3장. 마코브 체인 몬테칼로(MCMC)의 수렴진단, 정확도, 효율

2장에서는 메트로폴리스, 메트로폴리스-헤스팅스, 깁스표본기법 등의 MCMC 알고리즘을 소개하였다. 실제로 MCMC 알고리즘을 사용하여 사후추론을 위해서는 몇 가지 고려할 사항들이 있다. 이 장에서는 MCMC 체인의 수렴진단, 정확도, 효율에 대하여 알아보기로 한다.

3.1 MCMC의 수렴진단

MCMC 기법은 표본이 마코브체인으로부터 생성되기 때문에 표본의 분포가 우리가 원하는 사후분포로 수렴하는 데까지는 시간이 걸린다. MCMC 체인에서 수렴에 이르기 전 단계를 준비단계(burn-in)라 부른다. 준비 단계에서 생성된 표본은 사후분포를 따르지 않으므로 버리고, 준비 단계 이후부터 생성된 표본을 가지고 추론을 수행해야 한다. MCMC 알고리즘 수행에서 스텝을 보통 반복(iteration)이라 부른다. 예를 들어 준비단계가 1000번째의 반복까지라고 하면, 1~1000번까지의 반복에서 생성된 표본은 버리고 1001~11000번째 반복에서 생성된 표본을 이용하여 모수의 사후기대치 사후구간 등을 추정한다. 그렇다면 질문은 'MCMC에서 초기의 몇 번째 반복까지를 준비단계로 볼 것인가?'이다. 이에 대한 답은 '경우에 따라 다르다' 이다. 어떤 경우에는 10번째에 수렴에 이를 수도 있지만 어떤 경우에는 50000번째에도 수렴이 이르지 못할 수도 있다. 통계적 모형, 자료, 그리고 알고리즘에 따라 수렴속도가 제각각 다르기 때문에 언제 MCMC 체인이 수렴에 도달하는지 일률적으로 말하는 것은 불가능하며, 주어진 문제마다 MCMC 수렴성 여부를 확인하고 준비단계 이후의 표본만 사용하여야 한다.

그렇다면 MCMC 체인이 몇 번 반복 후에 수렴하는지 어떻게 확인할 것인가? 물론 MCMC 체인을 무한히 반복하면 수렴에 이르지만 실제 우리가 수행하는 MCMC 체인의 반복수는 유한하기 때문에 몇 번째 반복 이후에 MCMC 체인이 충분히 수렴에 가깝다고 봐도 되는지, 즉, 표본이 사후표본이라고 간주할 수 있는지를 알아야 한다. 불행하게도 현재까지 이를 확인할 수 있는 완벽한 방법은 존재하지 않는다. 다시 말하면, 이러 이러한 조건들이 만족되면 몇 번째 반복 이후에는 MCMC 체인이 수렴하는 것을 보장한다고 하는 충분조건은 존재하지 않는다. 그러나 우리는 만약 수렴에 도달했다면 어떤 조건들이 만족되어야 하는지 필요조건들은 알고 있으므로 이 필요조건들을 확인하여 수렴의 여부를 "대략적으로" 확인하는 방법을 사용한다. 주의할 점은 필요조건들이 만족되었다고 해서 수렴성을 보장하는 것은 아니고 단지 수렴에 위배되는 증거들을 찾지 못함으로 수렴의 가능성이 높다고 간주하는 것이다.

현재 사용되고 있는 MCMC 수렴 진단방법들은 수렴의 필요조건으로 다음의 두 가지를 고려한다. 첫째, 마코브체인이 수렴에 이르면 초기치의 영향이 사라진다. 즉, 수렴에 이르렀다면 초기치로 어느 값을 사용하든 상관없이 표본의 분포가 안정적으로 사후분포에 도달해야 한다. 둘째, 마코브체인이 수렴에 이르면 체인이 사후분포의 전체 영역을 방문하며, 각 영역을 그 영역에서의 사후밀도함수 값에 비례하여 방문한다. 즉, 체인이 어느 특정 영역에

지나치게 오래 머무르지 않고 전체 영역을 자유롭게 이동하며, 사후밀도함수 값이 큰 영역은 자주 방문하여 상대적으로 많은 수의 표본이 생성되도록 하고 반면 사후밀도함수 값이 작은 영역은 뜸하게 방문하여 상대적으로 작은 수의 표본이 생성되도록 할 것이다.

위의 두 필요조건을 확인하기 위해 가장 많이 사용되는 방법은 표본의 시간적 궤도(trajectory)를 살펴보는 시각적 방법이다. 이차원 그래프의 가로축에는 반복수(iteration number)를 세로축에는 각 반복마다 생성된 표본의 값을 표시하는 경로그림을 그린다. 서로 다른 초기치에서 출발한 여러 개의 MCMC 체인에 대하여 경로그림을 겹쳐 그려보아 이들 경로그림이 어느 시점 이후에 초기치의 영향을 벗어나 서로 잘 겹쳐지는지 눈으로 확인하는 것이다.

