ESC 22 FALL / WEEK 9

Metropolis-Hastings

(Chapter 10. Nonconjugate priors and Metropolis-Hastings algorithms)

김송희 박태주

Contents

Metropolis-Hastings algorithm

- Posterior 구하려면?
  - *conjugate* or *semiconjugate* prior distribution:
    ⇒ **Monte Carlo method** or the **Gibbs sampler**
  - *conjugate* prior distribution is unavailable or undesirable:
    - if the *full conditional distributions* of the parameters do not have a standard form, the Gibbs sampler cannot be easily used.
    ⇒ **Metropolis-Hastings algorithm**
- **Metropolis-Hastings algorithm**: generic method of approximating the posterior distribution corresponding to any combination of prior distribution and sampling model.
- ex) Poisson regression(GLM), longitudinal regression(correlated observations over time)

Example: Song sparrow reproductive success

- Sample of 52 female song sparrows
- Age, number of new offspring was recorded for each sparrow

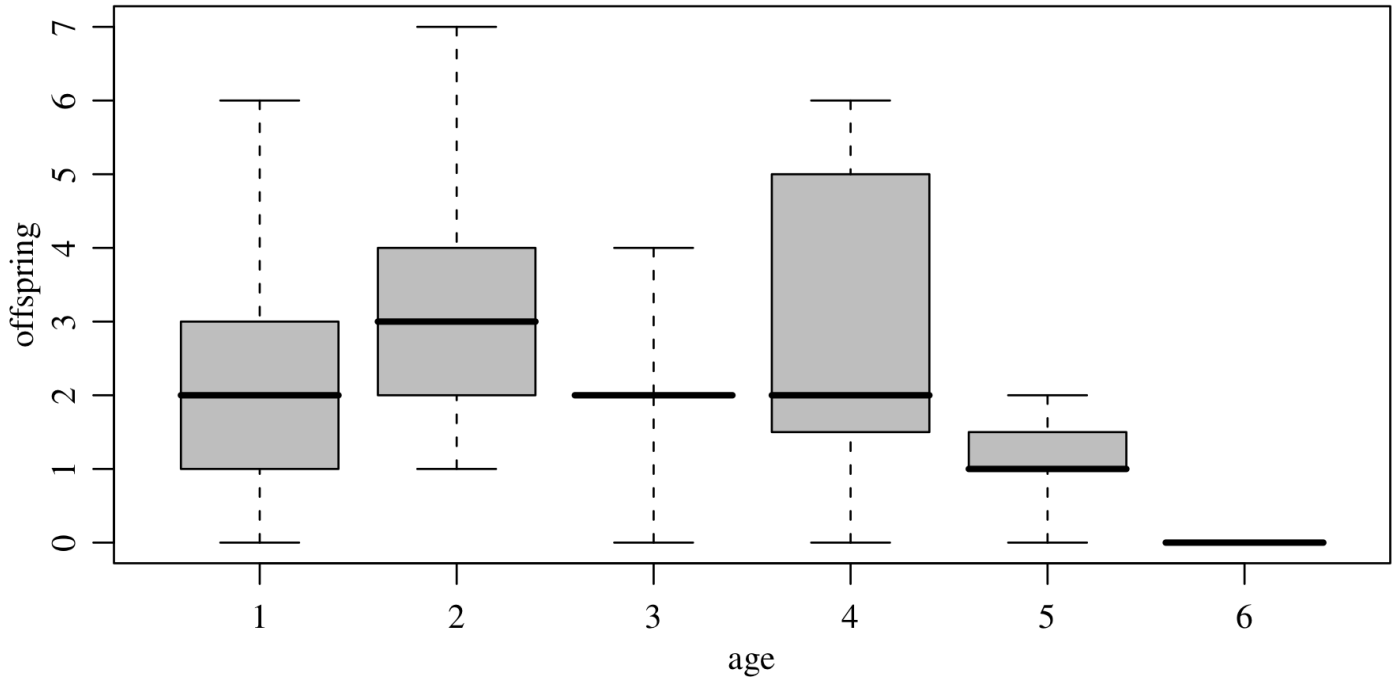


Fig. 10.1. Number of offspring versus age.

✓ Two-year-old birds had the highest median

✓ The number of offspring declines beyond two years of age

➢ This is not surprising from a biological point of view: One-year-old birds are in their first mating season and are relatively inexperienced compared to two-year-old birds. As birds age beyond two years they experience a general decline in health and activity.

Example: Song sparrow reproductive success

- Lets fit a probability model!
  ⇒ relationship between age and reproductive success, population forecasts, …
- # of offspring for each bird is a non-negative integer {0, 1, 2, …}
  ⇒ **Poisson model**

$$Y = \text{number of offspring conditional on } x = \text{age}$$

simple model $\{Y|x\} \sim \text{Poisson}(\theta_x)$

$\theta_x = \beta_1 + \beta_2 x + \beta_3 x^2$ (X)

log link $\log \text{E}[Y|x] = \log \theta_x = \beta_1 + \beta_2 x + \beta_3 x^2$

$\text{E}[Y|x] = \exp(\beta_1 + \beta_2 x + \beta_3 x^2)$

poisson regression model $\{Y|\boldsymbol{x}\} \sim \text{Poisson}(\exp[\boldsymbol{\beta}^T \boldsymbol{x}])$

⊃ *GLM*

10.1 Generalized linear models

Calculating posterior p(β|X, y) for GLM

As in the case of ordinary regression, a natural class of prior distributions for β is the class of multivariate normal distributions. However, for neither the Poisson nor the logistic regression model would a prior distribution from this class result in a multivariate normal posterior distribution for β. Furthermore, standard conjugate prior distributions for generalized linear models do not exist (except for the normal regression model).

- 일반적인 β에 대한 prior 형태: multivariate normal

- 하지만 poisson regression 의 경우 posterior가 multivariate normal 형태로 나오지 않는다.

- 그럼 prior가 다른 형태라면?: GLM의 경우, conjugate prior 존재하지 않음

- How to calculate the posterior distribution:
  1. **grid-based approximation** (Section 6.2) : p(y|X, β) × p(β) on a three-dimensional grid of β-values -> too big, inefficient
  2. approximation based on **Monte Carlo samples**: Although *independent* Monte Carlo sampling from the posterior is not available for this Poisson regression model, the next section will show how to construct a Markov chain that can approximate p(β|X, y) for any prior distribution p(β).

Metropolis algorithm: Intuition

- 왜 하는지?

Let's consider a very generic situation where we have a sampling model $Y \sim p(y|\theta)$ and a prior distribution $p(\theta)$. Although in most problems $p(y|\theta)$ and $p(\theta)$ can be calculated for any values of y and θ, $p(\theta|y) = p(\theta)p(y|\theta)/\int p(\theta')p(y|\theta')\ d\theta'$ is often hard to calculate due to the integral in the denominator. If we were able to sample from $p(\theta|y)$, then we could generate $\theta^{(1)}, \ldots, \theta^{(S)} \sim$ i.i.d. $p(\theta|y)$ and obtain Monte Carlo approximations to posterior quantities, such as

적분이 어려운 경우 Monte Carlo sampling 통해 근사 posterior 분포 구할 수 있음

- 만약 iid가 아니라서 Monte Carlo sampling 불가능하다면?

