했을 때 3 배 정도 빠른 것을 확인할 수 있었다. 또한 다음 graph 들에서 알 수 있듯이, important sampling 을 이용하여 추정했을 때에는 모든 iteration 에서 Monte Carlo simulation 을 하였을 때보다 더 적은 iteration 으로 수렴하는 것을 알 수 있었다.







2) Probit-Normal Linear Mixed Model with Salamander data

			Date			
Females	June 4	June 8	June 12	June 16	June 20	June 24
1	1	1	1	0	1	1
2	1	1	1	1	1	1
RBF 3	RB 1	WS 0	RB 1	WS 1	RB 1	WS 1
4	1	1	1	0	1	1
5	1	1	1	1	1	1
6	1	1	1	0	1	1
7	0	0	0	1	0	0
RBF 8	WS 0	RB 1	WS 0	RB 0	WS 1	RB 1
9	0	0	1	1	1	1
10	0	0	1	0	1	0
1	0	1	1	1	0	1
2	0	0	0	1	0	0
WSF 3	RB 0	WS 0	RB 0	WS 0	RB 0	WS 1
4	0	1	1	1	0	1
5	0	1	0	0	0	0
6	0	0	1	0	0	0
7	1	1	1	0	1	1
WSF 8	WS 1	RB 0	WS 1	RB 0	WS 1	RB 0
9	1	1	1	1	1	0
10	1	0	0	1	1	0

Salamander data 는 도룡뇽의 짝짓기 성공 여부를 나타내는 data 이며, 도룡뇽의 type 은 rough-butt(RB)과 whiteside(WS)의 2 가지가 있다. 따라서 cross type 은 RB/RB, RB/WS, WS/RB, WS/WS 의

4 가지가 존재하며, 위와 같이 20 쌍의 도룡뇽을 대상으로 6 번을 실험하여 총 120 번의 짝짓기를 관찰하였다. 암수 도룡뇽은 각각은 RB-type 10 마리와 WS-type 10 마리로 이루어져 있으며, RBF 는 RB-type 의 암컷 도룡뇽, WSF 는 WS-type 의 암컷 도룡뇽을 의미한다. 짝짓기의 성공 여부를 나타내는 변수를 W 라고 할 때, 성공 확률 y 를 예측하는 모델은 다음과 같다.

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}_{\mathrm{f}}\mathbf{u}_{\mathrm{f}} + \mathbf{Z}_{\mathrm{m}}\mathbf{u}_{\mathrm{m}} + \varepsilon; \quad w_i = I(y_i \geq 0)$$

$$\mathbf{u}_{\mathrm{f}} \sim N_{20}(\mathbf{0}, \sigma_{\mathrm{f}}^2\mathbf{I}); \quad \mathbf{u}_{\mathrm{m}} \sim N_{20}(\mathbf{0}, \sigma_{\mathrm{m}}^2\mathbf{I}); \quad \varepsilon \sim N_n(\mathbf{0}, \mathbf{I})$$

위 식에서 $\beta = (\beta_{RR}, \beta_{RW}, \beta_{WR}, \beta_{WW})^T$는 cross type 에 대한 fixed effect 를 나타내며, $\mathrm{u_f}$와 $\mathrm{u_m}$은 각각 암컷과 수컷의 random effect 를 나타낸다. X, $\mathrm{Z_f}$, $\mathrm{Z_m}$은 각각 cross type, 암컷의 식별 번호, 수컷의 식별 번호를 나타내는 design matrix 이다. 이 모델에서 EM 알고리즘을 통해 모수의 MLE 를 추정하기 위해 다음과 같이 Gibbs sampling 을 이용하여 y 를 sampling 한 뒤 complete data 를 이용하였다.