(예 3.1) $X_i|\mu \sim N(\mu,1)$, $i=1,..,20$, 인 경우에 $\bar{x}=4$가 관측되었다. μ에 대한 사전분포로 $N(0,10^2)$를 가정하고, μ의 사후분포로부터 표본을 추출하기 위한 랜덤워크 메트로폴리스-헤스팅스 기법을 수행해 보자. 각 반복에서 새로운 표본의 후보를 $N(\mu^{curr},\delta^2)$에서 추출하기로 하고, $\delta=0.2$ 을 선택하였다.

그림 3.1 은 예 3.1 에서 3개의 MCMC 체인에 대한 경로그림이다. 그림 3.1 (a) 를 보면 이 3개의 체인은 서로 다른 초기치에서 출발했지만 약 100번 이후에는 서로 잘 겹쳐지며 증가 또는 감소의 경향(trend)을 보이지 않고 안정적인 모양을 보임을 알 수 있다. 따라서 수렴시점을 100번 이후로 정하면 될 것이다. 수렴시점을 100으로 보아도 무방할 것으로 보이나 조금 더 넉넉하게 500으로 생각해보자. 그림 3.1 (b) 는 위 3개의 MCMC 체인들에서 501~2500번째 반복까지의 표본에 대한 경로그림이다. 그림 3.1 (b) 를 보면 서로 잘 겹쳐져 있으며 증가 또는 감소의 경향이 보이지 않고 안정적인 모양을 보인다. 따라서 500번째 반복 이후부터 생성된 표본은 사후분포를 따르는 표본일 가능성이 높다. 다시 한 번 주의할 점은 위와 같이 잘 겹쳐져 있고 안정적인 모양을 보인다고 해서 수렴을 100% 보장하는 점은 아니라는 것이다. 예를 들어 사후밀도함수가 다봉함수(multimodal)인 경우 3개의 체인이 모두 사후밀도함수의 봉우리가 높은 최고 봉우리 구간에서 빠져나와 다른 봉우리 쪽으로 이동하지 못한 채 최고 봉우리 구간에서만 표본을 생성할 경우, MCMC 표본의 분포가 실제 사후밀도함수와 상당히 다름에도 불구하고 위와 같은 안정적인 경로그림을 보여줄 수 있다. 그러나 사후밀도함수가 단봉함수인 경우에는 3개의 체인이 모두 어느 특정 구간에서만 머물 가능성은 매우 낮다.

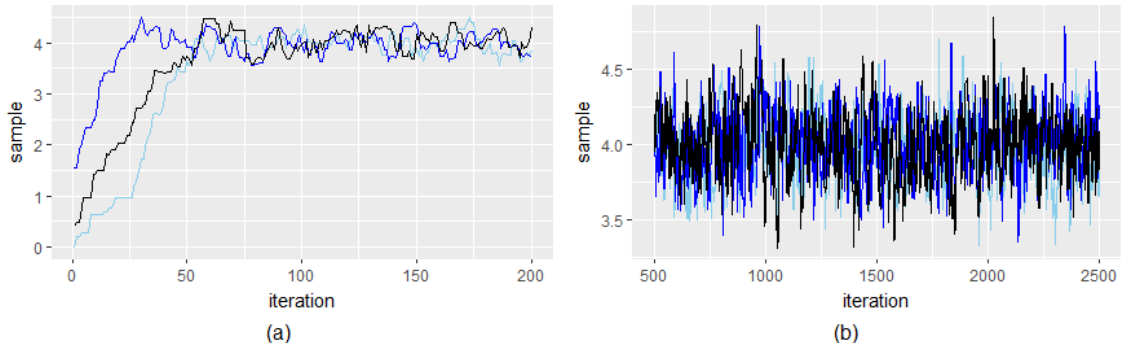


그림 3.1 : 예 3.1에서 3개의 체인에 대한 경로그림

서로 다른 체인에서 생성된 표본이 같은 분포로 수렴하는지를 확인하는 또 다른 방법은 서로 다른 MCMC체인의 표본으로부터 표본의 밀도함수를 그려보아 이들이 잘 일치하는지 살펴보는 방법이다. 그림 3.2 (a) 는 예 3.1에서 3개의 서로 다른 MCMC 체인으로부터 생성된 처음 200개의 표본으로 추정된 밀도함수를 겹쳐 그린 것이다. 3개의 밀도함수가 일치하지 않는 것으로 보아 이 표본들은 수렴에 이르기 전 단계의 표본이다. 그림 3.2 (b) 는 위의 3개 체인의 501~10000번째 반복에서 얻은 표본으로 추정된 밀도함수를 겹쳐 그린 것이다. 그림 3.2 (a) 와 달리 3개의 밀도함수가 비교적 잘 일치하므로 500번 이후에는 수렴에 이르렀을 가능성이 높다.