But what if we cannot sample directly from $p(\theta|y)$? In terms of approximating the posterior distribution, the critical thing is not that we have i.i.d. samples from $p(\theta|y)$ but rather that we are able to construct a large collection of θ-values, $\{\theta^{(1)}, \ldots, \theta^{(S)}\}$, whose empirical distribution approximates $p(\theta|y)$. Roughly speaking, for any two different values θ_a and θ_b we need

$$\frac{\#\{\theta^{(s)}\text{'s in the collection} = \theta_a\}}{\#\{\theta^{(s)}\text{'s in the collection} = \theta_b\}} \approx \frac{p(\theta_a|y)}{p(\theta_b|y)}.$$

Posterior 근사하는데 사실 iid이냐 아니냐는 크게 중요하지 않음.

그냥 충분히 많은 theta값들을 샘플링해서 emprical하게 근사 분포 구하면 됨!

Metropolis algorithm: Intuition

- New value θ^*를 working collection $\{\theta^{(1)}, \ldots, \theta^{(s)}\}$에 추가할지 안할지?

Let's think intuitively about how we might construct such a collection. Suppose we have a working collection $\{\theta^{(1)}, \ldots, \theta^{(s)}\}$ to which we would like to add a new value $\theta^{(s+1)}$. Let's consider adding a value θ^* which is nearby $\theta^{(s)}$. Should we include θ^* in the set or not? If $p(\theta^*|y) > p(\theta^{(s)}|y)$ then we want more θ^*'s in the set than $\theta^{(s)}$'s. Since $\theta^{(s)}$ is already in the set, then it seems we should include θ^* as well. On the other hand, if $p(\theta^*|y) < p(\theta^{(s)}|y)$ then it seems we should not necessarily include θ^*. So perhaps our decision to include θ^* or not should be based on a comparison of $p(\theta^*|y)$ to $p(\theta^{(s)}|y)$. Fortunately, this comparison can be made even if we cannot compute $p(\theta|y)$:

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y)} \frac{p(y)}{p(y|\theta^{(s)})p(\theta^{(s)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}. \qquad (10.1)$$

Metropolis algorithm

Having computed r, how should we proceed?

If $r > 1$:

Intuition: Since $\theta^{(s)}$ is already in our set, we should include θ^* as it has a higher probability than $\theta^{(s)}$.

Procedure: Accept θ^* into our set, i.e. set $\theta^{(s+1)} = \theta^*$.

If $r < 1$:

Intuition: The relative frequency of θ-values in our set equal to θ^* compared to those equal to $\theta^{(s)}$ should be $p(\theta^*|y)/p(\theta^{(s)}|y) = r$. This means that for every instance of $\theta^{(s)}$, we should have only a "fraction" of an instance of a θ^* value.

Procedure: Set $\theta^{(s+1)}$ equal to either θ^* or $\theta^{(s)}$, with probability r and $1 - r$ respectively.

Metropolis algorithm

This is the basic intuition behind the famous *Metropolis algorithm*. The Metropolis algorithm proceeds by sampling a proposal value θ^* nearby the current value $\theta^{(s)}$ using a *symmetric proposal distribution* $J(\theta^*|\theta^{(s)})$. Symmetric here means that $J(\theta_b|\theta_a) = J(\theta_a|\theta_b)$, i.e. the probability of proposing $\theta^* = \theta_b$ given that $\theta^{(s)} = \theta_a$ is equal to the probability of proposing $\theta^* = \theta_a$ given that $\theta^{(s)} = \theta_b$. Usually $J(\theta^*|\theta^{(s)})$ is very simple, with samples from $J(\theta^*|\theta^{(s)})$ being near $\theta^{(s)}$ with high probability. Examples include

- $J(\theta^*|\theta^{(s)}) = \text{uniform}(\theta^{(s)} - \delta, \theta^{(s)} + \delta)$;
- $J(\theta^*|\theta^{(s)}) = \text{normal}(\theta^{(s)}, \delta^2)$.

The value of the parameter δ is generally chosen to make the approximation algorithm run efficiently, as will be discussed in more detail shortly.

Metropolis algorithm: Procedure

1. Sample $\theta^* \sim J(\theta|\theta^{(s)})$; proposal distribution (symmetric)
2. Compute the acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})} .$$

3. Let

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{(s)} & \text{with probability } 1 - \min(r, 1). \end{cases}$$

Step 3 can be accomplished by sampling $u \sim \text{uniform}(0, 1)$ and setting $\theta^{(s+1)} = \theta^*$ if $u < r$ and setting $\theta^{(s+1)} = \theta^{(s)}$ otherwise.

Example: Normal distribution with known variance

- Conjugate normal, known variance

$$\theta \sim \text{normal}(\mu, \tau^2) \quad \text{prior}$$

$$\{y_1, \ldots, y_n|\theta\} \sim \text{i.i.d. normal}(\theta, \sigma^2) \quad \text{likelihood}$$

$$\{\theta|y_1, \ldots, y_n\} \sim \text{normal}(\mu_n, \tau_n^2) \quad \text{posterior}$$

$$\begin{aligned} \mu_n &= \bar{y}\frac{n/\sigma^2}{n/\sigma^2 + 1/\tau^2} + \mu\frac{1/\tau^2}{n/\sigma^2 + 1/\tau^2} \\ \tau_n^2 &= 1/(n/\sigma^2 + 1/\tau^2). \end{aligned}$$

- True posterior 계산

Suppose $\sigma^2 = 1, \tau^2 = 10, \mu = 5, n = 5$ and $\boldsymbol{y} = (9.37, 10.18, 9.16, 11.60, 10.33)$.
For these data, $\mu_n = 10.03$ and $\tau_n^2 = .20$, and so $p(\theta|\boldsymbol{y}) = \text{dnorm}(10.03, .44)$.

$$r = \frac{p(\theta^*|y)}{p(\theta^{(s)}|y)} = \frac{p(y|\theta^*)p(\theta^*)}{p(y)} \frac{p(y)}{p(y|\theta^{(s)})p(\theta^{(s)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(s)})p(\theta^{(s)})}$$

Example: Normal distribution with known variance

- Metropolis approximation - acceptance ratio:

$$r = \frac{p(\theta^*|\boldsymbol{y})}{p(\theta^{(s)}|\boldsymbol{y})} = \left(\frac{\prod_{i=1}^n \text{dnorm}(y_i, \theta^*, \sigma)}{\prod_{i=1}^n \text{dnorm}(y_i, \theta^{(s)}, \sigma)}\right) \times \left(\frac{\text{dnorm}(\theta^*, \mu, \tau)}{\text{dnorm}(\theta^{(s)}, \mu, \tau)}\right).$$

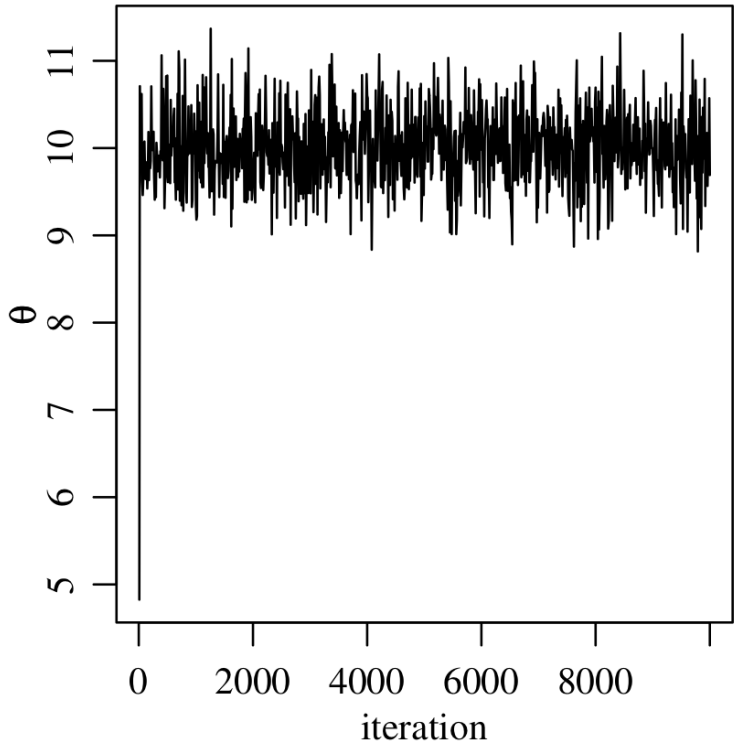
In many cases, computing the ratio r directly can be numerically unstable, a problem that often can be remedied by computing the logarithm of r:

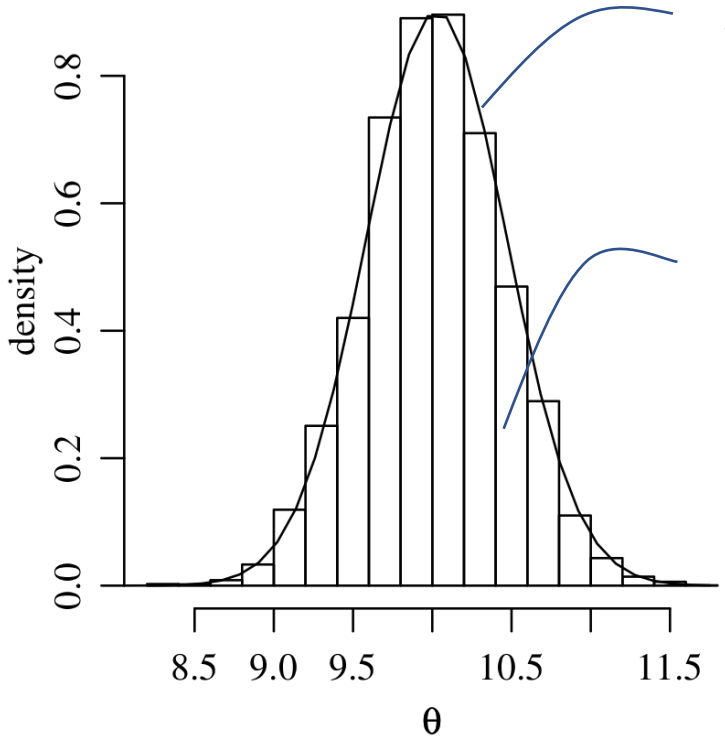
$$\log r = \sum_{i=1}^{n} [\log \text{dnorm}(y_i, \theta^*, \sigma) - \log \text{dnorm}(y_i, \theta^{(s)}, \sigma)] + \log \text{dnorm}(\theta^*, \mu, \tau) - \log \text{dnorm}(\theta^{(s)}, \mu, \tau).$$

- R코드 FCB p.176 참고:
  - Sample 10000개
  - Starting value $\theta^{(0)} = 0$
  - Proposal distribution: normal with var=2

Example: Normal distribution with known variance

Posterior mean(10.03)으로 수렴





곡선: True posterior N(10.03, 0.2)

히스토그램: Metropolis simulation

=> Clearly the empirical distribution of the simulated values is very close to the true posterior distribution.

Fig. 10.3. Results from the Metropolis algorithm for the normal model.

Output of the Metropolis algorithm

- The Metropolis algorithm generates a *dependent* sequence (Markov Chain).
  ⇒ **autocorrelation** 가능성
- As $S \to \infty$, the approximation is exact, but in practice we cannot run the Markov chain forever.
  ⇒ The standard practice in MCMC approximation (⊃ Metropolis, Gibbs sampler) is:

1. run algorithm until some iteration B for which it looks like the Markov chain has achieved stationarity; burn-in period
2. run the algorithm S more times, generating $\{\theta^{(B+1)}, \ldots, \theta^{(B+S)}\}$;
3. discard $\{\theta^{(1)}, \ldots, \theta^{(B)}\}$ and use the empirical distribution of $\{\theta^{(B+1)}, \ldots, \theta^{(B+S)}\}$ to approximate $p(\theta|y)$.

- Correlation can be adjusted by selecting an optimal value of δ.
  ⇒ decrease **correlation**, increase the **rate of convergence**, increase the **effective sample size**.

week2_part3_MCMC Diagnostics 참고

Output of the Metropolis algorithm

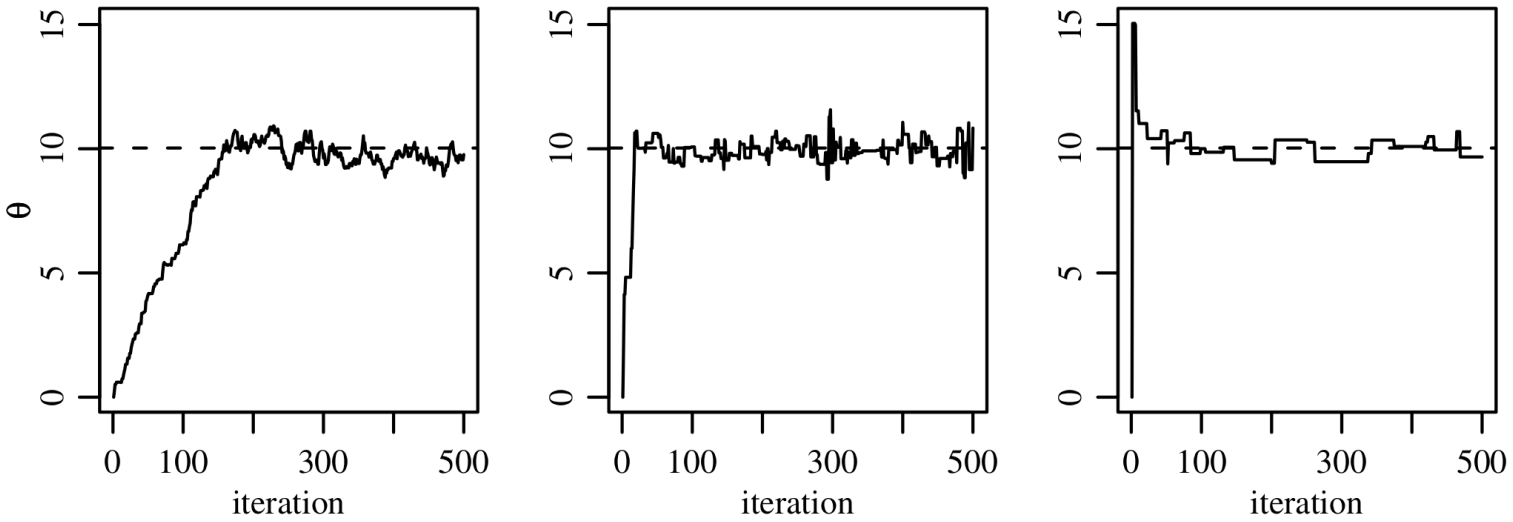


Fig. 10.4. Markov chains under three different proposal distributions. Going from left to right, the values of δ^2 are 1/32, 2 and 64 respectively.