$$\begin{aligned}
&Y_1^{(b+1)} \text{ from } f(y_1 \mid y_2^{(b)}, y_3^{(b)}, \ldots, y_n^{(b)}, \mathbf{W}; \boldsymbol{\theta}^{(k)}) \\
&\vdots \\
&Y_i^{(b+1)} \text{ from } f(y_i \mid y_1^{(b+1)}, y_2^{(b+1)}, \ldots, y_{i-1}^{(b+1)}, y_{i+1}^{(b)}, \ldots, y_n^{(b)}, \mathbf{W}; \boldsymbol{\theta}^{(k)}) \\
&\vdots \\
&Y_n^{(b+1)} \text{ from } f(y_n \mid y_1^{(b+1)}, y_2^{(b+1)}, \ldots, y_{n-1}^{(b+1)}, \mathbf{W}; \boldsymbol{\theta}^{(k)}).
\end{aligned}$$

위에서 $f(\mathrm{y_i}|y_j, j \neq i; w_1, \ldots, w_n) = f(y_i|y_j, j \neq i; w_i)$ 은 Truncated Normal 분포를 따르기 때문에 이를 이용하여 y 를 sampling 할 수 있었고, 이를 통해 EM 알고리즘에 의해 MLE 를 추정할 수 있었다. 그러나 2-2 에서 설명한 sample size 를 결정하는 알고리즘을 구현하지 못했고, 따라서 적절한 sampling 을 하지 못함에 따라 결과 도출에는 실패하였다.

IV. Conclusion

MCEM 알고리즘은 log-likelihood의 기댓값을 계산하기 어려운 경우에 Monte Carlo simulation을 통해서 그 값을 추정하여 EM 알고리즘을 적용시킬 수 있다는 데에 의의가 있다. 또한 이 과정에서 target density로부터 직접적인 sampling이 어려울 경우 MCMC를 이용하여 sample을 얻을 수 있고, 이를 예제에 적용하여 구현해봄으로써 실제로 어떻게 알고리즘이 적용되는지 알 수 있었다. 특히 모든 계산 알고리즘에서 고려해야 하는 계산 비용에 대한 부분을 깊이 있게 고려해볼 수 있었다. 계산 시간을 줄이기 위해 모든 iteration에서 새롭게 sampling하지 않고 important sampling을 응용하였으며, 적절한 sample size를 선택하기 위해 MCMC에서 근사적으로 독립인 subsample을 얻어 오차를 추정하는 방법을 알 수 있었다.

Monte Carlo simulation에서 sample size가 부족하다면 추정치가 정확하지 않을 것이며, 과하다면 계산 비용의 면에서 비효율적이라고 할 수 있다. 실제로 예제 3-2에서는 각각의 EM 단계에서 필요한 적절한 sample size를 계산하지 못함에 따라 실제 MLE 값과 매우 먼 값이 추정되어 결과적으로 적절한 MLE의 추정에 실패하였다. 이를 통해 Monte Carlo simulation에서 sample size 결정의 중요성을 실감할 수 있었다. 그러나 오차의 근사치를 구하는 수식이 매우 복잡하여 결과를 도출하지 못한 점이 아쉬웠다.

References

1. Geof H. Givens, Jennifer A. Hoeting (2013) "Computational Statistics (2nd ed.)", Wiley

2. James G. Booth, James P. Hobert (1999) "Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm", Journal of the Royal Statistical Society. Series B (Statistical Methodology), Vol. 61, No. 1, pp. 265-285

3. McCullagh, P., Nelder, J. A. (1989) "Generalized Liner Models (2nd ed.)", Chapman and Hall

4. McCulloch, C. E. (1994) "Maximum Likelihood Variance Components Estimation for Binary Data", Journal of the American Statistical Association Vol. 89, pp. 330-335.

5. Richard A. Levine, George Casella (2001) "Implementations of the Monete Carlo EM Algorithm", Journal of Computational and Graphical Statistics, Vol. 10, No, 3, pp. 422–439

6. Richard A. Levine, Juanjuan Fan (2003) "An automated Markov chain Monte Carlo EM algorithm", Journal of Statistical Computation & Simulation, Vol. 74, No. 5, pp. 349–360