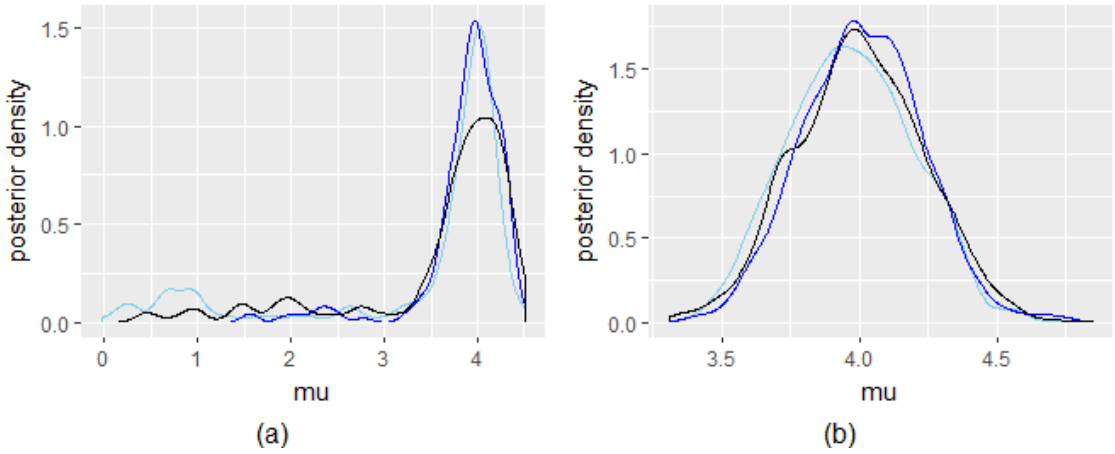


그림 3.2 : 예 3.1에서 3개의 체인으로부터 추정된 사후밀도함수
(a) 반복 1~200, (b) 반복 501~10000

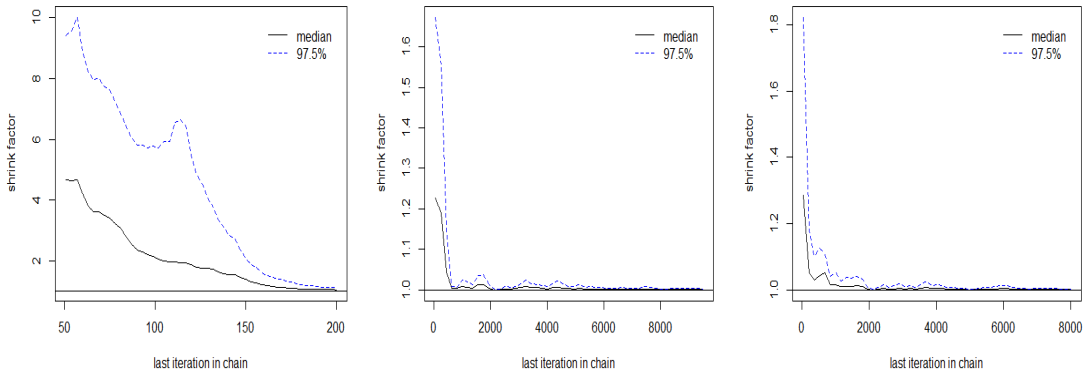
이상의 경로그림과 표본 밀도함수 그림은 시각적 진단 방법이다. 시각적 진단은 눈으로 보아 확인할 수 있는 좋은 방법이지만 추정하고자 하는 모수의 수가 매우 많을 경우 모든 모수에 대하여 그림을 그려보는 것은 현실적으로 어려울 수 있다. Gelman and Rubin (1993) 은 간단한 수치적 측도로 수렴성을 진단하는 방법을 제안하였는데 이 방법은 현재 널리 쓰이고 있으며 거의 모든 수렴 진단 패키지에서 이 수치를 제공한다.

동일한 모수에 대하여 초기치를 달리 하는 여러 개의 MCMC 체인을 이용하여 표본을 생성할 경우, 하나의 체인 내에 존재하는 표본의 분산 WV(within-chain variance) 와 서로 다른 체인 사이에 존재하는 표본의 분산 BV(between-chain variance)가 존재한다. 만약 수렴이 잘되어 서로 다른 체인이 잘 섞인다면 WV와 BV는 같은 값을 가질 것이고 두 분산의 비율은 1 값을 가질 것이다. 그러나 만약 어떤 체인이 나머지 체인들과 겹치지 않고 다

른 영역에서 표본을 생성하다면 BV는 WV보다 클 것이고, 잘 섞이지 않는 체인의 수가 많을수록 그리고 섞이지 않는 정도가 심할수록 BV와 WV의 비율이 커질 것이다. 간단히 Gelman 상수라고 불리는 Gelman-Rubin 통계량은 축소지수(shrink factor) 또는 잠재적 축소지수(potential scale reduction factor)라고 불리며 N을 MCMC 체인에서 반복수, M을 체인의 수라 할 때 다음 식으로 주어진다.

$$\text{Gelman 상수} = \frac{N-1}{N} + \frac{M+1}{MN} \cdot \frac{B}{W}$$

위 식에서 만약 MCMC 체인이 수렴에 이르렀다면 Gelman-Rubin 통계량 값은 1에 가까운 값을 가질 것이므로, Gelman 상수가 1에 가까우면 수렴 가능성이 높다고 본다. 통상적으로 Gelman 상수가 1.1 이하이면 수렴에 이르렀다고 본다. Gelman 상수에 대한 95% 신뢰구간을 구할 수 있으므로 신뢰구간의 상한가가 1.1을 넘지 않는지 확인하여 수렴을 진단한다.