10.3 The Metropolis algorithm for Poisson regression

Recall: Song sparrow reproductive success

- Model: $\{Y|\boldsymbol{x}\} \sim \text{Poisson}(\exp[\boldsymbol{\beta}^T\boldsymbol{x}])$ x: age, y: # of offspring

  $$\log \text{E}[Y|x] = \log \theta_x = \beta_1 + \beta_2 x + \beta_3 x^2$$

- Prior distributions for the regression coefficients ~ iid normal(0, 100)
- Metropolis approximation - acceptance ratio:

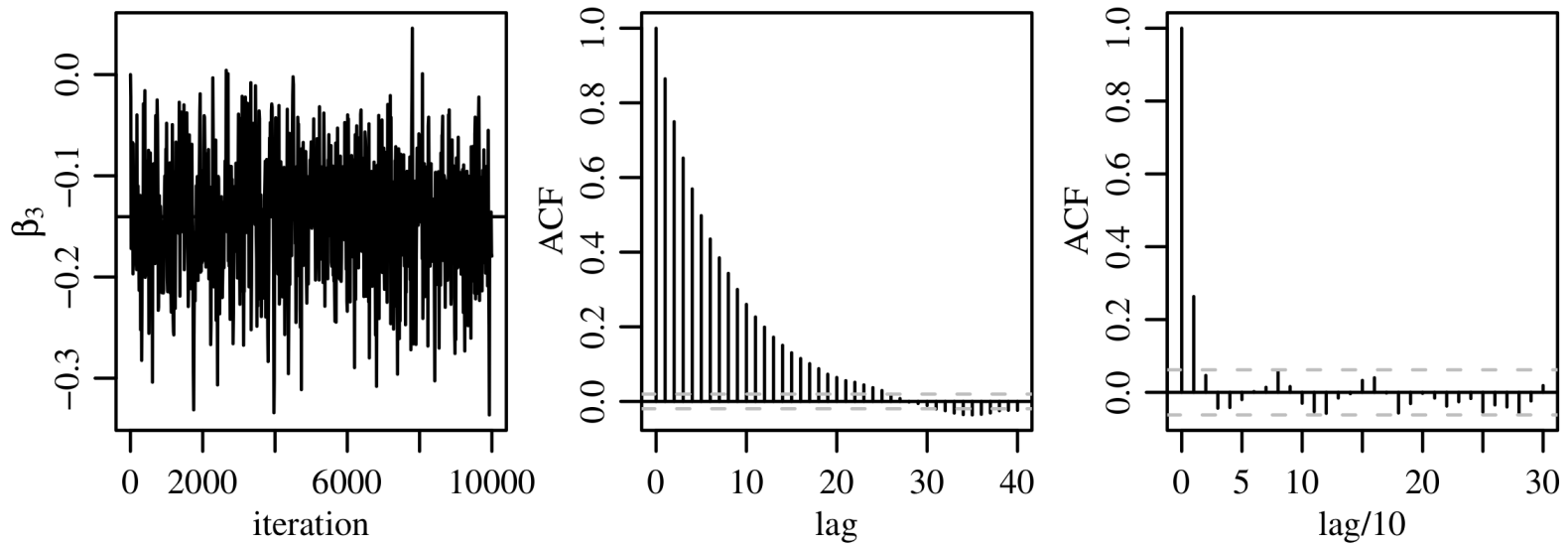
$$r = \frac{p(\boldsymbol{\beta}^*|\mathbf{X}, \boldsymbol{y})}{p(\boldsymbol{\beta}^{(s)}|\mathbf{X}, \boldsymbol{y})}$$

$$= \frac{\prod_{i=1}^n \text{dpois}(y_i, \boldsymbol{x}_i^T\boldsymbol{\beta}^*)}{\prod_{i=1}^n \text{dpois}(y_i, \boldsymbol{x}_i^T\boldsymbol{\beta}^{(s)})} \times \frac{\prod_{j=1}^3 \text{dnorm}(\beta_j^*, 0, 10)}{\prod_{j=1}^3 \text{dnorm}(\beta_j^{(s)}, 0, 10)}.$$

likelihood: pois prior: normal(0, 100)

- R코드 FCB p.176 참고

Recall: Song sparrow reproductive success



"thinning"

Fig. 10.5. Plot of the Markov chain in β_3 along with autocorrelation functions.

Recall: Song sparrow reproductive success

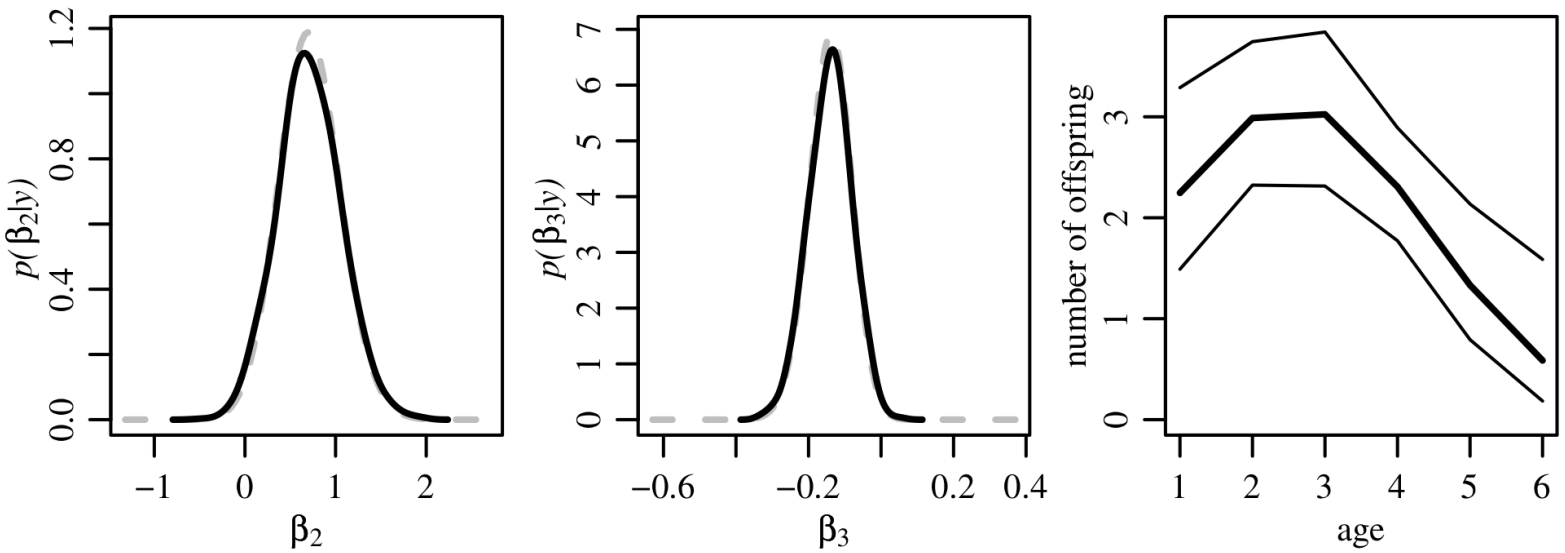
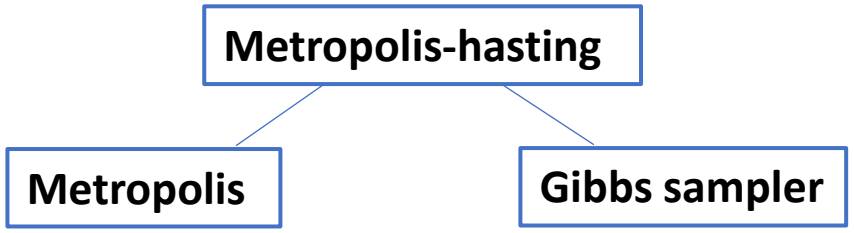


Fig. 10.6. The first two panels give the MCMC approximations to the posterior marginal distributions of β_2 and β_3 in black, with the grid-based approximations in gray. The third panel gives 2.5%, 50% and 97.5% posterior quantiles of $\exp(\boldsymbol{\beta}^T \boldsymbol{x})$.

10.4 Metropolis, Metropolis-hastings and Gibbs sampler

Recall that a Markov chain is a sequentially generated sequence $\{x^{(1)}, x^{(2)}, \ldots\}$ such that the mechanism that generates $x^{(s+1)}$ can depend on the value of $x^{(s)}$ but not on $\{x^{(s-1)}, x^{(s-2)}, \ldots x^{(1)}\}$.

In this section we will show that these two algorithms are in fact special cases of a more general algorithm, called the *Metropolis-Hastings algorithm*. We will then describe why Markov chains generated by the Metropolis-Hastings algorithm are able to approximate a target probability distribution. Since the Gibbs and Metropolis algorithms are special cases of Metropolis-Hastings, this implies that these two algorithms are also valid ways to approximate probability distributions.



10.4.1 The Metropolis-Hastings algorithm

Problem

We'll first consider a simple example where our target probability distribution is $p_0(u, v)$, a bivariate distribution for two random variables U and V. In the one-sample normal problem, for example, we would have $U = \theta$, $V = \sigma^2$ and $p_0(u, v) = p(\theta, \sigma^2|\boldsymbol{y})$.

Gibbs sampler

Recall that the Gibbs sampler proceeds by iteratively sampling values of U and V from their conditional distributions: Given $x^{(s)} = (u^{(s)}, v^{(s)})$, a new value of $x^{(s+1)}$ is generated as follows:

1. update U: sample $u^{(s+1)} \sim p_0(u|v^{(s)})$;
2. update V: sample $v^{(s+1)} \sim p_0(v|u^{(s+1)})$.

10.4.1 The Metropolis-Hastings algorithm

Metropolis

1. update U:
   a) sample $u^* \sim J_u(u|u^{(s)})$;
   b) compute $r = p_0(u^*, v^{(s)})/p_0(u^{(s)}, v^{(s)})$;
   c) set $u^{(s+1)}$ to u^* or $u^{(s)}$ with probability $\min(1, r)$ and $\max(0, 1-r)$.
2. update V:
   a) sample $v^* \sim J_v(v|v^{(s)})$;
   b) compute $r = p_0(u^{(s+1)}, v^*)/p_0(u^{(s+1)}, v^{(s)})$;
   c) set $v^{(s+1)}$ to v^* or $v^{(s)}$ with probability $\min(1, r)$ and $\max(0, 1-r)$.

Here, J_u and J_v are separate symmetric proposal distributions for U and V.

10.4.1 The Metropolis-Hastings algorithm

Metropolis-Hastings

- generalizes both by allowing arbitrary proposal distribution.

1. update U:
   a) sample $u^* \sim J_u(u|u^{(s)}, v^{(s)})$;
   b) compute the acceptance ratio

$$r = \frac{p_0(u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \times \boxed{\frac{J_u(u^{(s)}|u^*, v^{(s)})}{J_u(u^*|u^{(s)}, v^{(s)})}};$$

   c) set $u^{(s+1)}$ to u^* or $u^{(s)}$ with probability $\min(1, r)$ and $\max(0, 1-r)$.
2. update V:
   a) sample $v^* \sim J_v(v|u^{(s+1)}, v^{(s)})$;
   b) compute the acceptance ratio

$$r = \frac{p_0(u^{(s+1)}, v^*)}{p_0(u^{(s+1)}, v^{(s)})} \times \boxed{\frac{J_v(v^{(s)}|u^{(s+1)}, v^*)}{J_v(v^*|u^{(s+1)}, v^{(s)})}};$$

   c) set $v^{(s+1)}$ to v^* or $v^{(s)}$ with probability $\min(1, r)$ and $\max(0, 1-r)$.

J_u, J_s
- are not required to be symmetric
- do not depend on U and V values in our previous sequence => Markov chain

10.4.1 The Metropolis-Hastings algorithm

$$r = \frac{p_0(u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \times \boxed{\frac{J_u(u^{(s)}|u^*, v^{(s)})}{J_u(u^*|u^{(s)}, v^{(s)})}};$$

"correction factor"

1. **To "Metropolis"**
   - If proposed distributions are symmetric, correction factor will be 1
   - Then the acceptance probability is the same as in the Metropolis algorithm

2. **To "Gibbs sampler"**
   - If we use the full conditionals as proposal distributions $J_u(u^*|u^{(s)}, v^{(s)}) = p_0(u^*|v^{(s)})$.

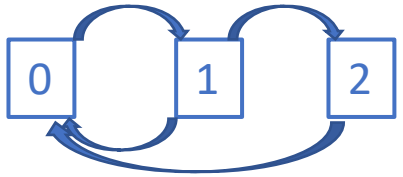
$$\begin{aligned} r &= \frac{p_0(u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \times \frac{J_u(u^{(s)}|u^*, v^{(s)})}{J_u(u^*|u^{(s)}, v^{(s)})} \\ &= \frac{p_0(u^*, v^{(s)})}{p_0(u^{(s)}, v^{(s)})} \frac{p_0(u^{(s)}|v^{(s)})}{p_0(u^*|v^{(s)})} \\ &= \frac{p_0(u^*|v^{(s)})p_0(v^{(s)})}{p_0(u^{(s)}|v^{(s)})p_0(v^{(s)})} \frac{p_0(u^{(s)}|v^{(s)})}{p_0(u^*|v^{(s)})} \\ &= \frac{p_0(v^{(s)})}{p_0(v^{(s)})} = 1, \end{aligned}$$

10.4.2 Why does the Metropolis-Hastings algorithm work?

Three properties of sequence $\{x^{(1)}, x^{(2)}, \dots\}$

- Irreducible(기약) : able to go from any one value of X to any other
  - Four states = {0,1,2,3}
  - Transition matrix = $\begin{Vmatrix} 0 & 0.1 & 0.9 & 0 \\ 0.8 & 0.1 & 0 & 0.1 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \end{Vmatrix}$ =>
  - Has two classes ⇔ reducible
  - Has one class ⇔ irreducible

- Aperiodic(비주기성) : make the distribution of X to converge
  - Period of state i : state i 로 다시 돌아오는데 걸리는 시간들의 최소공배수
  - If period = 1, aperiodic / else periodic



- 2 0 1 2 : 3times
- 2 0 1 0 1 2 : 5 times
- …

- Recurrent : return to special value of X from time to time as we run our Markov chain
  - 단계를 무한히 진행하여도 언젠가는 돌아올 수 있음을 뜻함. Not recurrent하다면 무한히 진행하였을 때 갈 수 있는 state가 없음.

10.4.2 Why does the Metropolis-Hastings algorithm work?

Theorem 2 *(Ergodic Theorem) If* $\{x^{(1)}, x^{(2)}, \ldots\}$ *is an irreducible, aperiodic and recurrent Markov chain, then there is a unique probability distribution* π *such that as* $s \to \infty$,

- $\Pr(x^{(s)} \in A) \to \pi(A)$ *for any set* A;
- $\frac{1}{S} \sum g(x^{(s)}) \to \int g(x)\pi(x)\, dx$.

The distribution π is called the *stationary distribution* of the Markov chain. It is called the stationary distribution because it has the following property:

> If $x^{(s)} \sim \pi$,
> and $x^{(s+1)}$ is generated from the Markov chain starting at $x^{(s)}$,
> then $\Pr(x^{(s+1)} \in A) = \pi(A)$.

In other words, if you sample $x^{(s)}$ from π and then generate $x^{(s+1)}$ conditional on $x^{(s)}$ from the Markov chain, then the *unconditional* distribution of $x^{(s+1)}$ is π. Once you are sampling from the stationary distribution, you are always sampling from the stationary distribution.