(a) 반복 1~200 (b) 반복 501~10000 (c) 반복 2001~10000

그림 3.3 : 예 3.1에서 반복수에 따른 Gelman 상수

그림 3.3 (a) 는 3개의 체인에서 처음 200개의 반복에 대하여 계산한 Gelman 상수이다. 이 값들이 1보다 상당히 크므로 아직 수렴에 이르지 않은 것으로 판단한다. 그림 3.3 (b)와 (c)는 각각 3개의 체인에서 501~10000과 2001~10000 번째 반복으로부터의 표본을 가지고 계산한 Gelman 상수이다. 그림 3.3 (b)는 Gelman 상수가 1보다 크다. 반면 그림 3.3 (c)는 95% 신뢰구간의 상한값이 1.1보다 약간 크긴 하지만 중앙값이 1.05에 가까우므로 수렴 가능성이 높다고 본다.

Gelman 상수 이외에 수렴성을 진단하는 통계량으로는 Brooks and Gelman(1998), Geweke(1992), Raftery-Lewis(1992), Heidelberg and Welch(1983) 등이 있다. R 패키지 boa(Smith, 2007)는 이들 통계량 값들을 제공하여 MCMC 표본의 수렴진단과 사후추론에 유용하게 사용될 수 있다.

예 3.1에서 이상의 세 가지 진단 방법을 모두 고려할 경우, MCMC 체인에서 2000번 이후의 표본을 추출하여 사후추론에 사용하면 적절하겠다.

위에서 제시한 진단방법을 적용하는 프로그램은 다음과 같다. Gelman 상수 계산을 위해 coda 패키지의 gelman.plot 함수를 사용하였다. 참고로, coda 패키지를 사용하기 위해서는 각 체인으로부터 추출된 표본이 mcmc.list 형식으로 저장되어 있어야 한다. mcmc.list 형식으로 표본을 저장하는 방법은 프로그램을 참고하라.

```
install.packages("coda")
library(coda)

##### Read Data #####
n=20; xbar=4; sigma=1
mu0=0 ; sigma0=10
dataList=list(n=n,xbar=xbar,sigma=sigma, mu0=mu0, sigma0=sigma0)

#### Function To Compute Posterior Kernel ####
post.kernel=function(mu,dataList){
post.kernel= exp( -0.5*( ((dataList$xbar-mu)*sqrt(dataList$n)/dataList$sigma )^2
                         + ((mu-dataList$mu0)/dataList$sigma0 )^2 ))
return(post.kernel)
}

##### Function To Perform Random Walk Metropolis ####
Metro=function(nsim,mu.init, delta, dataList){

mu.samples=mu.init
mu.curr=mu.init

for (iter in 1:nsim){
  mu.prop = rnorm(1, mean=mu.curr, sd=delta)
  alpha=min (1, post.kernel(mu.prop,dataList)
                /post.kernel(mu.curr,dataList) )
  u=runif(1)
  mu.next= mu.prop*(u<alpha)+mu.curr*(u>alpha)
  mu.samples=rbind(mu.samples, mu.next)
  mu.curr=mu.next
}

return(mu.samples)
}

#################################################

delta=0.2
```

```
nsim=10000
n.chains=3

mu.Samples=matrix(0, nsim, n.chains)

for( i in 1:n.chains){
mu.init = rnorm(1,mu0,2)
mu.Samples[, i]= Metro(nsim-1,mu.init,delta, dataList)
}

nwarm=500; nsim=10000

#### Fig3.1 예 3.1에서 3개의 체인에 대한 경로그림 ####
plot(mu.Samples[(nwarm+1):nsim,1], xlab="iteration", ylab="sample",type="l",main="", sub="(b)")
lines(mu.Samples[(nwarm+1):nsim,2], col=2)
lines(mu.Samples[(nwarm+1):nsim,3], col=3)

savePlot(file="figure/Fig_3_1_traceplot", type=c("bmp"), device=dev.cur())

#### Fig3.2 예 3.1에서 3개의 체인으로부터 추정된 사후밀도함수 ####
par(mfrow=c(1,2))
nwarm=0; nsim=200
plot(density(mu.Samples[(nwarm+1):nsim,1]), xlab="mu", ylab="posterior density", main="",
sub="(a)")
lines(density(mu.Samples[(nwarm+1):nsim,2]),col=2)
lines(density(mu.Samples[(nwarm+1):nsim,3]),col=3)

nwarm=500; nsim=10000
plot(density(mu.Samples[(nwarm+1):nsim,1]), xlab="mu", ylab="posterior density", main="",
sub="(b)")
lines(density(mu.Samples[(nwarm+1):nsim,2]),col=2)
lines(density(mu.Samples[(nwarm+1):nsim,3]),col=3)
savePlot(file="figure/Fig_3_2_postdensity", type=c("bmp"), device=dev.cur())

#### Fig3.3 예 3.1에서 반복수에 따른 Gelman 상수 ####
mu1.Samples=mcmc(mu.Samples[1:200,])
mu1.codaSamples=mcmc.list(list(mu1.Samples[,1],mu1.Samples[,2],mu1.Samples[,3]))
gelman.plot(mu1.codaSamples, col=c("black", "blue"))
savePlot(file="figure/Fig_3_3_a_Gelman", type=c("bmp"), device=dev.cur())

mu2.Samples=mcmc(mu.Samples[501:10000,])
mu2.codaSamples=mcmc.list(list(mu2.Samples[,1],mu2.Samples[,2],mu2.Samples[,3]))
gelman.plot(mu2.codaSamples, col=c("black", "blue"))

mu3.Samples=mcmc(mu.Samples[2001:10000,])
```

```
mu3.codaSamples=mcmc.list(list(mu3.Samples[,1],mu3.Samples[,2],mu3.Samples[,3]))
gelman.plot(mu3.codaSamples, col=c("black", "blue"))
savePlot(file="figure/Fig_3_3_b_Gelman", type=c("bmp"), device=dev.cur()) ■
```