10.4.2 Why does the Metropolis-Hastings algorithm work?

"Proof" that $\pi(x) = p_0(x)$

The theorem above says that the stationary distribution of the Metropolis-Hastings algorithm is unique, and so if we show that p_0 is a stationary distribution, we will have shown it is *the* stationary distribution.

discrete random variable. Suppose $x^{(s)}$ is sampled from the target distribution p_0, and then $x^{(s+1)}$ is generated from $x^{(s)}$ using the Metropolis-Hastings algorithm. To show that p_0 is the stationary distribution we need to show that $\Pr(x^{(s+1)} = x) = p_0(x)$.

p.185~187

10.5 Combining the Metropolis and Gibbs algorithms

In complex models it is often the case that conditional distributions are available for some parameters but not for others. In these situations we can combine Gibbs and Metropolis-type proposal distributions to generate a Markov chain to approximate the joint posterior distribution of all of the parameters.

section we do this in the context of estimating the parameters in a regression model for time-series data where the errors are temporally correlated. In this case, full conditional distributions are available for the regression parameters but not the parameter describing the dependence among the observations.

- Data에 따라 모델에 있는 각각의 parameter의 conditional distribution을 구할 수 없는 경우들도 있어서, Gibbs와 Metropolis algorithm을 같이 쓴다.

10.5 Combining the Metropolis and Gibbs algorithms

Ice core data

Scientists deduced historical atmospheric conditions of the last few hundred thousands years.

- N = 200
- X = CO2 concentration value
- Y = Temperature
- OLS : $\widehat{\mathrm{E}}[Y|x] = -23.02 + 0.08x$

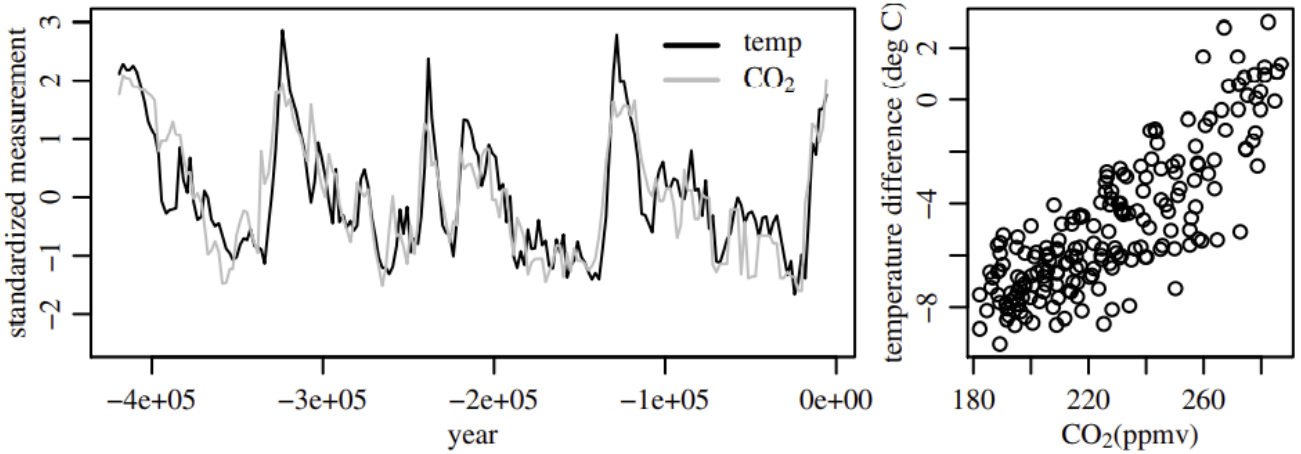


Fig. 10.7. Temperature and carbon dioxide data.

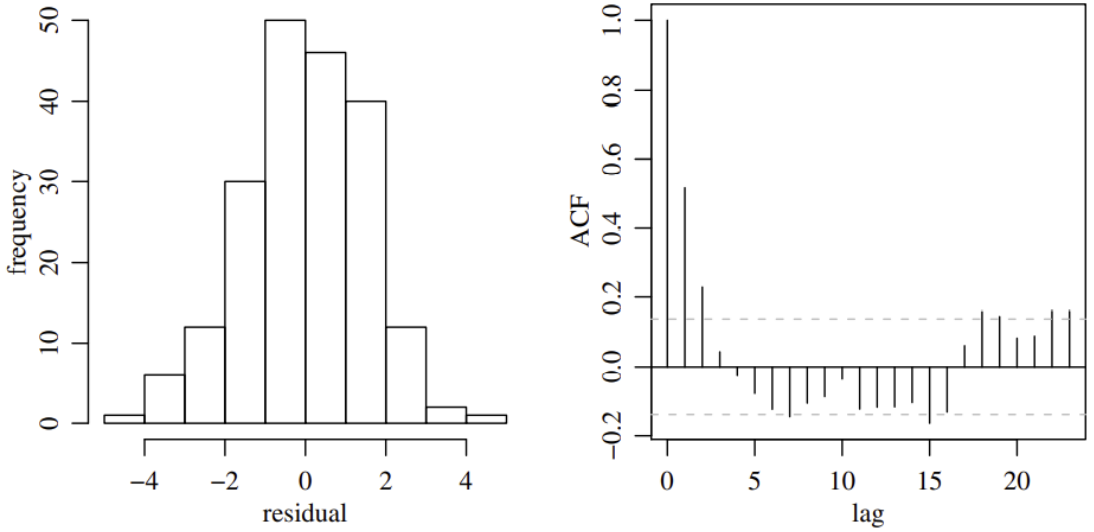


Fig. 10.8. Residual analysis for the least squares estimation.

10.5.1 A regression model with correlated errors

- Ordinary regression model

$$\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \sim \text{multivariate normal}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}).$$

- Need more appropriate model
- Error terms are temporally correlated
- We must change covariance matrix, in which can represent positive correlation between sequential observations.
- *"first-order autoregressive structure"*

$$\Sigma = \sigma^2\mathbf{C}_\rho = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \rho^2 & \rho & 1 & & \\ \vdots & \vdots & & \ddots & \\ \rho^{n-1} & \rho^{n-2} & & & 1 \end{pmatrix}$$

10.5.1 A regression model with correlated errors

Having observed $\boldsymbol{Y} = \boldsymbol{y}$, the parameters to estimate in this model include $\boldsymbol{\beta}$, σ^2 and ρ. Using the multivariate normal and inverse-gamma prior distributions of Section 9.2.1 for $\boldsymbol{\beta}$ and σ^2, it is left as an exercise to show that

$$
\begin{aligned}
\{\boldsymbol{\beta}|\mathbf{X},\boldsymbol{y},\sigma^2,\rho\} &\sim \text{multivariate normal}(\boldsymbol{\beta}_n,\Sigma_n)\ ,\ \text{where} \qquad (10.2)\\
\Sigma_n &= (\mathbf{X}^T\mathbf{C}_\rho^{-1}\mathbf{X}/\sigma^2+\Sigma_0^{-1})^{-1}\\
\boldsymbol{\beta}_n &= \Sigma_n(\mathbf{X}^T\mathbf{C}_\rho^{-1}\boldsymbol{y}/\sigma^2+\Sigma_0^{-1}\boldsymbol{\beta}_0)\ ,\ \text{and}\\
\{\sigma^2|\mathbf{X},\boldsymbol{y},\boldsymbol{\beta},\rho\} &\sim \text{inverse-gamma}([\nu_0+n]/2,[\nu_0\sigma_0^2+\text{SSR}_\rho]/2)\ ,\ \text{where}\\
\text{SSR}_\rho &= (\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^T\mathbf{C}_\rho^{-1}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta}).
\end{aligned}
$$

=> Gibbs sampler

- Section 9.2.1 참고

If we knew the value of ρ we could use the Gibbs sampler to approximate $p(\boldsymbol{\beta},\sigma^2|\mathbf{X},\boldsymbol{y},\rho)$ by iteratively sampling from the full conditional distributions given by the equations in 10.2. Of course ρ is unknown and so we will need to estimate it as well with our Markov chain. Unfortunately the full conditional distribution for ρ will be nonstandard for most prior distributions, suggesting that the Gibbs sampler is not applicable here and we may have to use a Metropolis algorithm (although a discrete approximation to $p(\rho|\mathbf{X},\boldsymbol{y},\boldsymbol{\beta},\sigma^2)$ could be used).

=> Metropolis

10.5.1 A regression model with correlated errors

Step1,2 : Gibbs sampler
Step3 : Metropolis

1. Update $\boldsymbol{\beta}$: Sample $\boldsymbol{\beta}^{(s+1)} \sim$ multivariate normal$(\boldsymbol{\beta}_n, \Sigma_n)$, where $\boldsymbol{\beta}_n$ and Σ_n depend on $\sigma^{2(s)}$ and $\rho^{(s)}$.
2. Update σ^2: Sample $\sigma^{2(s+1)} \sim$ inverse-gamma$([\nu_0+n]/2, [\nu_0\sigma_0^2+\text{SSR}_\rho]/2)$, where SSR_ρ depends on $\boldsymbol{\beta}^{(s+1)}$ and $\rho^{(s)}$.
3. Update ρ:
   a) Propose $\rho^* \sim$ uniform$(\rho^{(s)}-\delta, \rho^{(s)}+\delta)$. If $\rho^* < 0$ then reassign it to be $|\rho^*|$. If $\rho^* > 1$ reassign it to be $2-\rho^*$.
   b) Compute the acceptance ratio

proposal distribution = "reflecting random walk"

$$r = \frac{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\beta}^{(s+1)}, \sigma^{2(s+1)}, \rho^*)p(\rho^*)}{p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\beta}^{(s+1)}, \sigma^{2(s+1)}, \rho^{(s)})p(\rho^{(s)})}$$

and sample $u \sim$ uniform(0,1). If $u < r$ set $\rho^{(s+1)} = \rho^*$, otherwise set $\rho^{(s+1)} = \rho^{(s)}$.

10.5.2 Analysis of the ice core data

We'll use diffuse prior distributions for the parameters, with $\boldsymbol{\beta}_0 = \mathbf{0}$, $\Sigma_0 = \text{diag}(1000)$, $\nu_0 = 1$ and $\sigma_0^2 = 1$. Our prior for ρ will be the uniform distribution on $(0, 1)$. The first panel of Figure 10.9 plots the first 1,000 values $\{\rho^{(1)}, \ldots, \rho^{(1000)}\}$ generated using the Metropolis-Hastings algorithm above. The acceptance rate for these values is 0.322 which seems good, but the autocorrelation of the sequence, shown in the second panel, is very high. The effective sample size for this correlated sequence of 1,000 ρ-values is only 23, indicating that we will need many more iterations of the algorithm to obtain a decent approximation to the posterior distribution.

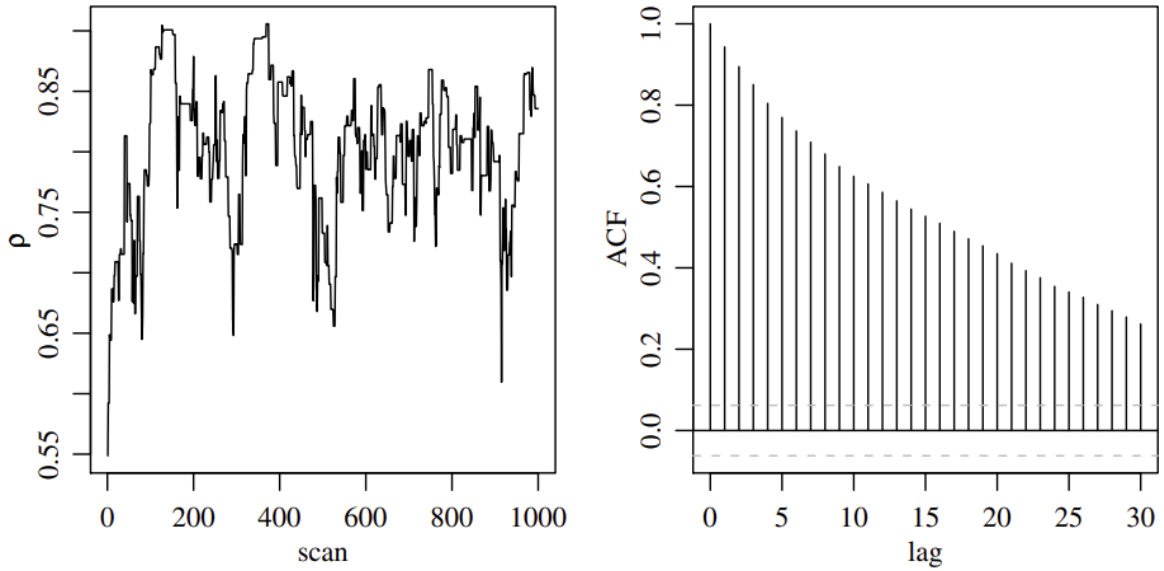


Fig. 10.9. The first 1,000 values of ρ generated from the Markov chain.

10.5.2 Analysis of the ice core data

Suppose we were to generate 25,000 scans for a total of $25,000 \times 4 = 100,000$ parameter values. Storing and manipulating all of these values can be tedious and somewhat unnecessary: Since the Markov chain is so highly correlated, the values of $\rho^{(s)}$ and $\rho^{(s+1)}$ offer roughly the same information about the posterior distribution. With this in mind, for highly correlated Markov chains with moderate to large numbers of parameters we will often only save a fraction of the scans of the Markov chain. This practice of throwing away many iterations of a Markov chain is known as *thinning*. Figure 10.10 shows the thinned output of a 25,000-scan Markov chain for the ice core data, in which only every 25th scan was saved. Thinning the output reduces it down to a manageable 1,000 samples, having a much lower autocorrelation than 1,000 sequential samples from an unthinned Markov chain.