3.2 MCMC 추정치의 오차 추정

MCMC 체인에서 준비단계 이후에 충분한 수의 반복으로부터 생성된 모수의 표본으로부터 모수의 사후추정치를 구할 수 있다. 모수의 표본 분포가 대칭에서 크게 벗어나지 않으면 보통 표본 평균을 사후추정치로 사용한다. 이렇게 얻어진 사후 추정치는 표본으로부터 얻어진 것이기 때문에 변동성을 가진다. 즉, MCMC 체인은 랜덤으로 표본을 생성하기 때문에 수행할 때 마다 표본이 달라지므로 모수의 사후 추정치 값이 달라질 수 있다. 사후 추정치 값이 MCMC 에 따라 얼마나 달라질 수 있는지의 불확실성은 사후 추정치의 MCMC 표준오차(MCSE)로 측정할 수 있다.

MCSE는 사용되는 MCMC 표본의 수에 따라 달라지며, 물론 표본의 수가 많아지면 작아진다. 독립적인 표본의 경우 평균의 표준오차는 표본 수의 제곱근에 반비례한다고 알려져 있다. 하지만 MCMC 표본은 마코브체인으로부터 생성되었기 때문에 독립적인 표본이 아니며 서로 자기상관(autocorrelation)을 갖는다. 그림 3.4는 예 3.1의 MCMC 표본에 대한 자기상관을 표시한 ACF(Auto Correlation Function) 그림이다. 가로축은 표본끼리의 간격(lag)을 나타내고 세로축은 주어진 간격을 갖는 표본들의 상관계수를 나타낸다. 그림을 보면, 바로 이웃한 (lag=1) 표본끼리의 상관계수는 0.9로 매우 높고 간격이 벌어질수록 상관계수 값이 작아짐을 볼 수 있다. 높은 자기상관을 갖는다는 것은 다음 시점(next iteration)의 표본이 현 시점(current iteration)의 표본과 밀접한 관련이 있으므로 정보의 중복이 심하여 새로이 추가되는 정보가 미미함을 의미한다. 따라서 자기상관이 높은 10개의 표본은 독립적인 10개의 표본에 비하여 적은 양의 정보를 제공한다.

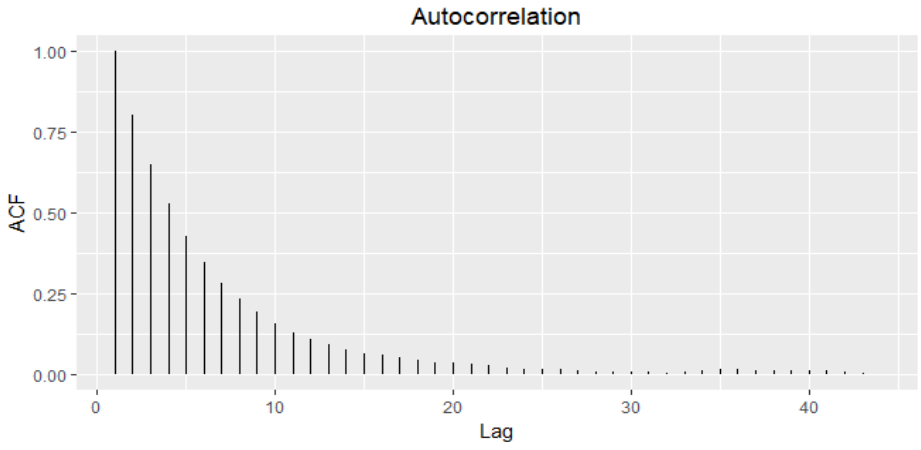


그림 3.4 : 예 3.1 표본의 자기상관 그림

그렇다면, MCMC 표본의 자기상관으로 인하여 축소된 정보의 양은 어느 정도일까? 달리 말하면 자기상관을 갖는 MCMC 표본이 N 개 있다고 하면, 이는 정보의 양을 기준으로 할 때 몇 개의 독립적 표본과 맞먹는다고 볼 수 있는가? 자기상관을 갖는 N 개의 표본과 맞먹는 독립적인 표본의 수를 효용표본수(ESS, Effective sample size)라 한다. Radford(2001, p.99) 가 제안한 효용표본수는