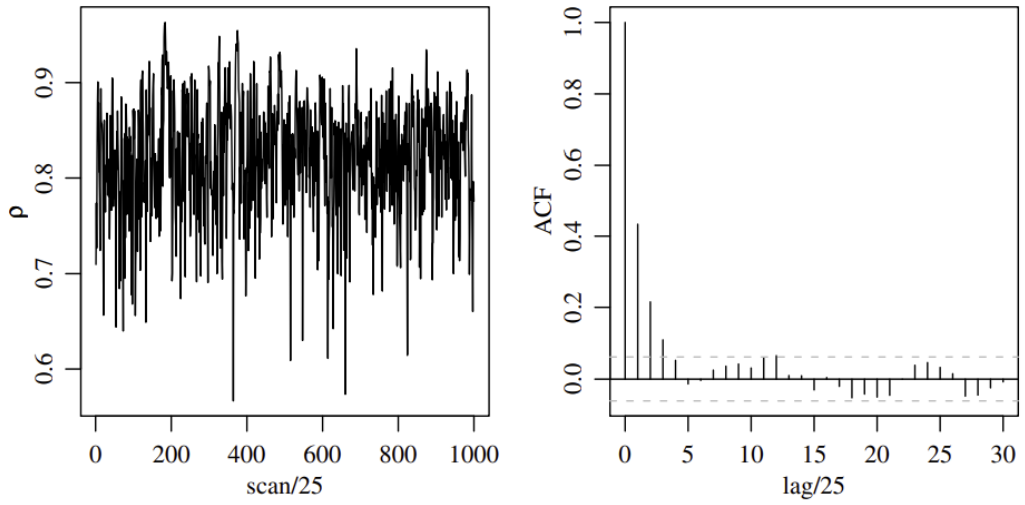
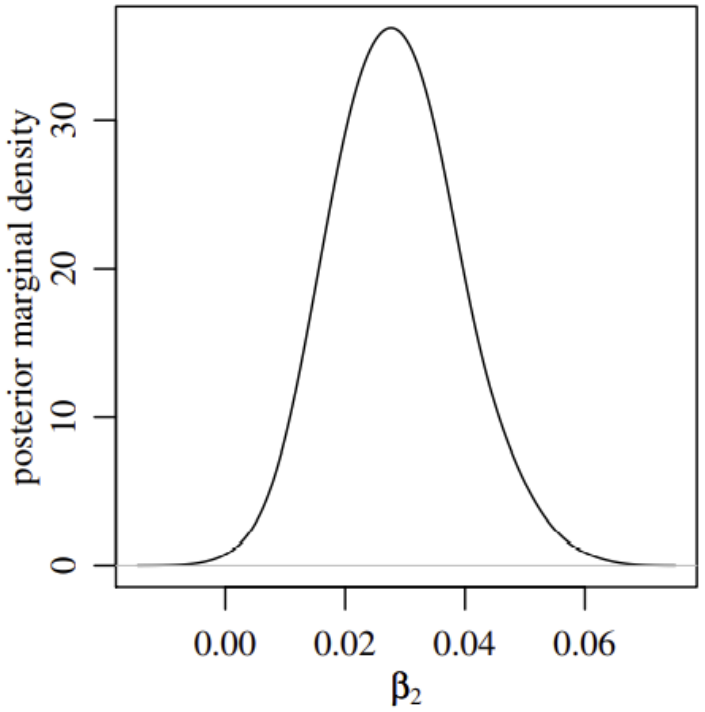


Fig. 10.10. Every 25th value of ρ from a Markov chain of length 25,000.

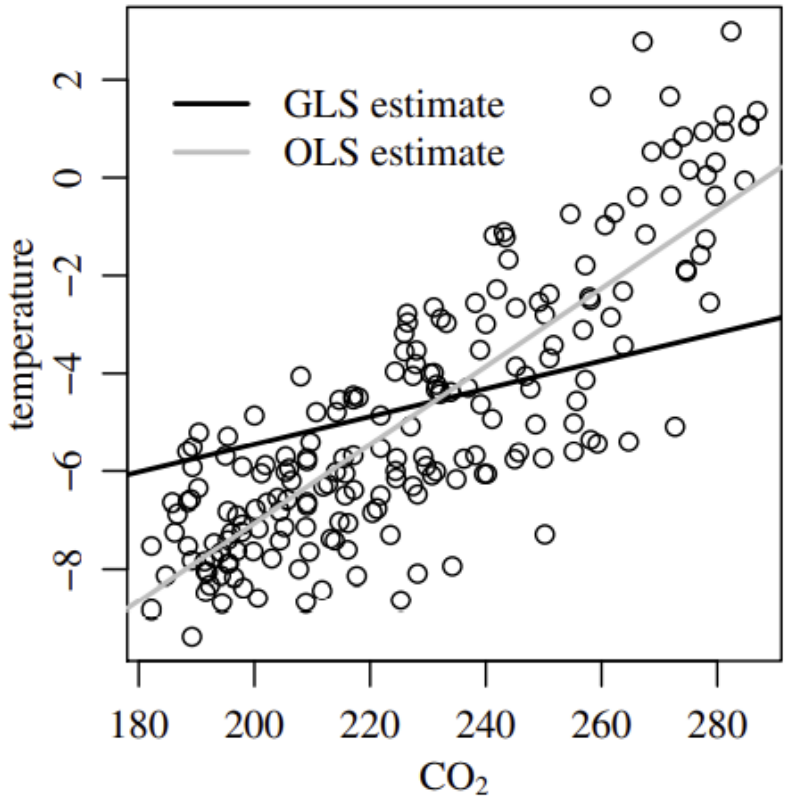
10.5.2 Analysis of the ice core data



The Monte Carlo approximation to the posterior density of β_2, the slope parameter, appears in the first panel of Figure 10.11. The posterior mean of β_2 is 0.028 and a posterior 95% quantile-based confidence interval is $(0.01, 0.05)$, indicating evidence that the relationship between CO_2 and temperature is positive.

10.5.2 Analysis of the ice core data

positive. However, as indicated in the second plot this relationship seems much weaker than that suggested by the OLS estimate of 0.08. For the OLS estimation, the small number of data points with high y-values have a larger amount of influence on the estimate of β. In contrast, the GLS model recognizes that many of these extreme points are highly correlated with one another and down-weights their influence. We note that this "weaker" regression coefficient is a result of the temporally correlated data, and not of the particular prior distribution we used or the Bayesian approach in general. The reader is encouraged to repeat the analysis with different prior distributions, or to perform a non-Bayesian GLS estimation for comparison. In any case, the data analysis indicates evidence of a relationship between temperature measurements and the CO_2 measurements that precede them in time.



HW

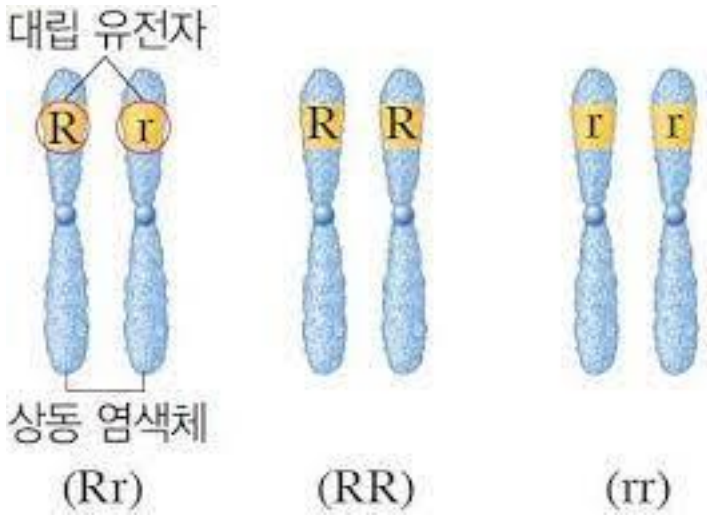
대립유전자

Example 2: Estimating an allele frequency

A standard assumption when modelling genotypes of bi-allelic loci (e.g. loci with alleles A and a) is that the population is "randomly mating". From this assumption it follows that the population will be in "Hardy Weinberg Equilibrium" (HWE), which means that if p is the frequency of the allele A then the genotypes AA, Aa and aa will have frequencies $p^2, 2p(1-p)$ and $(1-p)^2$ respectively.

A simple prior for p is to assume it is uniform on $[0, 1]$. Suppose that we sample n individuals, and observe n_{AA} with genotype AA, n_{Aa} with genotype Aa and n_{aa} with genotype aa.

⇒ Metropolis algorithm to sample from the posterior distribution of p.



Sample:

Running this sample for $n_{AA} = 50, n_{Aa} = 21, n_{aa}=29$.

$$\Rightarrow nA = 50 + 50 + 21 = 121, \quad na = 29 + 29 + 21 = 79$$

Theoretical posterior:

In this case is available analytically; since we observed 121 As, and 79 as, out of 200, the posterior for p is Beta(121+1,79+1).