$$ESS = \frac{N}{1+2\sum_{k=1}^{\infty} ACF(k)}$$

로 주어진다. $ACF(k)$는 간격 k를 갖는 표본들의 상관계수이다. 현실적으로는 무한대의 간격을 갖는 표본들의 상관계수를 구할 수 없으므로 통상적으로 $ACF(k) < 0.05$ 가 되는 k 까지만 계산하고 그 이후로는 0 값으로 취급하여 무시한다.

예 3.1에서 ESS를 구해보자. acf 함수를 사용하여 자기상관을 구해보면 lag=17 이후로는 자기상관이 0.05 이하가 됨을 볼 수 있다. lag=1부터 lag=17까지의 자기상관을 이용하여 하나의 초기치로부터 시작하여 2000번 이후에 생성된 8000개의 MCMC 표본에 대하여 ESS를 구하면 ESS=208 이 됨을 알 수 있다. 실제 MCMC 표본의 개수는 8000개이지만 높은 자기상관으로 인하여 효용표본수는 훨씬 작은 208개이다.

```
#### ACF값 ####
mu.postSamples=as.matrix(mu3.codaSamples)
aa=acf(mu.postSamples); aa
```

```
Autocorrelations of series 'mu.postSamples', by lag

     0      1      2      3      4      5      6      7      8      9     10     11     12
 1.000  0.799  0.643  0.518  0.419  0.340  0.273  0.219  0.175  0.138  0.113  0.093  0.078
    13     14     15     16     17     18     19     20     21     22     23     24     25
 0.066  0.054  0.045  0.042  0.038  0.035  0.032  0.031  0.029  0.026  0.023  0.019  0.017
    26     27     28     29     30     31     32     33     34     35     36     37     38
 0.016  0.016  0.018  0.017  0.011  0.003 -0.004 -0.012 -0.013 -0.012 -0.014 -0.019 -0.022
    39     40     41     42     43
-0.028 -0.029 -0.028 -0.030 -0.027
```

```
#### Fig 3.4 예 3.1 표본의 자기상관 그림 ####
par(mfrow=c(1,1))
plot( acf(mu.Samples[(nwarm+1):nsim,1]) , main="Autocorrelation")
savePlot(file="figure/Fig_3_4_acf", type=c("bmp"), device=dev.cur())

mu.postSamples=as.matrix(mu3.codaSamples)
aa=as.vector(unlist(acf(mu.postSamples)))
aa=as.numeric(aa[2:17])
EES=(nsim-nwarm)/( 1+ 2*sum( aa) ); EES ■
```

효용표본수가 얼마나 커야 신뢰할만한 사후추정치를 구할 수 있을까? 적절한 효용표본 수는 추정하고자 하는 모수에 따라 다르다. 표본이 밀집한 사후분포의 중앙 부분, 예를 들면 사후기대치나 사후중앙값을 추정하고자 할 때는 효용표본수가 작아도 괜찮지만 표본이 희박

한 부분, 예를 들면 95% 사후구간 추정에 필요한 2.5% 사후백분위나 97.5% 사후백분위 등을 추정하고자 할 때는 효용표본수가 커야 신뢰할 만한 추정치를 얻을 수 있다. 통상적으로 사후기대치나 사후중앙값을 추정할 때 효용표본 수는 몇 백 개 정도로 충분하지만 사후구간 추정을 위해서는 몇 만 개 이상의 표본이 사용해야 어느 정도 신뢰할 만한 결과를 얻을 수 있다.

모수 θ에 대한 N개의 MCMC 표본으로부터 평균을 구하여 θ의 사후추정치 $\hat{\theta}$를 구하였다면, $\hat{\theta}$의 변동성을 측정하는 표준오차는

$$MCSE(\hat{\theta}) = \frac{SD}{\sqrt{ESS}}$$

이다. 여기에서 SD는 θ에 대한 N개의 MCMC 표본으로부터 구한 표준편차이다. SD 가 같다고 할 때 자기상관이 큰 표본일수록 실제 표본크기 N 에 비하여 ESS 는 작아지므로, 동일한 정확도의 사후추정치를 얻기 위해서는 자기 상관이 큰 MCMC 체인이 자기 상관이 작은 MCMC 체인에 비하여 더 많은 수의 반복수가 필요하다. 따라서 다른 조건이 모두 같다면 자기상관이 작은 표본을 생성하는 MCMC 알고리즘을 선택하는 것이 더 효율적이다.

3.3 효율적인 MCMC

MCMC는 대체로 모수의 사후표본 추출이 어려운 복잡한 모형에서 다변량 모수의 표본추출에 사용된다. 적절한 조건에 맞는 마코브 체인에서 표본을 추출할 경우 반복수를 충분히 늘리면 원하는 사후분포로 표본의 분포가 수렴하고, 수렴 후 충분히 큰 수의 표본을 사용하여 추론을 하면 원하는 정확도의 추정치를 얻을 수 있다. 그러나 현실적으로는 가능한 짧은 시간 내에 적절한 정확도의 추정치를 얻는 것이 바람직할 것이다. 이를 위해서는 MCMC의 효율성(efficiency)을 높이는 방법이 무엇인지 알아보는 것이 중요하다.

2장에서 소개된 깁스표본 기법은 다변량 모수의 각 원소모수의 조건부 사후분포로부터 표본을 생성하는 기법이다. 그런데, 보다 정확히 기술하자면 깁스표본기법은 다변량 모수를 블록 또는 그룹으로 나누어 각 블럭의 조건부 분포로부터 표본을 생성하는 방법이다. 이때 블럭은 하나의 일변량 모수 또는 다변량 모수가 될 수 있다. 예를 들면, $\theta=(\theta_1,...,\theta_4)$이 4차원 모수일 때 $\theta=((\theta_1,\theta_2),\theta_3,\theta_4)$와 같이 3개의 블록으로 나눈 다음 조건에 들어가는 모수의 값을 가장 최근의 값으로 대체하면서 (θ_1,θ_2)의 이변량 표본을 $\pi(\theta_1,\theta_2|\theta_3,\theta_4,x)$로부터 추출하고, θ_3의 표본을 $\pi(\theta_3|\theta_1,\theta_2,\theta_4,x)$ 로부터 추출하고, θ_4의 표본을 $\pi(\theta_4|\theta_1,\theta_2,\theta_3,x)$ 로부터 추출하는 것이다. 물론 $\pi(\theta_1,\theta_2|\theta_3,\theta_4,x)$로부터의 표본 추출이 용이해야 위의 방법을 적용할 수 있다. 이처럼 깁스표본 기법에서 블록 모수 (θ_1,θ_2)의 표본을 한 번에 추출하면, θ_1의 표본을 $\pi(\theta_1|\theta_2,\theta_3,\theta_4,x)$로부터 그리고 θ_2의 표본을 $\pi(\theta_2|\theta_1,\theta_3,\theta_4,x)$로부터 추출하는 2단계 방법에 비하여 수렴이 빨라지게 되어 효율이 좋아진다. 이를 쉽게 설명하면, 이차원 공간을 이동하는데 처음에 가로 방향으로 이동하고 다음에 세로 방향으로 이동하는 것 보다는 한 번에 대각선으로 이동할 수 있다면 더 빨리 이동할 수 있는 원리와 같다.

또한, 예를 들어 θ_3의 표본을 추출하는 단계의 조건에서 θ_1이 빠진 $\pi(\theta_3|\theta_2,\theta_4,x)$에서의 표본 추출이 용이하다면 θ_3의 '완전' 조건부 사후분포 $\pi(\theta_3|\theta_1,\theta_2,\theta_4,x)$ 대신 부분 조건부 사후분포 $\pi(\theta_3|\theta_2,\theta_4,x)$로부터 θ_3의 표본을 추출하는 것이 효율 면에서 더 좋고, 나아가 만약 주변사후분포 $\pi(\theta_3|x)$에서의 표본 추출이 용이하다면 $\pi(\theta_3|x)$로부터 θ_3의 표본을 추출하는 것이 효율 면에서 앞의 두 경우 보다 더 나은 방법이다. 쉽게 설명하자면 조건으로 들어가는 모수의 수가 작을수록 θ_3의 표본 추출에 제약이 덜해지므로 θ_3의 표본이 우리가 원하는 θ_3의 사후분포 영역을 빨리 커버할 수 있으므로 효율이 좋아진다.

메트로폴리스-헤스팅스 기법에서는 표본추출함수의 선택이 효율에 영향을 미친다. 랜덤워크 메트로폴리스 기법의 경우를 보면, 후보 표본을 현재 표본값을 평균으로 하는 (다변량) 정규분포로부터 추출한다. 즉, $\theta^{prop} \sim N_p(\theta^{curr},\delta^2 I_p)$인데 I_p는 대각값이 모두 1인 p차원 정방대각행렬이다. 이 때 대략적으로 표준편차의 3배인 3δ는 MCMC 체인이 이동시 사용하는 보폭의 허용치로 간주할 수 있다. δ가 지나치게 작으면 현재의 표본값에서 매우 가까운 지점에서 후보표본 θ^{prop}가 추출된다. 이 후보표본과 현재 표본의 사후밀도함수 값은 유사한 값을 가지므로 후보표본을 다음 단계의 표본으로 채택하는 채택확률은 높지만, 두 이웃한 표본들 간의 자기상관성은 클 것이고 따라서 체인의 수렴이 늦어지는 단점이 있다. 반면 δ가 지나치게 크면 체인의 허용 보폭이 크므로 후보 표본 θ^{prop}가 현재 표본에서 멀리 이동할 수 있어 사후분포의 다른 구간으로 쉽게 이동할 수 있는 장점이 있다. 그러나 사후밀도함수 값이 큰 구간에서 사후밀도함수 값이 작은 구간으로 이동할 가능성도 높아져서, 후보 표본을 다음 단계의 표본으로 채택하는 채택확률이 낮아짐으로 결과적으로 체인이 현재 표본에서 이동하지 않고 머무를 가능성이 높게 되어 수렴이 늦어진다.

따라서 너무 작거나 크지 않은 적절한 δ를 선택하는 것이 이상적이다. 이를 채택확률 기준으로 표현하면, 채택확률이 너무 크거나 작지 않게 δ를 선택하는 것이 랜덤워크 메트로폴리스 기법에서 효율을 높이는 방법이다. Robert, Gelman, Gilks(1997)은 p 차원 모수가 독립적으로 동일한 정규분포를 따르고 이들의 사후표준편차를 σ^π라 할 때

$$\delta = 2.4\ \sigma^\pi/\sqrt{p}$$

로 선택하면 점근적으로 최적의 수렴성을 가지며 이에 대응하는 채택확률은 약 0.234 임을 보여 주었다. 이후로 많은 실제 예에서 채택확률이 약 0.234 가 되도록 랜덤워크 메트로폴리스 표본추출분포의 표준편차를 선택하는 것이 MCMC 효율을 좋게 하는 것으로 관측되었으며 Robert and Rosenthal(2001)은 0.234 채택확률을 산출하는 δ값이 앞서 Robert, Gelman, Gilks(1997)이 가정한 조건을 완화한 경우에도 점근적으로 최적 효율을 가져옴을 보였다. 따라서 랜덤워크 메트로폴리스 기법의 적용 시 표본추출분포인 정규분포의 표준편차를 여러 값으로 바꾸어가면서 대략적으로 0.234 채택확률을 주는 값을 선택하는 것이 효율을 높이는 좋은 방법이다.

깁스표본 기법과 메트로폴리스-헤스팅스(랜덤워크 메트로 폴리스 포함) 기법 중 어느 기법이 더 효율적인지 단정할 수는 없다. 깁스 표본기법은 채택확률을 계산하지 않기 때문에

계산시간을 줄이는 효과가 있고 표본추출 분포를 선택하지 않기 때문에 초기의 조정과정(tuning) 없이 자동적으로 MCMC를 수행할 수 있어 많은 경우 메트로폴리스-헤스팅스 보다 효율적이라고 알려져 있다. 그러나 원소모수 간에 상관관계가 클 경우 체인의 보폭이 작아져 수렴속도가 매우 느려질 수 있는데 이 경우는 오히려 랜덤워크 메트로폴리스의 효율이 더 좋을 수도 있다.

모형의 모수를 어떻게 정하느냐에 따라서도 MCMC 효율이 달라질 수 있다. 일반적인 룰은 모수들간의 상관관계가 작아지도록 모수를 변환하는 것이 MCMC 효율을 높이는데 도움이 된다. 예를 들면 Gelfand et al.(1995)는 정규선형 합성모형에서 MCMC에 효율적인 모수 변환을 제시하였다. 이상의 알고리즘이나 모수변환 외에 컴퓨터상에서 MCMC 수행시간을 줄이는 방법으로 여러 개의 MCMC 체인들을 다수의 컴퓨터 프로세서를 사용하여 병렬적으로 실행한(parallel processing) 후 표본을 수합하는 방법이 있다.

BJ Smith, 2007. boa: An R Package for MCMC Output Convergence Assessment and Posterior Inference, *Journal of Statistical Software,* 21, 1-37.
Brooks S, Gelman A, 1998. General Methods for Monitoring Convergence of Iterative Simulations, *Journal of Computational and Graphical Statistics*, 7(4), 434-455.
Gelman, A. and Rubin, D.B. 1992. Inference from Iterative Simulation Using Multiple Sequences(with discussion), *Statistical Science*, 7, 457-511.
Geweke J, 1992. Bayesian Statistics, volume 4, chapter Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments. *Oxford University Press*, New York.
Heidelberger P, Welch P, 1983. Simulation Run Length Control in the Presence of an Initial Transient, Operations Research, 31, 1109{1144.
Raftery AL, Lewis S, 1992. Bayesian Statistics, volume 4, chapter How Many Iterations in the Gibbs Sampler? *Oxford University Press*, New York.
Roberts, G.O. and Rosenthal, J.S. 2001. Optimal scaling for various Metropolis-Hastings algorithms, *Statist. Sci.* 16 351-367. MR1888450
Roberts, G.O., Gelman, A. and Gilks, W.R. 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Appl. Probab., 7, 110-120.

Radford M. N., 2001. Annealed importance sampling, *Statistics and Computing*, 11, 125-139

Gelfand A. E. et al., 1995, Efficient Parametrisations for Normal Linear Mixed Models, *Biometrika*, 82, 479-488